Maestría
en Inteligencia
Analítica de Datos

Integración y despliegue continuos

Instrucciones:

En este taller exploraremos el servicio de predicciones de modelos de analítica a través de una API.

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 usando un servicio PaaS

1. En esta actividad emplearemos Railway App, un servicio para el despliegue de aplicaciones. Cree una cuenta en `https://railway.app/dashboard`
   Podrá acceder a un plan de prueba (Trial) con 5 dólares de créditos, que deben ser suficientes para ésta y otras actividades del curso. Note que puede usar su cuenta de GitHub para crear la cuenta Railway. En caso de que no se le active el plan de prueba, tiene también la posibilidad de usar el plan gratuito (Free), que incluye 1 dólar de crédito mensual, que también debe ser suficiente para los talleres de clase.
2. Adjunto a este enunciado encontrará el archivo *repo-api-starter.zip*. Descomprima el archivo localmente en una carpeta, que llamaremos raíz, tal que allí queden dos carpetas: .circleci y bankchurn-api. En esta carpeta raíz cree un repositorio git, y conéctelo con un repositorio nuevo en GitHub. Al terminar debe tener en su carpeta raíz 3 carpetas: .git, .circleci y bankchurn-api.
3. Ahora vamos a lanzar una máquina desde la que podamos conectarnos a railway para desplegar nuestra aplicación.
4. Lance una máquina virtual en AWS EC2 con **Ubuntu 22.04** server. Se recomienda una instancia small 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.**
5. Conéctese a la instancia a través de SSH, como en las actividades anteriores. El comando tiene la siguiente forma


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

6. Actualice las ubicaciones de los paquetes con el siguiente comando:

```
sudo apt update
```

7. Instale pip para python3

```
sudo apt install python3-pip
```

8. Para emplear Railway necesitamos del manejador de paquetes npm. Instale npm en su versión estándar para Ubuntu

```
sudo apt install npm
```

9. Ahora queremos actualizar npm y node para contar con versiones recientes, necesarias para Railway. Para esto instale nvm

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh |
    bash

source ~/.bashrc
```

10. Revise la versión de nvm y las versiones remotas disponibles de node

```
nvm --version

nvm ls-remote
```

11. Instale la última versión LTS (long-term support). de node En el momento de escribir esta guía, corresponde a la 20.9

```
nvm install v20.9.0
```

12. Revise la versión de node y npm

```
node --version

npm --version
```

13. Ahora sí podemos instalar railway usando npm

```
npm i -g @railway/cli
```

14. Revise la versión de railway

```
railway --version
```

15. Loguéese a railway con el comando


```
railway login --browserless
```

el cual genera una URL. Copie esta URL en un navegador (Chrome, Firefox, etc.) donde tenga abierta su cuenta Railway. Verifique el acceso en el navegador y revise en la terminal de la máquina que quedó logueado.

16. En su navegador asegúrese de tener abiertas sus cuentas de GitHub y Railway.
17. En el navegador vaya a su cuenta de Railway y enlace su cuenta con Github: click en New project, seleccione Deploy from GitHub repo, click en Configure GitHub App.
18. Seleccione la cuenta de Github donde instalará la aplicación Railway. Permita acceso a todos los repositorios. Confirme usando su método preferido de verificación.
19. Con su cuenta ya configurada y enlazada con GitHub, cree un nuevo proyecto vacío. Click en New project, seleccione Empty project.
20. En el proyecto creado agregue un servicio vacío: click en Add a service, seleccione Empty service.
21. En el proyecto encontrará en la parte inferior una caja con un mensaje "Set up your project locally". Al hacer click allí encontrará un paso de conexión con la siguiente forma

    ```
    railway link -p idproyecto
    ```

    donde idproyecto es un código de identificación del proyecto. Copie este ID. Cierre la ventana de diálogo.
22. De regreso a la vista del proyecto, de click en el servicio creado. A la derecha se abre una ventana con varias pestañas. Click en la pestaña Settings. En General encontrará el Service ID (identificador del servicio). Copie el ID de este servicio.
23. De regreso en la terminal en la máquina virtual, clone el repositorio de GitHub que creó anteriormente

    ```
    git clone https://github.com/usuario/repo.git
    ```

    ajustando el nombre de usuario y repositorio.
24. De regreso en la terminal en la máquina virtual, clone el repositorio de Git que creó anteriormente.
25. Ingrese a la carpeta del repositorio, liste los contenidos de la carpeta

    ```
    ls -la
    ```

    y verifique tiene las tres carpetas mencionadas anteriormente: .git, .circleci y bankchurn-api.
26. Ingrese a la carpeta bankchurn-api

    ```
    cd bankchurn-api
    ```

27. Desde esta carpeta enlace el proyecto y el servicio de Railway con el comando

    ```
    railway link -p proyectoid
    ```

donde proyectoid es el ID de proyecto que copió de Railway. Aquí puede seleccionar el nombre del servicio usando las flechas arriba y abajo, y dar enter para seleccionar el servicio creado. Si solo hay un servicio en el proyecto, puede simplemente dar enter.

28. Con el enlace realizado, estamos listos para lanzar la aplicación en Railway

    ```
    railway up --detach
    ```

    Tome un pantallazo de la salida de este comando en inclúyala en su reporte.
29. Regrese al sitio de Railway, en su proyecto y servicio. De click en la pestaña Deployments, allí verá que el despliegue se está realizando. Cuando termine de desplegar, queda en estado Active. De click en viewlogs y **tome un pantallazo de los logs para su reporte**.
30. Regrese a la ventana anterior (click en la x de la esquina superior izquierda). Click en la pestaña Settings. En la sección Networking y Public Networking, click en Generate Domain. Al generar el dominio **tome un pantallazo del dominio generado para su reporte**.
31. Note que se ha creado una URL. **Copie esa URL en su reporte.**
32. De click en la URL o cópiela y péguela en su navegador. Note que se despliega la API, como se había realizado anteriormente en un máquina virtual.
33. Click en here para ir a los docs. Expanda la ruta POST /api/v1/predict y **tome un pantallazo donde sea visible la URL y las rutas de la API para su reporte.**

Parte 2: despliegue continuo usando CircleCI

Ahora queremos integrar la API desarrollada en un ambiente de despliegue continuo, en particular en la plataforma CircleCI.

1. Cree una cuenta en CircleCi `https://circleci.com/`.
2. Cree un nuevo proyecto y seleccione conectarlo con su cuenta de Github.
3. En CircleCI seleccione el repositorio creado para esta actividad. Note que aparece un mensaje donde se indica que se ha encontrado el archivo .circleci/config.yml. Éste es el archivo de configuración que usa CircleCI para realizar las operaciones de alistamiento del ambiente y despliegue.
4. Siguiendo las instrucciones de CircleCi, en la terminal de su máquina virtual genere una llave SSH con el comando, ajustando el correo

    ```
    ssh-keygen -t ed25519 -f ~/.ssh/project_key -C email@example.com
    ```

    Se le solicita una contraseña, déjela vacía dando enter dos veces.
5. En el paso anterior se generó un par de claves de nombre project_key, una pública y una privada. Para ver la clave pública use el comando

    ```
    cat ~/.ssh/project_key.pub
    ```

Esta clave empieza con ssh- y termina con su correo electrónico. Cópiela.

6. En su navegador vaya a Github y allí al repositorio creado en esta actividad. Vaya a Settings (menú superior derecho) y en el menú izquierdo busque la sección Security y seleccione Deploy keys.
7. Click en el botón Add Deploy Key para agregar su llave. Indique un título, pegue la llave en el campo Key y seleccione Allow write access para permitir que se pueda escribir en el repositorio con esta llave. Github le pedirá verificar la operación. **Tome un pantallazo de la interfaz de GitHub donde sea visible su usuario, repositorio y la llave agregada, e inclúyalo en su reporte.**
8. De regreso en su terminal de la máquina virtual, imprima la llave privada en pantalla con el comando

   ```
   cat ~/.ssh/project_key
   ```

   Esta llave empieza con —–BEGIN OPENSSH PRIVATE KEY—– y termina —–END OPENSSH PRIVATE KEY—–. Copie la llave completa, incluyendo estas líneas de inicio y fin.
9. Vuelva a la interfaz de CircleCI y pegue la llave privada en el campo Private SSH Key, que encontrará en Project, Settings, SSH Keys.
10. Asigne un nombre al proyecto en CircleCI y click en Create Project. Esto lo llevará a la sección Dashboard de CircleCI, donde podrá ver el proyecto.
11. Para conectar CircleCI con Railway vamos ahora a la página de Railway app y seleccionemos el proyecto que creamos (no el servicio, sino el proyecto). A la derecha encuentra el menú Settings, de click allí y en el menú de la izquierda seleccione Tokens. Asigne un nombre a su token (evite espacios) y de click en Create para crear el Token. El Token aparece y se le muestra solo una vez. Cópielo y guárdelo.
12. Regresemos ahora a CircleCI y en el Dashboard del proyecto seleccione Project Settings. En el menú izquierdo seleccione Environment variables. Click en Add Environment Variable. Como nombre asigne RAILWAY_TOKEN y como valor asigne el Token copiado de Railway. Click en Add Environment variable.
13. Con esta configuración, CircleCI podrá comunicarse con Railway para el despliegue de la aplicación.
14. Regrese al Dashboard de su proyecto en CircleCI, y en Railway regrese a la vista de su proyecto y el servicio creado.
15. Vaya ahora a la terminal de su máquina virtual e ingrese a la carpeta raíz de su repositorio, donde hay tres carpetas: .circleci, .git y bankchurn-api. Confirme esto con el comando

    ```
    ls -la
    ```

16. Ingrese a la carpeta .circleci

    ```
    cd .circleci
    ```

    y revise sus contenidos

```
ls
```

Revise los contenidos del archivo config.yml

```
cat config.yml
```

17. Localmente Ud cuenta con acceso al repositorio, abra el archivo config.yml. Allí encontrará una sección de jobs y en particular el job deploy_app_to_railway. Al final de este job se ejecutan varios comandos, en particular

    ```
    railway up --service=NombreDelServicio
    ```

    que permite enlazar con railway. Modifique este archivo para cambiar el nombre del servicio por el que creó anteriormente.
18. Una vez realice esta modificación, agregue los cambios al repositorio git y envíelos al repositorio remoto. Si usa la consola de git, esto corresponde a los comandos

    ```
    git add .

    git commit -m "Incluir nombre del servicio en railway"

    git push origin
    ```

19. Regrese ahora a su Dashboard en CircleCI y podrá ver que con el push al repositorio se ha lanzado la ejecución. Se muestran los pasos de ejecución, desde el lanzamiento del ambiente (Spin up environment) hasta el despliegue a Railway (Deploy to Railway App). **Tome un pantallazo de esta ejecución en el dashboard, dejando visibles su usuario, proyecto y los pasos de ejecución. Inclúyalo en su reporte**.
20. Visite ahora el servicio en Railway App y verá que se ha iniciado el despliegue allí. Click en View Logs para ver el paso a paso de la ejecución. En un momento la ejecución concluye y se muestra que el servidor está corriendo con Uvicorn. **Tome un pantallazo donde se muestren estos logs, donde se evidencie el nombre de su proyecto y servicio. Inclúyalo en su reporte**.
21. Recuerde que Ud creó una URL para este servicio. Si regresa a la página del servicio, la puede encontrar en el menú Settings, Networking, Public Networking. Visítela y verifique que la API efectivamente está corriendo.
22. Al terminar esta actividad puede terminar la ejecución del servicio en Railway. Seleccione el servicio y allí el menú Deployments. Seleccione el despliegue en verde (ejecución), click en los 3 puntos y seleccione Remove.