

# Manual para el Empaquetamiento, Despliegue de API y Tablero Interactivo con Docker del Modelo Heart Attack

## Introducción

Este documento detalla los pasos necesarios para empaquetar el modelo de Heart Attack desarrollado en Python, incluyendo la configuración de un ambiente virtual, la creación, prueba y despliegue del paquete en una máquina virtual, asegurando un proceso escalable y replicable. Asimismo, explica cómo desplegar una API en una instancia EC2 utilizando Docker, abarcando desde la configuración del entorno hasta la construcción y ejecución de un contenedor. Finalmente, se incluyen las instrucciones para desplegar un tablero interactivo en Docker, conectado a la API previamente desplegada, lo que permite la interacción del usuario y la visualización de resultados, integrando todas las etapas en un flujo continuo y coherente.

## Empaquetamiento Modelo

### Parte 1: Configuración inicial de la máquina EC2 y Preparación del Ambiente

1. Lanzar una máquina virtual en AWS EC2:
2. Entra a la consola de AWS y selecciona la opción de instancias EC2.
3. Crea una instancia con las siguientes configuraciones:
4. Tipo de instancia: t2.small
5. Sistema operativo: Ubuntu Server
6. Almacenamiento: al menos 20 GB
7. Genera y descarga un archivo .pem para la clave de acceso. Este archivo será necesario para conectar tu máquina local a la instancia.
8. Subir archivos necesarios. Desde la terminal usa el siguiente comando para transferir el archivo package-src.zip a la máquina virtual. La llave y la ruta del archivo deben contener la ruta específica de donde estan almacenadas localmente:
  - `scp -i "ruta_a_llave.pem" "ruta_a_package-src.zip" ubuntu@IP:/home/ubuntu`
9. Conexión a la máquina virtual. Conéctate usando el siguiente comando:
  - `ssh -i "ruta_a_llave.pem" ubuntu@IP`
10. Actualizar paquetes e instalar herramientas. Ejecuta los siguientes comandos para actualizar el sistema operativo e instalar herramientas necesarias:
  - `sudo apt update`
  - `sudo apt install python3-pip zip unzip python3.12-venv`
11. Crear y activar ambiente virtual. Crea y activa un ambiente virtual llamado env-tox:
  - `python3 -m venv /home/ubuntu/env-tox`
  - `source /home/ubuntu/env-tox/bin/activate`
12. Descomprimir el paquete. Usa el siguiente comando para descomprimir el paquete:
  - `unzip package-src.zip`
13. Instalar herramientas necesarias. Instala tox y agrega su ruta al PATH:
  - `pip install tox`
  - `sudo apt-get install tox`
  - `export PATH=$PATH:/home/ubuntu/.local/bin`

## Parte 2: Ejecución del Empaquetamiento

14. Entrar a la carpeta del paquete.
  - `cd package-src`
15. Ejecutar pruebas de entrenamiento. puede generar un warning de unas variables que no se usan pero se puede ignorar:
  - `tox run -e train`
16. Ejecutar pruebas del paquete:
  - `tox run -e test_package`
17. Construir el paquete:
  - `python3 -m pip install --upgrade build`
  - `python3 -m build`
18. Transferir el archivo generado:
  - `mkdir /home/ubuntu/test`
  - `cp dist/model_heart-0.0.1-py3-none-any.whl /home/ubuntu/test`
19. Instalar el paquete:
  - `pip install model_heart-0.0.1-py3-none-any.whl`

## Parte 3: Pruebas con Datos Test

20. Subir archivos de prueba:
  - `scp -i "ruta_a_llave.pem" "ruta_a_archivo" ubuntu@IP:/home/ubuntu/test`
21. Ejecutar pruebas:
  - `python3 test-package.py`
22. Verificar paquetes instalados:
  - `pip freeze`

## Despliegue de la API con Contenedores Docker

### Parte 1: Instalación de Docker en la máquina EC2

23. Puede utilizar la misma instancia que utilizo para el empaquetamiento. Elimine versiones previas de Docker si las hubiera (este comando puede generar errores si no hay versiones previas):
  - `sudo apt-get remove docker docker-engine docker.io containerd runc`
1. Actualice el índice de paquetes:
  - `sudo apt-get update`
2. Instale dependencias necesarias para verificar certificados y administrar llaves GPG:
  - `sudo apt-get install ca-certificates curl gnupg`
  - `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`
3. Agregue el repositorio oficial de Docker:
  - `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
```

4. Actualice el índice de paquetes nuevamente:
  - `sudo apt-get update`
5. Instale Docker y sus componentes:
  - `sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin`
6. Verifique la instalación ejecutando el contenedor de prueba:
  - `sudo docker run hello-world`

## Parte 2: Despliegue de la API

7. Copie el archivo comprimido de la API desde su máquina local (abra otra terminal) a la máquina EC2:
  - `scp -i ubicacion_archivo/docker-api.zip ubuntu@IP:/home/ubuntu`
8. Descomprima el archivo en la máquina EC2:
  - `sudo apt install unzip`
  - `unzip docker-api.zip`
  - `cd docker-api`
9. Construya la imagen Docker a partir del archivo Dockerfile:
  - `sudo docker build -t heart-analytics:latest .`
10. Verifique que la imagen se haya creado listando las imágenes disponibles:
  - `sudo docker images`
11. Ejecute el contenedor de la API en el puerto 8001:
  - `sudo docker run -p 8001:8001 -it -e PORT=8001 heart-analytics`

## Parte 3: Configuración del grupo de seguridad

12. Permita el tráfico en el puerto 8001 desde cualquier IP.
13. En la consola de AWS EC2, selecciona la instancia.
14. Modifica las reglas del grupo de seguridad para agregar una regla de entrada:
15. Tipo: TCP
16. Puerto: 8001
17. Origen: Anywhere (0.0.0.0/0).
18. Ahora que se permita el tráfico en el puerto 8001, pruebe la API visitando la dirección pública de la máquina en el puerto 8001 desde su navegador o utilizando curl:
  - `curl http://ip_publica_instancia:8001`

## Despliegue del Tablero con Contenedores Docker

### Parte 1: Configuración inicial de la máquina EC2

24. Ahora cree una nueva instancia

25. Lanzar una máquina virtual en AWS EC2.
26. Entra a la consola de AWS y selecciona la opción de instancias EC2.
27. Crea una instancia con las siguientes configuraciones:
28. Tipo de instancia: t2.small
29. Sistema operativo: Ubuntu Server
30. Almacenamiento: al menos 20 GB
31. Configura para usar la llave.pem previamente creada para conectarse a esta maquina desde la terminal local.
32. Conéctese a la nueva máquina virtual desde su terminal local con el siguiente comando:
  - `ssh -i ruta_a_llave.pem ubuntu@IP`

## Parte 2: Instalación de Docker en la máquina EC2

1. Elimine versiones previas de Docker si las hubiera (puede ignorar errores si no hay versiones instaladas):
  - `sudo apt-get remove docker docker-engine docker.io containerd runc`
2. Actualice el índice de paquetes del sistema:
  - `sudo apt-get update`
3. Instale dependencias necesarias:
  - `sudo apt-get install ca-certificates curl gnupg`
  - `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`
4. Agregue el repositorio oficial de Docker al sistema:
  - `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`
5. Actualice nuevamente el índice de paquetes:
  - `sudo apt-get update`
6. Instale Docker y sus componentes:
  - `sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin`
7. Verifique la instalación ejecutando un contenedor de prueba:
  - `sudo docker run hello-world`

## Parte 3: Despliegue del tablero (Dash)

8. Copie el archivo comprimido del tablero desde su máquina local a la máquina EC2:
  - `scp -i ubicacion_archivo/docker-dash.zip ubuntu@IP:/home/ubuntu`
9. Descomprima el archivo en la máquina EC2:
  - `sudo apt install unzip`
  - `unzip docker-dash.zip`
  - `cd docker-dash`
10. Construya la imagen Docker para el tablero:

- `sudo docker build -t app:latest .`
11. Ejecute el contenedor del tablero conectándolo con la API previamente desplegada.  
Asegúrate de reemplazar IP con la IP pública de la máquina donde está ejecutándose la API:
    - `sudo docker run -p 8050:8050 -it -e PORT=8050 -e API_URL=IP app`

#### **Parte 4: Configuración del grupo de seguridad**

12. Permita el tráfico en el puerto 8050 desde cualquier IP.
13. En la consola de AWS EC2, selecciona la instancia.
14. Modifica las reglas del grupo de seguridad para agregar una regla de entrada:
15. Tipo: TCP
16. Puerto: 8050
17. Origen: Anywhere (0.0.0.0/0).

#### **Parte 5: Verificación del tablero**

18. Acceda al tablero desde su navegador web. Reemplaza IP\_PUBLIC con la dirección IP pública de la instancia. En un navegador, visite:
  - [http://IP\\_PUBLIC:8050](http://IP_PUBLIC:8050)
19. Pruebe la conexión con la API. Introduzca datos en el tablero y verifique que las solicitudes se transmiten correctamente a la API, devolviendo resultados visibles.