**A picture containing indoor, sitting, food, table

Description automatically generated**

TAREA 1

Docker

Maestría en Data Science

Product Development

Antonio Everardo Navas CONTRERAS, 14003163

Guatemala, 30 de octubre de 2021

**DOCKER**

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

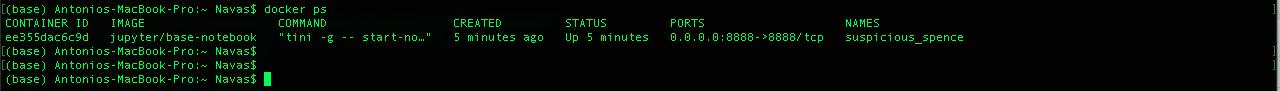
Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated



Text

Description automatically generated

Graphical user interface

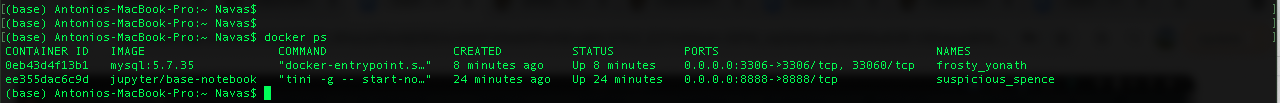
Description automatically generated

Text

Description automatically generated

A picture containing text, monitor, indoor, screenshot

Description automatically generated



Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Application

Description automatically generated with low confidence

**DOCKER-COMPOSE**

A picture containing text, electronics, screenshot

Description automatically generated

Text

Description automatically generated

A picture containing graphical user interface

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Application

Description automatically generated with medium confidence

**YML FILE**

version: '3.7'

services:

db:

image: mysql:5.7.35

volumes:

- db\_data:/var/lib/mysql

restart: always

ports:

- 3306:3306

environment:

MYSQL\_ROOT\_PASSWORD: test123

MYSQL\_DATABASE: test

MYSQL\_USER: test

MYSQL\_PASSWORD: test123

jupyter:

image: jupyter/base-notebook

ports:

- 8888:8888

volumes:

db\_data:

**COMMANDS**

$ docker <command> --help # Nos da ayuda de los comandos

$ docker run hello-world:latest # Corre y si no existe el contenedor lo descarga

$ docker pull busybox # Solamente descarga el contenedor

$ docker images # Ver imágenes

$ docker run busybox echo ‘Hello World’ # Envía parámetros al contenedor para ejecutarlos

$ docker ps # Ver contenedores corriendo

$ docker ps -a # Ver contenedores detenidos, Recomendable borrarlos

$ docker run -it busybox sh # Corre el contenedor en modo interactivo

$ docker rm <CONTAINER ID or NAME> # Borra los contenedor detenido, varios separados por espacio

$ docker container prune # Borra todos los contenedores detenidos

$ docker pull jupyter/base-notebook # Imagen de python jupyter

$ docker run -p 8888:8888 jupyter/base-notebook # Para correr jupyter, nos da un link con token

$ docker run -p 8888:8888 -d jupyter/base-notebook # Corre el contenedor “detached” sin mandarnos a consola

$ docker stop <CONTAINER ID or NAME> # Para detener un contenedor

$ docker network create --driver bridge my\_test\_network # Crear una red para comunicación entre containers

$ docker run -it --network my\_test\_network -p 3306:3306 -e "MYSQL\_ROOT\_PASSWORD=root123" -e "MYSQL\_DATABASE=test" -e "MYSQL\_USER=test" -e "MYSQL\_PASSWORD=test123" mysql:5.7.35 # Correr contenedor de MySQL y agregarlo a la red que se creó

$ docker run --network my\_test\_network -p 8888:8888 jupyter/base-notebook # Correr contenedor de jupyter y agregarlo a la red que se creó

$ docker network inspect my\_test\_network # Inspeccionar la red, para ver IP de los contenedores

$ docker exec -it <container NAME> sh # Para entrar al sh del contenedor

$ pip install mysql-connector-python # Para instalar conector de SQL desde jupyter python

$ pip install pandas # Para instalar pandas desde jupyter python

# Para ejecutar desde el jupyter notebook

from sqlalchemy import create\_engine

source = create\_engine('mysql+mysqlconnector://test:test123@172.20.0.2/test')

motor\_DB + driver :// user:password@IP\_Address / nombre\_schema

import pandas as pd

pd.read\_sql('select now()', con=source)

https://docs.docker.com/compose/install/

Después de correr el docker-compose

!pip install mysql-connector-python # Para instalar conector de SQL desde jupyter python

!pip install pandas # Para instalar pandas desde jupyter python

Reiniciar el kernel de jupyter

from sqlalchemy import create\_engine

source = create\_engine('mysql+mysqlconnector://test:test123@db/test')

motor\_DB + driver :// user:password@DNS\_Name\_by\_compose / nombre\_schema

import pandas as pd

pd.read\_sql(‘select now()’, con=source)