A step-by-step guide to running a scraper with Docker and Redis

To get the system up and running please follow the steps I took. I included the errors I have encountered and my solutions to them, however if you do not encounter the same error do not use the codes in their section.

## Creating ubuntu.

1024mb ram.

10 gb hard disk, VDI. Dynamically Allocated.

Ubuntu version 20.04.3

Normal Installation. (nis – root (username and password I used))

## Ubuntu console commands

Ubuntu came with Python 3.8.10 installed.

Sudo apt update

Followed this tutorial for the scraper code: <https://www.edureka.co/blog/web-scraping-with-python/>

Indented block errors.

Sudo apt-get update (trying to install beautifulsoup4)

#### Python 3 installation

Sudo apt install python3

#### Python 3 install error

Encountered error during python 3 install <https://unix.stackexchange.com/questions/374748/ubuntu-update-error-waiting-for-unattended-upgr-to-exit>

sudo dpkg-reconfigure -plow unattended-upgrades

sudo dpkg --configure -a

sudo apt update && sudo apt -f install && sudo apt full-upgrade

sudo dpkg-reconfigure -plow unattended-upgrades

#### Python 3 installation after solving the error

Sudo apt install python3 (worked)

Sudo apt install python3-pip

sudo pip install beautifulsoup4 (<https://subscription.packtpub.com/book/web-development/9781783289554/1/ch01lvl1sec08/installing-beautiful-soup>)

Now that we have installed everything we need for the basic version we can run the “scraper.py” on our machine and it will print the transaction with the highest value to the console every minute.

## Installing Docker

sudo apt-get update

sudo apt install docker.io

sudo docker --version

sudo systemctl status docker

sudo docker run hello-world

sudo docker images

sudo docker run docker/whalesay cowsay moby dick

sudo docker run -d -p 8080:5000 jcdemo/flaskapp

#### Creating the Images for MongoDB and Redis

!!! Disclaimer, even though we’re creating a Redis Image as well, our scraper code is not using it. Thus you can skip the parts for Redis which are colored red in this section. But I have documented it here regardless.

Sudo docker pull mongo

Sudo docker pull redis

Mkdir mongodb-docker

Mkdir redis-docker

cd mongo-docker

Sudo docker run -d -p 27017:27017 --network="host" -v ~/mongodb-docker:/data/db --name mongodb mongo:latest

You’d need to exit by typing exit in order to run the following commands:

Cd redis-docker

sudo docker run -d --network="host" --name redis redis

After running the last command (even if it’s only the mongodb command) navigate back to the desktop on the terminal.

Cd Desktop/

Then run our scraper that you downloaded to the machine by typing the following command:

Python3 scraper-docker.py

The program will run continuously and store the highest transaction of each minute into MongoDB.

## Afterword

Sorry for the mess. I had to rush this project since everything else was too overwhelming for me. I decided to scrap the individual parts for MongoDB and Redis. I also decided not to use Redis since my code was already finding the transaction with the highest value without the need to cache them. So, if I wanted to use Redis I would’ve to write the scraper from scratch, thus I decided to just remove it from the project even if costs me points from my final score in this class.

Serhat Albayrak

R0739124