Dokumentation Sever-Projekt:

**Basic Server Setup**

**Create Server**

Went on to DigitalOcean and created droplet with Ubuntu 16.04. preinstalled; Server in Germany (This is relevant for the programme). It has Python and a CronDaemon preinstalled.

**Access Server with PuTTY (For Windows)**

Download PuTTYGen, create an SSH key-pair, upload it to DigitalOcean;

Conncet to your server with PuTTY as “root” user

**Access Server for Mac/Linux**

Via command line? In any case as root user.

**Create user and configure**

Root user has a lot of administrative privileges, which can be destructive if you make mistakes. That’s why we create a user. When wanting root privileges for the user we prefix *sudo* (“substitute user do”) to a “root” command. ***username*** now stands for you chosen username

*# adduser* ***username***

Add password. Then give user ***username*** sudo-privileges:

*# usermod -aG sudo* ***username***

Now we want to be able to log in as ***username*** with SSH keys. Switch from root to ***username***

*# su –* ***username***

Make SSH-directory to save the SSH-key (like telling the system which password to accept…)

*$ mkdir ~/.ssh*

*$ chmod 700 ~/.ssh*

Nano is a text editor in Ubuntu, which can be used well in the terminal:

*$ nano ~/.ssh/authorized\_keys*

Past SSH key in the opened file, exit with *CTRL-x,* save changes with *y* and confirm with *ENTER*

Restrict permission of the *authorized\_keys* file:

*$ chmod 600 ~/.ssh/authorized\_keys*

**Set up LAMP**

**Install Pip3**

Basically, all installable programmes on ubuntu are listed in the Advanced Package Tool (APT). Before installing a new programme, we should always update the list of installable programmes:

*$ sudo apt-get update*

Now install the python3 module installer pip3, again with APT:

*$ sudo apt-get install python3-pip*

**Install MySQL**

MySQL is the language by which we can manipulate databases. Install with APT:

*$ sudo apt-get install mysql-server*

Set a password and remember it! We need it both to manually open MySQL and for the scraping programme.

*$ mysql\_secure\_installation*

For Python to connect to MySQL we need the modules pymysql and SQLalchemy:

*$ sudo pip3 install pymysql*

*$ sudo pip3 install SQLalchemy*

Other modules we need:

*$ sudo pip3 install pandas*

*$ sudo pip3 install bs4*

**Create MySQL Database and table**

***Note that if you are not in windows, then every mysql command must be in capitalised letters (except where you reference a name like “financialdata” or “option\_chains”.***

Log into MySQL and enter password:

*$ mysql -u root -p*

Change the database name if you want, but it will be easier if we stick to the same name:

*mysql > create database financialdata;*

*mysql > use financialdata;*

Same applies to the table name. Tables are the primary format for data – think of them like an excel sheet. While creating the table we specify exactly which variables and which format we expect:

*mysql > create table option\_chains(Database\_ID varchar(31) not null primary key, Contract\_Name varchar(21) not null, Last\_Day\_Traded date null, Strike float(6,2) not null, Last\_Price float(6,2) null, Bid float(6,2) null, Ask float(6,2) null, Volume int null, Open\_Interest int null, Implied\_Volatility float(6,2) null, Price\_of\_Underlying float(6,2) not null, Date date not null, Maturity\_Date date not null, Ticker\_Underlying varchar(5) not null, Option\_Type varchar(4) not null);*

Exit mysql:

mysql > exit;

**Install Apache 2**

Apache makes sure that you can access the server via a webpage and then handles your requests.

*$ sudo apt-get install apache2*

**Create Scraper:**

**Make python file**

We want the file in */usr/local/bin*, so let’s move there:

*$ cd /*

*$ cd usr*

*$ cd local*

*$ cd bin*

Create python file:

*$ sudo nano option\_scrape.py*

Copy the content of the appended python file and paste it by right-clicking into the nano editor. Scroll down to the last line of the file and insert (change) the tickers you want to scrape. The format is a capitalized string in quotation marks, and each ticker separated by a comma (see appended python file). Then in the same lane replace the placeholder YOUR\_PASSWORD\_FOR\_MYSQL with your actual password for MySQL. Then exit with *CTRL-x,* save changes with *y* and confirm with *ENTER.*

Next we change the permissions of the file, so that an external programme can execute the script:

*$ sudo chmod +x option\_scrape.py*

**Set up Cron Daemon**

The server came with a Cron Daemon preinstalled. If you give the Cron Daemon a “Crontab”, then it will execute it regularly as specified. It is automatically saved to the correct place, so let’s open it up where we are and, if given the option, select or favourite editor nano:

$ crontab -e

Insert this single line into the editor:

|  |
| --- |
| 00 05 \* \* 1-5 /usr/local/bin/option\_scraper |

This line tells cron to execute every weekday of the year at 5am.

**--DONE—**

Now the file should be executed regularly and insert data into our database.

For testing purposes, you can execute the python3 file manually by entering:

$ cd /

$ python3 /usr/local/bin/option\_scraper.py

And you can see whether there is anything in the MySQL table by entering:

$ mysql -u root -p

mysql > use financialdata

mysql > show \* in option\_chains

**HAVEN’T DONE THIS YET – only seems to complicate things for the time being**

**Set up a basic firewall**

Only allow applications registered on UFW to access the server:

*$ sudo ufw app list*

Make sure ufw allows our openSSH so that we can use PuTTY to log in:

*$ sudo ufw allow OpenSSH*

*$ sudo ufw enable*

*$ sudo ufw status*

The last command shows all allowed applications. Any additional application you install, will have to be added to the ufw list.