



MIAD

Maestría
en Inteligencia
Analítica de Datos



Despliegue con contenedores en la nube: Docker y AWS ECS

Instrucciones:

En este taller exploraremos el despliegue de soluciones analíticas usando contenedores Docker en el servicio AWS ECS.

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 con contenedores en ECS

1. En esta actividad necesitamos las credenciales de AWS para realizar algunas operaciones. En AWS Academy, inicie el Laboratorio de aprendizaje. En el menú de la esquina superior derecha encontrará el ítem AWS Details. Allí se encuentra el ítem AWS CLI, de click en Show. Copie los datos de conexión que allí aparecen: `aws_access_key_id`, `aws_secret_access_key` y `aws_session_token`. Guárdelos en un archivo de texto para usarlos más adelante.
2. Vaya ahora a la consola de AWS y en la esquina superior izquierda (Servicios) busque el servicio EC2.
3. 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.**
4. 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.

5. Actualice el índice de paquetes

```
sudo apt-get update
```

6. Instale zip unzip

```
sudo apt install zip unzip
```

7. Copie localmente los instaladores de la interfaz de línea de comandos de AWS (CLI)

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
```

8. Descomprima el archivo y ejecute el instalador

```
unzip awscliv2.zip
```

```
sudo ./aws/install
```

9. Verifique la versión de AWS CLI

```
aws --version
```

10. Configure la llave de acceso y la llave secreta de AWS con el comando

```
aws configure
```

el cual le solicitará el AWS Access Key y el AWS Secret Access Key, los cuales copió al inicio del taller. También se solicita Region, donde debe especificar us-east-1. Finalmente le pide *Default output format*, el cuál puede saltar con un enter.

11. Configure ahora el token de la sesión con el comando

```
aws configure set aws_session_token SESSIONTOKEN
```

reemplazando SESSIONTOKEN por el token que copió al inicio del taller.

12. Verifique que su cuenta quedó bien configurada solicitando el listado de usuarios de IAM

```
aws iam list-users
```

que debe retornar un listado vacío en formato JSON.

13. Vaya a la consola de AWS; en la esquina superior izquierda (Servicios) busque el servicio ECR o Elastic Container Registry. Éste es el servicio de registro de imágenes para contenedores.

14. Cree un repositorio dando click en Crear repositorio.



15. Seleccione repositorio Privado. Defina un nombre para el repositorio.
16. Click en Crear repositorio.
17. Una vez creado el repositorio regresa a la consola de Amazon ECR y allí puede ver la lista de repositorios. Seleccione el repositorio creado y copie la URI del repositorio, que tiene un formato como el siguiente

```
1234567890.dkr.ecr.us-east-1.amazonaws.com/nombrerepositorio
```

Parte 2: instalación de Docker, creación de la imagen y enlace con AWS ECR

1. Ahora pasamos a instalar Docker en la máquina, para lo cual requerimos eliminar posibles versiones anteriores y agregar el repositorio de la última versión estable de Docker.
2. 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
```

3. 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
```

4. 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
```

5. 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
```

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

```
sudo apt-get update
```



7. Instale Docker Engine, containerd, y Docker Compose

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

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

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

9. 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*.

10. 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.

11. 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.

12. Ahora etiquete la imagen creada con la URI del repositorio en ECS

```
sudo docker tag bankchurn-api:latest repoURI
```

reemplazando *repoURI* por la URI del repositorio copiada anteriormente.

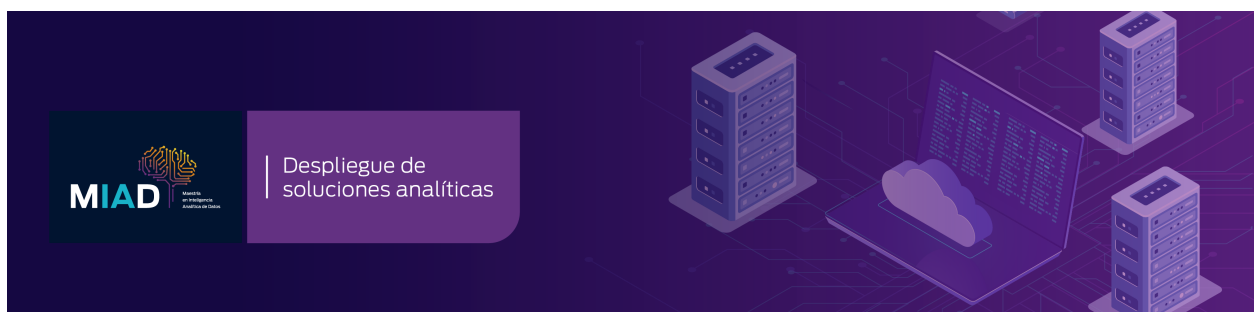
13. Liste nuevamente las imágenes de docker con el comando

```
sudo docker images
```

Debe contar con la imagen recién etiquetada. Incluya un pantallazo de la salida en su **reporte**. Su IP privada debe ser visible.

14. Ahora debemos loguearnos en el registro de AWS ECR para subir la imagen que creamos. Ejecute el comando

```
aws ecr get-login-password --region us-east-1 | sudo docker login --username AWS --password-stdin repoURI
```



reemplazando repoURI por la URI del repositorio copiada anteriormente. Debe recibir una confirmación de logueo exitoso. Incluya un pantallazo de la salida en su **reporte**. Su IP privada debe ser visible.

15. Ahora podemos enviar la imagen del contenedor al registro de AWS ECR con el comando
`sudo docker push repoURI`

reemplazando repoURI por la URI del repositorio copiada anteriormente. Esta operación toma unos minutos. Al terminar tome un pantallazo de la salida para su **reporte**. Su IP privada debe ser visible.

16. Confirme que la imagen subió al registro. Para esto vaya a la consola de AWS ECR, click en su repositorio y verifique que allí está la imagen creada en la máquina. Tome un pantallazo de la consola para su **reporte**. El SHA256 de su imagen y su usuario deben ser visibles.
17. Copie la URI de la imagen (diferente a la URI del repositorio) para usarla más adelante.

Parte 3: creación del Clúster y despliegue en AWS ECS

1. Ahora crearemos el Clúster donde se ejecutarán los contenedores.
2. Vaya a la consola de AWS y en la esquina superior izquierda (Servicios) busque el servicio ECS o Elastic Container Service.
3. En el menú izquierdo seleccione Clústeres y click en crear Clúster.
4. Asigne un nombre al clúster. Como Infraestructura seleccione AWS Fargate, que es un servicio completamente administrado.
5. Aquí puede aparecer un mensaje de error. Puede ignorarlo y dar click en Crear.
6. Ahora debe regresar a la consola de ECS y esperar la creación del Clúster.
7. Una vez esté creado, de click en el nombre del Clúster. Debe aparecer en estado Activo y sin servicios ni tareas. Tome un pantallazo de la consola para su **reporte**. Su usuario de AWS debe ser visible.
8. En el menú izquierdo vaya a Definiciones de tareas y click en Crear una nueva definición de tarea (Crear definición de tarea).
9. Asigne un nombre a la definición de tarea. Como Tipo de lanzamiento seleccione el mismo AWS Fargate.
10. En roles de tarea seleccione LabRole (definido por AWSAcademy).
11. En roles de ejecución de tareas seleccione LabRole (definido por AWSAcademy).
12. En Contenedor:1 asigne un nombre al contenedor esencial, en URI de la imagen pegue la URI copiada anteriormente, y marque Sí en Contenedor esencial.
13. En Mapeos de puertos incluya el puerto 8001, las demás características puede dejarlas en sus valores autocompletados.



14. En Límites de recursos fije un límite estricto de memoria de 3 GB.
15. En Variable de entorno click en Agregar variable de entorno. En Clave indique PORT y en valor 8001. De esta manera podemos hacer referencia a ésta variable en el archivo run.sh (en la carpeta bankchurn-api).
16. Deje las demás opciones en sus valores por defecto y click en Crear.
17. Una vez se ha creado la definición de tarea tome un pantallazo de la consola con la definición de tarea y los contenedores asociados para su **reporte**. Su usuario de AWS debe ser visible.
18. Regrese ahora al Clúster. Seleccione Servicios y click en Crear servicio.
19. En la Configuración de implementación seleccione Servicio y en Familia seleccione la definición de tarea recién creada. La revisión debe actualizarse a la más reciente.
20. Asigne un nombre al servicio. Deje las demás opciones en su valor por defecto y click en Crear.
21. Al regresar a la ventana del Clúster, en el panel de Servicios debe observar el Servicio en ejecución.
22. El lanzamiento toma unos minutos. Cuando termine de click en el nombre del servicio.
23. Seleccione la pestaña Redes y click en el Grupo de seguridad. En la página del grupo de seguridad, seleccione Editar reglas de entrada y habilite el tráfico por el puerto 8001 desde toda dirección IPv4.
24. De regreso en la página del servicio, click en el tab Tareas para mostrar la Tarea en ejecución y el contenedor asociado. Tome un pantallazo de la consola con la tarea y el contenedor para su **reporte**. Su usuario de AWS debe ser visible.
25. Click en el ID de la tarea en ejecución. En la sección Configuración puede encontrar la IP pública. Copie esta IP y en un navegador visite la página IP:8001. Allí debe poder acceder a la API desplegada. Tome un pantallazo de la API en ejecución para su **reporte**. La IP pública debe ser visible. Note el nombre de la API en la página docs.
26. Para terminar edite el archivo config.py que encuentra en la carpeta bankchurn-api/app, y modifique el PROJECT_NAME incluyendo su nombre. Puede hacerlo directamente en la máquina o localmente y luego actualizando el repositorio en la máquina.
27. Con el archivo ya actualizado en la máquina, desde la carpeta raíz del repositorio construya nuevamente la imagen

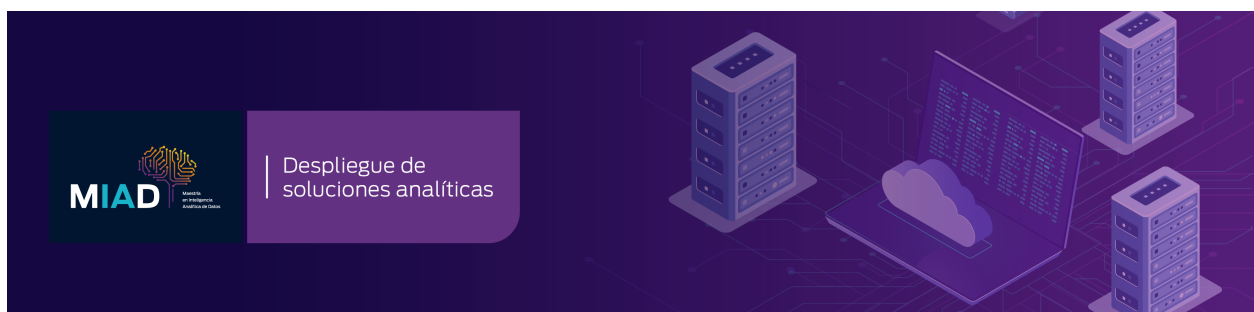
```
sudo docker build -t bankchurn-api:latest .
```

etiquétela con la URI del repositorio en ECS

```
sudo docker tag bankchurn-api:latest repoURI
```

y envíela al registro de AWS ECR con el comando

```
sudo docker push repoURI
```



reemplazando siempre repoURI con la URI del repositorio.

28. Ahora actualice el servicio en AWS ECS con el comando

```
aws ecs update-service --cluster namecluster --service servicename --task-  
definition taskdefname --force-new-deployment
```

donde namecluster se refiere al nombre del Clúster creado, servicename al nombre del Servicio, y taskdefname a la Definición de tarea creada.

29. Regrese a AWS ECS y observe en el Servicio que se actualiza el despliegue.

30. Verifique que se ha hecho la actualización yendo al Servicio y a la Tarea en ejecución. Observe que la IP pública ha cambiado.

31. Copie la IP pública y visite en el navegador la página IP:8001. Allí debe estar la nueva versión de su API. Navegue a docs y observe que el nombre se ha actualizado. Tome un pantallazo de la API en ejecución para su **reporte**. La IP pública debe ser visible.