Alumno: Marvin Antonio Donis Gonzalez  
Carnet: 21005386

Product Development

Clase 2 – Docker

Docker instalado, verificando versión:



Verificando que la instalación es correcta corriendo un contenedor:

Texto

Descripción generada automáticamente

Extraer una imagen de un registro (asume que es “latest” o la más actualizada si no se especifica una versión):

Texto

Descripción generada automáticamente

Verificar imágenes instaladas en nuestra máquina:

Texto

Descripción generada automáticamente

Corriendo un segundo contenedor:

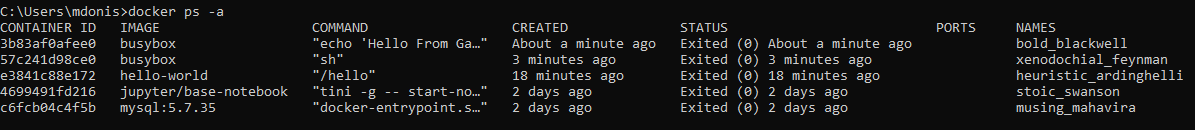


Contenedores que están corriendo:

Texto

Descripción generada automáticamente

Listado de contenedores que han sido ejecutados:



Correr y entrar a un contenedor e interactuar con el shell “sh”:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Eliminar contenedores:

Texto

Descripción generada automáticamente

Para eliminar todos los contenedores se puede hacer **Docker container prune**

Docker y Python

Instalar Jupyter

Texto

Descripción generada automáticamente

Ejecutar Jupyter por medio del mapeo de puertos máquina:contenedor:

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Verificando contenedor y mapeo de puertos:



Crear una red para conectar Jupyter y MySQL:

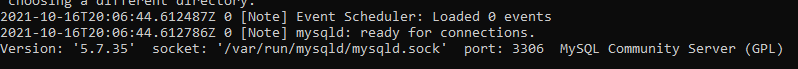


Texto

Descripción generada automáticamente

Ejecutar un contenedor de MySQL en la red previamente creada e interactivamente:





Ahora nos conectamos de nuevo, esta vez especificando el puerto para tener acceso a la base de datos desde nuestra máquina:



Ahora se ejecutará Jupyter pero en la misma red del contenedor de la base de datos para poder hacer una conexión:

Texto

Descripción generada automáticamente

Instalación de librerías de Python ingresando al contenedor que contiene Jupyter:

Texto

Descripción generada automáticamente

Ubicar ip del host: 172.18.0.2 y conectarnos a la base de datos vía Jupyter:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Docker-Compose y Python

Verificando instalación:



Se crea archivo docker-compose.yml, se almacena en una carpeta y se ejecuta el comando docker-compose up:

Texto

Descripción generada automáticamente

Luego se instalan las librerías necesarias en Python y se realiza la conexión con la base de datos:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Como paso final se detienen los contenedores:

Texto

Descripción generada automáticamente