

Instalando graph-tool para Windows

Diego López Tamayo *

Contents

1. Instalar Docker	1
2. Configurar Docker	2
3. Descargar imagen de graph-tool	4
4. Lanzar sesión interactiva a través del Docker Container.	5
5. Notas finales	6
6. (Posible) Alternativa: Google Colab	8

Para utilizar graph-tool en Windows se requiere previamente:

- Tener instalada una distribución de Python. Puede ser [Anaconda](#) (recomendando) o instalar Python [directamente](#).
- En caso de ser Anaconda y ser la única distribución a utilizar, seleccionar “Add Anaconda to my PATH environment variable” en las opciones de instalación.
- Esta opción puede generar conflictos con otras versiones de Python previamente instaladas. Por tanto, sólo es recomendado si es la única distribución de Python a utilizar.

La versión de graph-tool para Windows funciona a través de un entorno virtual creado por Tiago Peixoto en un contenedor de **Docker**, este programa permite compartir aplicaciones entre plataformas (MacOS, Windows, Linux) a través de “contenedores”. [Este video](#) explica brevemente cómo funciona Docker. El contenedor incluye una versión de Python con graph-tool y algunos otros paquetes útiles de Python, como Matplotlib, Pandas, IPython y Jupyter.

Debemos primero instalar **Docker** y algunos componentes adicionales de **Windows Subsystem for Linux (WLS)**.

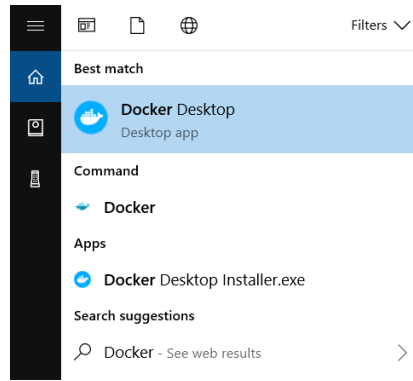
1. Instalar Docker

- **1.1** Entramos al siguiente enlace: [Docker for Windows](#) y presionamos el botón **Get Docker**
 - La versión estándar de Docker es compatible con Windows 10 Pro, Enterprise, o Education. De acuerdo al sitio oficial, el instalador que acabamos de descargar es el mismo para Windows Home, sin embargo **requiere habilitar las funciones de WSL 2**, mientras que en las otras versiones de Windows esto sólo es recomendado. En la barra de tareas buscar **“Información del sistema”** para saber la versión de Windows que tenemos.
- **1.2** Al entrar al instalador se ejecuta todo automáticamente y tarda unos 2 minutos en instalar. Al finalizar, se añade el ícono de Docker en nuestra barra de tareas. No se abre ninguna ventana adicional.

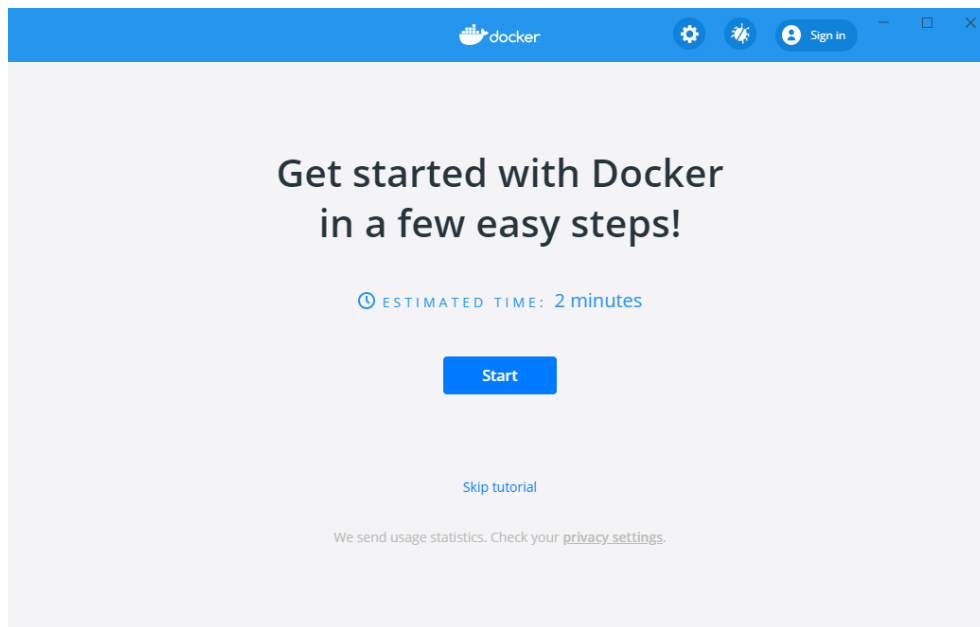
*El Colegio de México, diego.lopez@colmex.mx

2. Configurar Docker

- **2.1** En la barra de tareas buscamos la aplicación de Docker Desktop y le damos doble click para abrirla

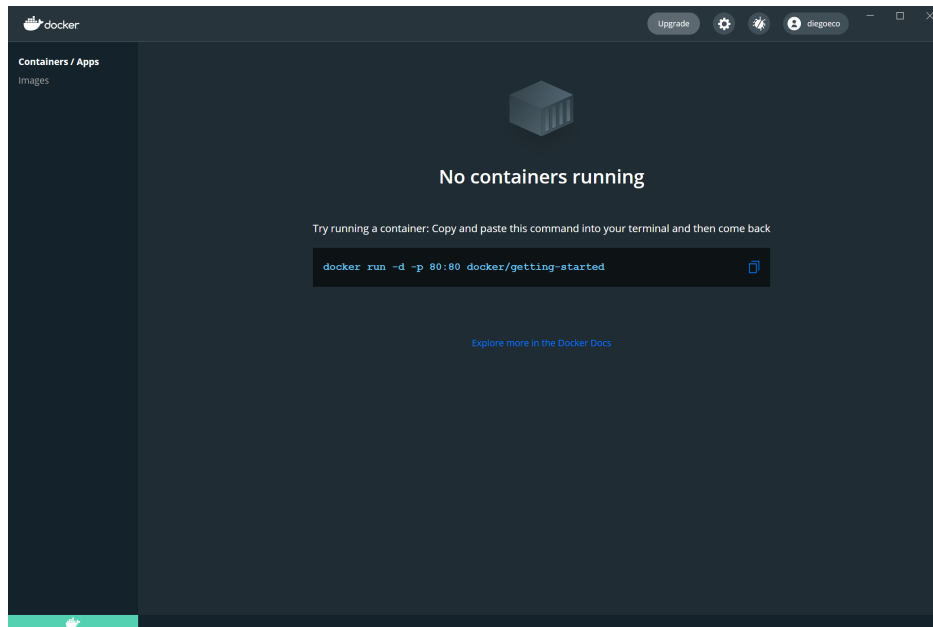


- **2.2** Nos dará la bienvenida y probablemente pida actualizar la versión, aceptamos las actualizaciones y llegaremos a una pantalla como esta: (Si aparece un aviso relacionado a que se necesita *WLS 2 backend* para finalizar la instalación, pasa al punto 2.5)

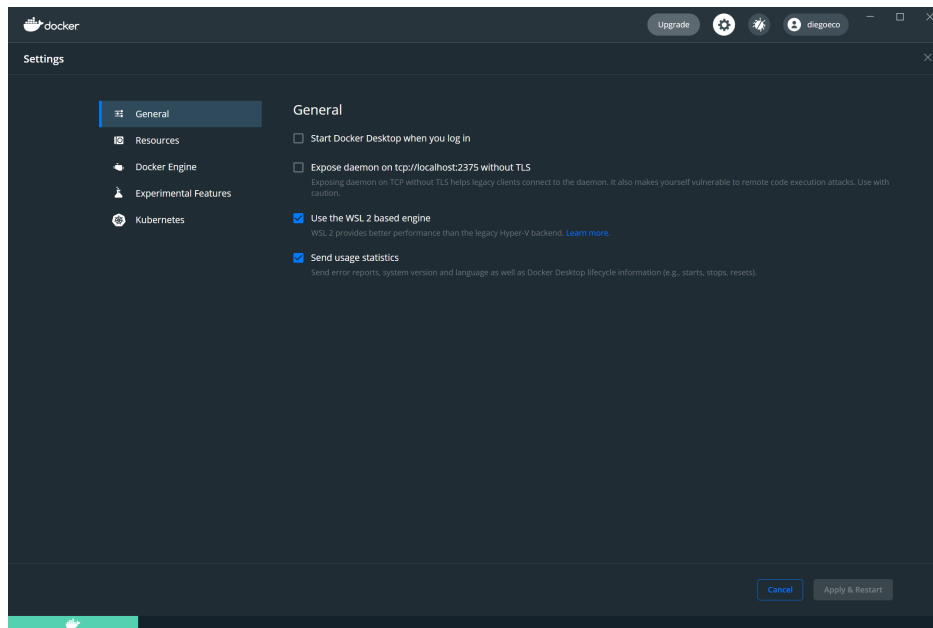


- **2.3** Podemos dar "Skip Tutorial" . El tutorial nos muestra cómo utilizar Docker desde la terminal con una serie de comandos y al final nos invita a abrir una cuenta en Docker Hub (Un servicio en línea similar a Git para contenedores). No es necesario seguir el tutorial.
- **2.4** Veremos una pantalla como la siguiente, donde muestra dos pestañas *Containers/Apps* que nos muestra las aplicaciones que se están ejecutando en tiempo real (debe estar vacía) y *Images* que nos muestra las "imágenes" o bloques de instrucciones para ejecutar un container. Aquí una descripción de

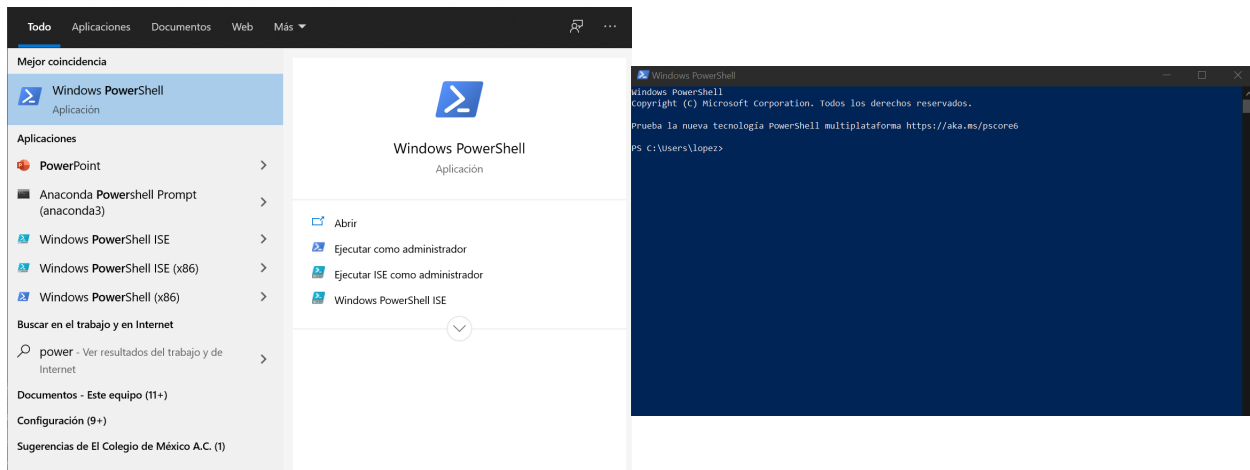
lo que es un *image*.



- **2.5** Debemos entrar a los ajustes y revisar que **esté activada la opción WSL 2 based engine**. Si no es así, debemos instalar/actualizar ajustes adicionales para utilizar componentes de Linux en Windows. Este paso puede ser un poco técnico:



- Debemos de estar en una sesión con permisos de Administrador. (Si eres el único usuario de tu computadora, seguro tienes permisos de Admin.)
- Descargamos el paquete de actualización del kernel de Linux de [este enlace](#) y seguimos los pasos de instalación sin modificar nada. Es una instalación muy rápida.
- Desde la barra de tareas, abrir **Windows Power Shell** o **Windows Power Shell (x86)** y mostrará una ventana como la siguiente:



- De forma secuencial y esperando a que se ejecute cada uno, copiar y pegar los siguientes comandos. Estar atento si requiere confirmar algún paso [Y/N] escribiendo Y y oprimiendo Enter. (Si hay algún problema, seguir los pasos 1 al 5 de [este tutorial](#))

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

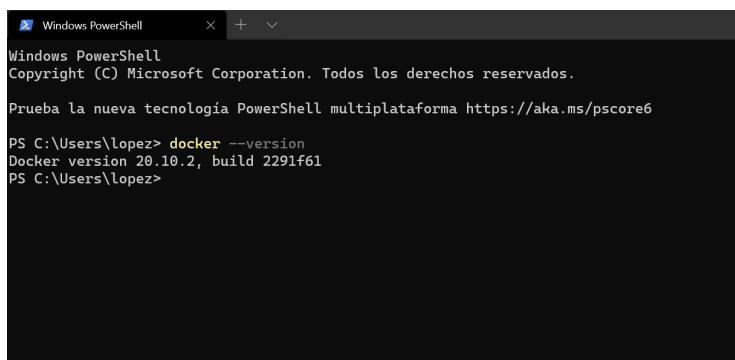
```
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

```
wsl --set-default-version 2
```

- Reiniciamos nuestro equipo y volvemos a entrar a los ajustes de Docker. La opción **WSL 2 based engine** debería estar activa ahora.

3. Descargar imagen de graph-tool

- La instalación de Docker creó un PATH en la terminal de Windows llamado **docker** . Para revisar que todo funcione bien abrimos de nuevo **Power Shell** y escribimos **docker --version** y debemos ver algo como lo siguiente:

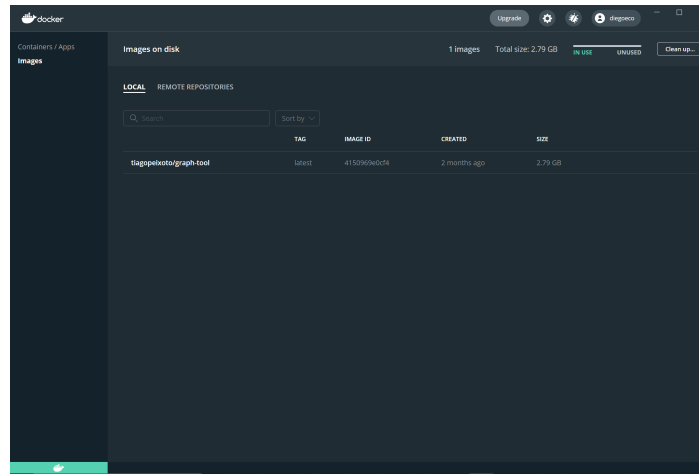


- Copiamos y pegamos el siguiente comando para obtener la imagen de graph-tool. Toma unos minutos en descargar y veremos el progreso de la descarga en la ventana de la terminal. El peso total es de 2.8 Gbs aproximadamente

<https://git.skewed.de/count0/graph-tool/-/wikis/installation-instructions#installing-using-docker>

4. Lanzar sesión interactiva a través del Docker Container.

- Si accedemos a la aplicación de Docker, en la pestaña Images debemos ver instalada localmente la imagen de graph-tool



- Ahora podemos crear una sesión de Python y/o un Jupyter Notebook desde la sesión creada por Docker. Para esto debemos de escribir en nuestra terminal de Power Shell lo siguiente:

- Para una sesión con IPython (versión interactiva de Python) copiar y pegar:

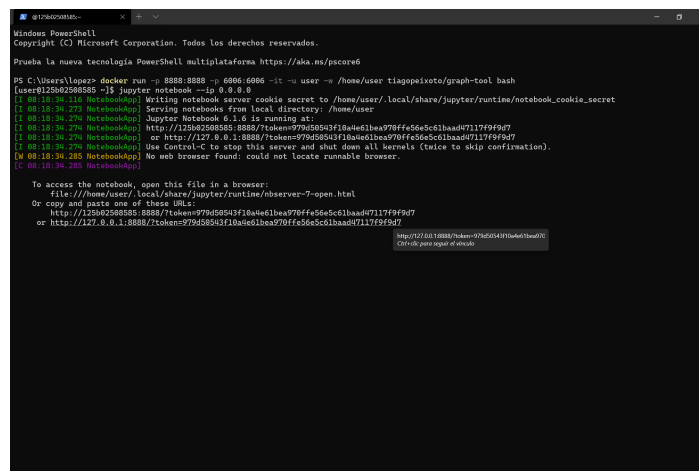
```
docker run -it -u user -w /home/user tiagopeixoto/graph-tool ipython
```

- Para lanzar una sesión de Jupyter Notebook necesitamos dos comandos. El primer comando inicia una sesión de shell interactiva. El segundo inicia Jupyter

```
docker run -p 8888:8888 -p 6006:6006 -it -u user -w /home/user tiagopeixoto/graph-tool bash
```

- Este segundo sigue de inmediato e inicia Jupyter. El código después de user@ puede cambiar cada que iniciemos una nueva sesión interactiva. Veremos entonces la siguiente pantalla:

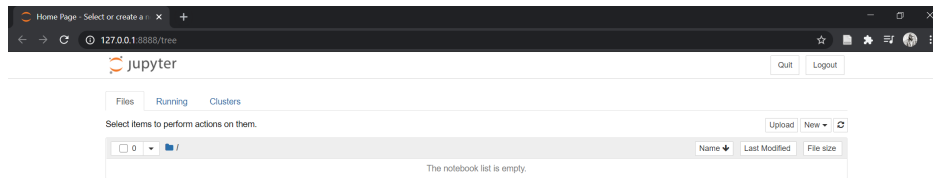
```
jupyter notebook --ip 0.0.0.0
```



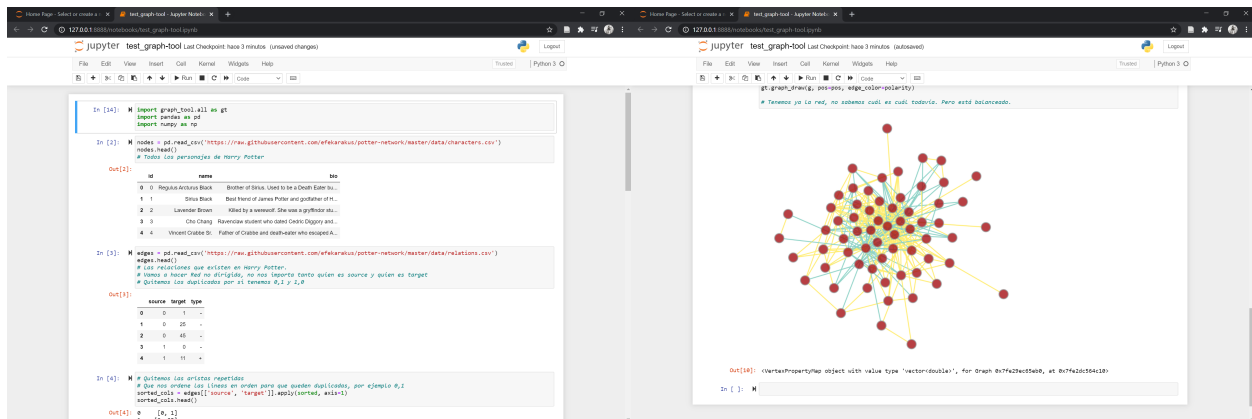
- Tenemos que dar Ctrl + Click en el último enlace para abrir Jupyter Notebook, como lo conocemos, en nuestro navegador.

5. Notas finales

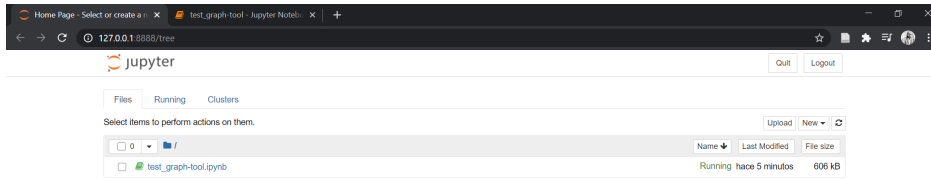
- Notarás que al lanzar un Jupyter Notebook, el directorio está vacío (a diferencia de cuando lo abrimos desde Anaconda, por ejemplo). Esto sucede porque se está creando desde un ambiente virtual de Python dentro de Docker.



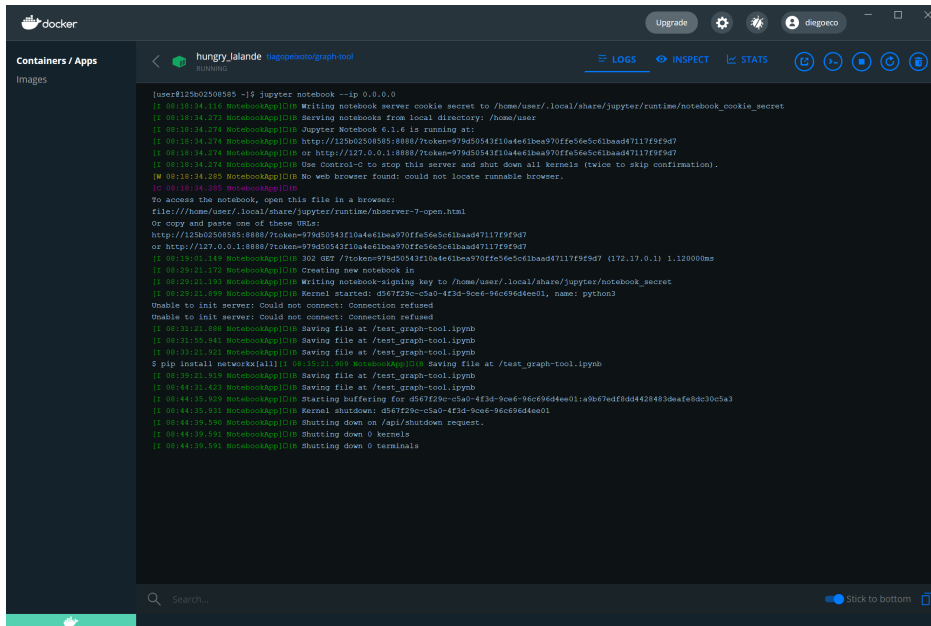
- Podemos crear un nuevo Notebook e importar las librerías principales (numpy, pandas, matplotlib y por supuesto graph-tool) pero **no sé cómo podemos instalar librerías adicionales en este entorno virtual de Docker. No funciona pip install ni conda install ¿Estamos limitados a las librerías del container de Docker?**



- También es importante que **descargues tu archivo de trabajo mientras el entorno de Docker sigue abierto** y guardes tu archivo .ipynb en algún lugar conocido. Recuerda que el directorio donde estamos trabajando es virtual y está dentro del contenedor de Docker, al cerrar la sesión se cierra ese ambiente.



- Al cerrar el Notebook se detiene el kernel y la terminal. Podemos observar en la aplicación de Docker que el Container sigue activo. Sin embargo al cerrar Docker o detener el container esa sesión queda como inactiva y **no sé como volver a acceder a ella**. Se debe repetir el *Paso 4. Lanzar sesión interactiva*.



6. (Posible) Alternativa: Google Colab

No he utilizado esta vía, pero parece ser una buena alternativa:

[Google Colaboratory](#) (también conocido como Colab) es un entorno de Jupyter Notebooks gratuito que se ejecuta en la nube y almacena los Notebooks en Google Drive. Dado que está basado en Linux, se puede instalar graph-tool en una sesión en la nube [siguiendo estas instrucciones](#).

En un Notebook nuevo de Colab copiar y pegar:

```
!echo "deb http://downloads.skewed.de/apt bionic main" >> /etc/apt/sources.list
!apt-key adv --keyserver keys.openpgp.org --recv-key 612DEFE798507F25
!apt-get update
!apt-get install python3-graph-tool python3-cairo python3-matplotlib
```

A partir de aquí podemos empezar a llamar todas las librerías que ya conocemos (Colab ya tiene instalado las básicas como Numpy, Pandas, Matplot, Seaborn) y trabajar en Notebook desde aquí para las tareas y proyecto. Si se requiere descargar el pdf está la opción de **Imprimir** ⇒ **Guardar como PDF** y funciona lo suficientemente bien.

