SERVIDOR WEB EN AMAZON WEB SERVICE

Ejercicio 1. Lanzamiento de un servidor web Apache en AWS:

El objetivo de esta práctica es crear una máquina virtual de Ubuntu en Amazon Web Service e instalar un servidor web de Apache.

Para realizarlo, puedes seguir este enlace donde se explica paso a paso el procedimiento:

https://jairogarciarincon.com/clase/curso-instalacion-y-puesta-en-marcha-de-un-entorno-aws/configuracion-creacion-y-lanzamiento-de-una-instancia-ec2

Sube una página web diferente a la de por defecto y prueba a conectarte con un navegador.

Ejercicio 2. Lanzamiento de un servidor web Apache en AWS mediante un contenedor de Docker.

En la instancia anterior, instala Docker siguiendo la documentación oficial:

https://docs.docker.com/engine/install/ubuntu/

Docker es una aplicación que nos permite crear contenedores a partir de un repositorio de imágenes públicas “Docker hub”. Un contenedor es un entorno independiente y aislado para lanzar aplicaciones. Lo que nos permite desplegar un contenedor y ponerlo en línea de manera muy rápida y segura.

Ejercicio 3. Pequeño servidor web en Java

Analiza y prueba el código que implementa un servidor web mediante sockets. El código de este servidor lo puedes encontrar en el apartado ejemplos de la página de Moodle del módulo. El servidor acepta peticiones por el puerto 8066 de un cliente. Según la URL el servidor contesta con diferente información:
http://localhost:8066 -> da mensaje de bienvenida
http://localhost:8066/mendoza muestra un párrafo de un libro
Para cualquier otra url, por ejemplo http://localhost:8066/uno -> mensaje de error.
Para programar un servidor HTTP se ha seguido el siguiente esquema:

- Crear el SocketServer asociado al puerto 8066.
- Esperar peticiones de los clientes.
- Aceptar peticiones de los clientes.
- Procesar la petición: intercambio de mensajes según protocolo + transmisión
- Cerrar socket cliente.

Ejecuta el servidor y conecta cualquier navegador (Firefox, Chrome, etc) y obtén las páginas web del servidor.

Una vez lo pruebes en local con un navegador web, el objetivo es ejecutar un .jar de esta aplicación dentro de un contenedor Docker en la máquina de AWS.

Para realizarlo tendrás que hacer estos pasos:

1. Crear un .jar de la aplicación.
2. Subir el .jar a la máquina virtual. Hay varias maneras de copiar un fichero local a una máquina remota, pero la más utilizada es con el comando **scp**. En esta página puedes ver información de este comando:

   https://www.ionos.es/digitalguide/servidores/configuracion/scp-de-linux/

3. Crear un fichero Dockerfile en el mismo directorio donde esté el fichero .jar que tendrá un aspecto similar al siguiente:

```
FROM openjdk:18-jdk-alpine3.15.
WORKDIR /app
EXPOSE 5000
COPY Servidor.jar Servidor.jar
CMD ["java", "-jar", "Servidor.jar"]
```

En la línea FROM se especifica que queremos usar una imagen que tenga instalado el openjdk 18 con el SO Alpine. La versión del jdk tiene que coincidir con la versión con la cual se ha compilado el código fuente.
Con WORKDIR se establece el directorio de trabajo dentro del contenedor.
EXPOSE hace que el contenedor exponga el puerto 5000 que es donde está escuchando el servidor.
COPY copiamos el .jar desde la máquina local al contenedor.
CMD el comando que queremos ejecutar en el contenedor.

4. A continuación, realizar los siguientes comandos:
   a. Construir la imagen
      i. docker build -t servidor:v1 **.**
   b. Para ver las imágenes guardadas
      i. docker images
   c. Ejecutar contenedor
      i. docker run –name servidor -p 5000:5000 servidor:v1
   d. Para borrar una imagen guardada
      i. Docker rmi 7b23f7db6e8b (siendo lo último el identificador de la imagen que se obtiene con el comando docker images)

Ejercicio 4: Mejora del pequeño servidor web

Se trata de modificar el servidor web del ejemplo anterior para establecerle dos mejoras.

1. Una de ellas, es convertirlo en un servidor multihilo de tal manera que acepte conexiones simultáneas de varios clientes.
2. La última es que devuelva también la cabecera Date con la fecha y hora actuales en el formato que especifica el protocolo.