Maestría
en Inteligencia
Analítica de Datos

Despliegue con contenedores: tableros y APIs

Instrucciones:

En este taller exploraremos el despliegue de soluciones analíticas usando contenedores Docker, conectando un tablero con una API de predicción.

La entrega de este taller consiste en un reporte en formato de documento de texto, donde pueda incorporar fácilmente capturas de pantalla, textos y elementos similares. Puede utilizar formatos como Word, LibreOffice, Markdown, u otros.

Parte 1: despliegue de la API

En esta primera parte usaremos una máquina virtual para desplegar allí nuestra API empleando un contenedor Docker.

1. En la consola de EC2 lance una instancia t2.small, **Ubuntu server** con 20 GB de disco. **Incluya un pantallazo de la consola de AWS EC2 con la máquina en ejecución en su reporte. Su usuario de AWS y las IPs privada y pública deben estar visible en el pantallazo.**
2. Para conectarse a la instancia, en una terminal emita el comando

   ```
   ssh -i /path/to/llave.pem ubuntu@IP
   ```

   donde */path/to/* se refiere a la ubicación del archivo *llave.pem* que descargó, e IP es la dirección IP pública de la instancia EC2 que lanzó. Si prefiere, en la terminal puede navegar a la ubicación del archivo llave.pem y emitir el comando

   ```
   ssh -i llave.pem ubuntu@IP
   ```

   Note que usamos en este caso ubuntu en vez de ec2-user, pues éste es el usuario creado por defecto como administrador con sistema operativo Ubuntu server.
3. Ahora pasamos a instalar Docker en la máquina, para lo cual requerimos eliminar posible versiones anteriores y agregar el repositorio de la última versión estable de Docker.
4. En la máquina virtual, elimine versiones anteriores de Docker (esto genera un error si no hay versiones anteriores, en cuyo caso puede continuar sin problema)


```
sudo apt-get remove docker docker-engine docker.io containerd runc
```

5. Actualice el índice de paquetes

```
sudo apt-get update
```

6. Instale dependencias para verificar certificados (ca-certificates), obtener objetos con su URL (curl) y administrar llaves PGP (gnupg)

```
sudo apt-get install ca-certificates curl gnupg
```

7. Agregue la llave de Docker

```
sudo install -m 0755 -d /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -
    o /etc/apt/keyrings/docker.gpg

sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

8. Agregue el repositorio de Docker a su sistema para la instalación

```
echo \
  "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker
      .gpg] https://download.docker.com/linux/ubuntu \
  "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

9. Actualice nuevamente el índice de paquetes con este nuevo repositorio incluido

```
sudo apt-get update
```

10. Instale Docker Engine, containerd, y Docker Compose

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-
    plugin docker-compose-plugin
```

11. Para verificar su instalación, descargue, construya y ejecute la imagen hello-world

```
sudo docker run hello-world
```

12. Adjunto a este enunciado encontrará el archivo *docker-api-starter.zip*. Descomprima el archivo **localmente** en una carpeta, que llamaremos raíz, tal que allí quede la carpeta bankchurn-api y el archivo Dockerfile. En esta carpeta raíz cree un repositorio git, y conéctelo con un repositorio nuevo en GitHub. Al terminar debe tener en su carpeta raíz 2 carpetas, .git y bankchurn-api, así como el archivo Dockerfile.

13. Regrese a la terminal de su máquina virtual en EC2 y clone allí el repositorio. Ingrese a la carpeta del repositorio, donde debe encontrar la misma estructura de carpetas (.git, bankchurn-api) y el Dockerfile.
14. A partir del archivo Dockerfile construya la imagen que usará más adelante para lanzar contenedores

    ```
    sudo docker build -t bankchurn-api:latest .
    ```

    Note que el comando termina con un espacio y un punto. Esto indica que se usa la carpeta actual como base para construir la imagen.
15. Liste las imágenes de docker con el comando

    ```
    sudo docker images
    ```

    Debe contar con la imagen recién creada de bankchurn-api y la hello-world creada anteriormente. Incluya un pantallazo de la salida en su **reporte**. Su IP privada debe ser visible.
16. Ahora ejecute un contenedor usando la imagen creada con el comando

    ```
    sudo docker run -p 8001:8001 -it -e PORT=8001 bankchurn-api
    ```

    Incluya un pantallazo de la salida de este comando en su **reporte**. La IP privada debe ser visible.
17. Vaya ahora a la consola de EC2, seleccione su máquina virtual y modifique el grupo de seguridad para permitir tráfico por el puerto 8001. Es decir, en el grupo de seguridad de la máquina edite las reglas de entrada y agregue una que permita tráfico por el puerto TCP 8001 desde cualquier IP (anywhere IPv4).
18. Copie la IP pública de su máquina y en un navegador local visite la página IP:8001. Allí debe aparecer la API de bankchurn en ejecución. Incluya un pantallazo del navegador en su **reporte**. La dirección (IP pública) debe ser visible.

Parte 2: despliegue del tablero

Ahora usaremos otra máquina virtual para desplegar allí el tablero empleando un contenedor Docker. **Tenga en cuenta que es imprescindible mantener encendida la máquina virtual utilizada en la Parte 1 para realizar esta sección.**

1. En la consola de EC2 lance otra instancia t2.small, **Ubuntu server** con 20 GB de disco. **Incluya un pantallazo de la consola de AWS EC2 con la máquina en ejecución en su reporte. Su usuario de AWS y las IPs privada y pública deben estar visible en el pantallazo.**
2. Para conectarse a la instancia, en **otra terminal** emita el comando

    ```
    ssh -i /path/to/llave.pem ubuntu@IP
    ```

donde */path/to/* se refiere a la ubicación del archivo *llave.pem* que descargó, e IP es la dirección IP pública de la instancia EC2 que lanzó. Si prefiere, en la terminal puede navegar a la ubicación del archivo llave.pem y emitir el comando

```
ssh -i llave.pem ubuntu@IP
```

Note que usamos en este caso ubuntu en vez de ec2-user, pues éste es el usuario creado por defecto como administrador con sistema operativo Ubuntu server.

3. Ahora pasamos a instalar Docker en la máquina, para lo cual requerimos eliminar posible versiones anteriores y agregar el repositorio de la última versión estable de Docker.
4. En la máquina virtual, elimine versiones anteriores de Docker (esto genera un error si no hay versiones anteriores, en cuyo caso puede continuar sin problema)

   ```
   sudo apt-get remove docker docker-engine docker.io containerd runc
   ```

5. Actualice el índice de paquetes

   ```
   sudo apt-get update
   ```

6. Instale dependencias para verificar certificados (ca-certificates), obtener objetos con su URL (curl) y administrar llaves PGP (gnupg)

   ```
   sudo apt-get install ca-certificates curl gnupg
   ```

7. Agregue la llave de Docker

   ```
   sudo install -m 0755 -d /etc/apt/keyrings

   curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -
       o /etc/apt/keyrings/docker.gpg

   sudo chmod a+r /etc/apt/keyrings/docker.gpg
   ```

8. Agregue el repositorio de Docker a su sistema para la instalación

   ```
   echo \
     "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker
         .gpg] https://download.docker.com/linux/ubuntu \
     "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
     sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
   ```

9. Actualice nuevamente el índice de paquetes con este nuevo repositorio incluido

   ```
   sudo apt-get update
   ```

10. Instale Docker Engine, containerd, y Docker Compose


```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-
    plugin docker-compose-plugin
```

11. Para verificar su instalación, descargue, construya y ejecute la imagen hello-world

```
sudo docker run hello-world
```

12. Adjunto a este enunciado encontrará el archivo *docker-dash-starter.zip*. Descomprima el archivo **localmente** en una carpeta, que llamaremos raíz, tal que allí quede la carpeta appy el archivo Dockerfile. En esta carpeta raíz cree un repositorio git, y conéctelo con un repositorio nuevo en GitHub. Al terminar debe tener en su carpeta raíz 2 carpetas, .git y app, así como el archivo Dockerfile.
13. Regrese a la terminal de su máquina virtual en EC2 y clone allí el repositorio. Ingrese a la carpeta del repositorio, donde debe encontrar la misma estructura de carpetas (.git, app) y el Dockerfile.
14. A partir del archivo Dockerfile construya la imagen que usará más adelante para lanzar contenedores

    ```
    sudo docker build -t bankchurn-dash:latest .
    ```

    Note que el comando termina con un espacio y un punto. Esto indica que se usa la carpeta actual como base para construir la imagen.
15. Liste las imágenes de docker con el comando

    ```
    sudo docker images
    ```

    Debe contar con la imagen recién creada de bankchurn-dash y la hello-world creada anteriormente. Incluya un pantallazo de la salida en su **reporte**. Su IP privada debe ser visible.
16. Ahora ejecute un contenedor usando la imagen creada con el comando

    ```
    sudo docker run -p 8050:8050 -it -e PORT=8050 -e API_URL=X.Y.Z.W bankchurn-
        dash
    ```

    cambiando X.Y.Z.W por la IP pública de la máquina donde está corriendo la API. Incluya un pantallazo de la salida de este comando en su **reporte**. La IP privada debe ser visible.
17. Vaya ahora a la consola de EC2, seleccione su máquina virtual y modifique el grupo de seguridad para permitir tráfico por el puerto 8050. Es decir, en el grupo de seguridad de la máquina edite las reglas de entrada y agregue una que permita tráfico por el puerto TCP 8050 desde cualquier IP (anywhere IPv4).
18. Copie la IP pública de su máquina y en un navegador local visite la página IP:8050. Allí debe aparecer el tablero de bankchurn en ejecución. Incluya un pantallazo del navegador en su **reporte**. La dirección (IP pública) debe ser visible.

19. Interactúe con el tablero y verifique que las solicitudes que se realizan se transmiten a la API y de allí de obtienen las predicciones que aparecen en el tablero. Incluya en su **reporte** un pantallazo donde aparezca la interfaz del tablero y las terminales de los dos contenedores en ejecución, mostrando los registros de la ejecución. Deben ser visibles la IP pública en el tablero y las IPs privadas en las terminales.
20. Compare los Dockerfile del tablero y de la API y comente en su **reporte** tres diferencias que considere sustanciales.

Más allá: ECS y Railway

1. Note que el despliegue anterior en máquinas virtuales lo realizamos usando contenedores Docker.
2. Con estos mismos artefactos (Dockerfile y carpetas del tablery la API) podemos realizar un despliegue similar usando los servicios ECS y Railway.
3. Tanto en ECS como en Railway podemos usar el Dockerfile como base para el despliegue sin ajustar nada más en nuestros artefactos.
4. Como ya hemos realizado este tipo de despliegue en semanas anteriores y estamos en entrega de proyecto, no le solicitamos hacerlo nuevamente, pero lo invitamos a probarlo opcionalmente.