# Instalacja środowiska na zajęcia Apache Spark

Z tego co mi wiadomo, używają Państwo obrazu Ubuntu ze strony: <http://repo.kodolamacz.pl/> login/hasło: kodolamacz/kodolamacz

Środowisko przygotowane przez prowadzących Bootcampu jest w zasadzie wystarczające do przeprowadzenia zajęć (z lekką modyfikacją). Jednakże przezorny zawsze ubezpieczony dostarczyłem kilka alternatywnych sposobów na przygotowanie sobie środowiska na zajęcia ze Spark.

Poniżej zamieszczam równoważne sposoby. Wybieramy wedle uznania i łatwości ich osiągnięcia. Jeśli wybierzecie opcję najszybszą: 1a lub 1b to polecam dla sportu zrobić na swoich VB instalację Docker’a (2). Pokażę Wam jak można wykorzystać tą technologię do nauki big-data itp.

**Spis treści**

[**Instalacja środowiska na zajęcia Apache Spark**](#_xjbmgarxt8vr) **1**

[1 - Używasz VB Ubuntu-16.04-DS.ova dostarczonego przez poprzednich prowadzących](#_p6gyx2ihasyf) 1

[a - Zmiana wersji Python w Anaconda](#_kd1efj39e3dq) 1

[b - Instalacja Spark zgodnego z Python 3.6](#_avt09p6bfvlk) 1

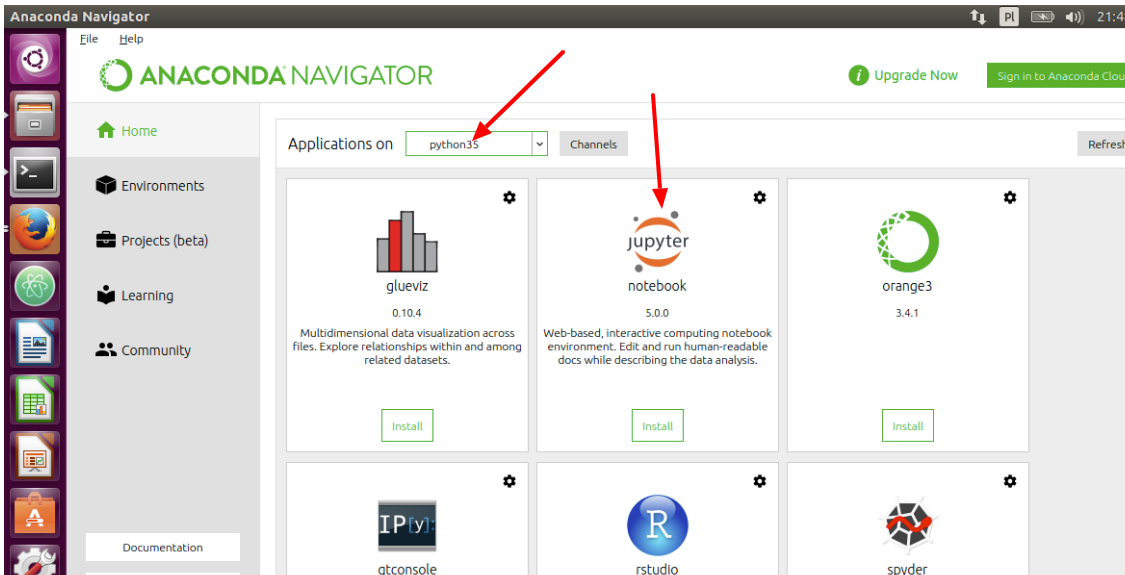
[2 - Wykorzystanie technologii Docker’owej](#_12vweg8t51fv) 1

[3 - Użycie VB z którego korzystam na swoich szkoleniach](#_enu7mzfofzii) 2

## 1 - Używasz VB Ubuntu-16.04-DS.ova dostarczonego przez poprzednich prowadzących

Tutaj w zasadzie jesteśmy w domu. Praktycznie wszystko jest gotowe z tym mankamentem, że na środowisku zainstalowany jest Python 3.6 i nieszczęśliwie ściągnięty Spark 2.1.0. Niestety akurat ta wersje się ze sobą gryzą. Można albo:

### a - Zmiana wersji Python w Anaconda

1. W anaconda wybieramy Applications on: Python 35 a następnie instalujemy Jupyter. KONIEC :)  
   

### b - Instalacja Spark zgodnego z Python 3.6

1. Ściągamy ze strony: <http://ftp.man.poznan.pl/apache/spark/spark-2.3.2/spark-2.3.2-bin-hadoop2.7.tgz>
2. Rozpakowujemy do ~/spark-2.3.2-bin-hadoop2.7
3. Usuwamy linka sympolicznego i wskazujemy na nowy (w terminalu):
   1. cd ~
   2. rm spark
   3. ln -s spark-2.3.2-bin-hadoop2.7 spark
4. Zmieniamy ostatnią linijkę w pliku ~/.bashrc
   1. nano ~/.bashrc
   2. linijkę:   
      export PYTHONPATH=$SPARK\_HOME/python:$SPARK\_HOME/python/lib/py4j-0.10.4-src.zip:$PYTHONPATH
   3. zamieniamy na:  
      export PYTHONPATH=$SPARK\_HOME/python:$SPARK\_HOME/python/lib/py4j-0.10.**7**-src.zip:$PYTHONPATH
5. Restart maszyny
6. KONIEC :)

## 2 - Wykorzystanie technologii Docker’owej

Na wszelki wypadek gdyby środowisko Anacondy nie podołało albo ktoś nie korzystał z dostarczonego VB zdecydowałem się postawić środowisko zapasowe na [Docker](https://www.docker.com/)’ach. Dlaczego? Opowiem na zajęciach.

To co trzeba zrobić zamieściłem poniżej.

|  |
| --- |
| Jeśli ktoś nie używa VM dostarczonej przez URL wyżej a:   1. Posiada Linuxa: poniższą procedurę może wykonać na swoim własnym linuxie. 2. Jeżeli ktoś posiada Maca lub Windowsa - niestety nie jestem w stanie dostarczyć wsparcia (nie dysponuję komputerem z tymi systemami). Windows 10 PRO posiada natywne wsparcie dla Docker’a więc prawdopodobnie instalacja Dockera jest standardowa dla tego systemu, Dla wersji Windows poniżej 10 PRO sugeruję zatem użycie VM z linku powyżej. Jeśli jednak ktoś chce powalczyć to mogę dać namiar na tutorial jak samodzielnie zainstalować Jupyter’a wraz ze Sparkiem: <https://sigdelta.com/blog/how-to-install-pyspark-locally/> (na dole artykułu jest fragment programu, który powinien się uruchomić by wiedzieć, że wszystko jest OK. 3. W razie problemów z Windowsami/Mac’ami polecałbym jednak użycie dostarczonej VM (jak importować jest [TUTAJ](https://www.maketecheasier.com/import-export-ova-files-in-virtualbox/)). Po prostu chciałbym uniknąć marnowania czasu na walkę z konfiguracją. Gdyż nie zawsze się ją wygra, a pozostali tracą czas. |

Prosiłbym wykonać poniższe instrukcje w domu ze względu na to, że trwają jakiś czas oraz wymagają ściągnięcia ok. 5GB danych. Na zajęciach WiFi może przy wielu użytkownikach nie podołać przy takim transferze.

Logujemy się na maszynę z Ubuntu i otwieramy terminal:

|  |
| --- |
| Uwaga! Jeśli podczas wykonania pierwszego polecenia z **sudo** wystąpi błąd o braku możliwości założenia blokady na pliku **/var/lib/dpkg/lock** należy wykonać komendę: **sudo rm -f /var/lib/dpkg/lock** Możliwe, że ten zabieg będzie trzeba powtórzyć 2-3 razy. W moim przypadku za drugim razem udaje się wykonać pierwszą instrukcję zamieszczoną poniżej. Widocznie jest to konsekwencja popełnienia jakiegoś błędu podczas przygotowania obrazu VirtualBox.   1. Podczas wykonywania komend zawsze odpowiadamy na pytania twierdząco T/n -> T |

**W terminalu:**

sudo apt-get install apt-transport-https ca-certificates curl software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"

sudo apt-get update

sudo apt install docker.io

(tekst poniżej to **jedna** komenda, proszę skopiować całość, nawet jeśli zajmuje dwie linijki)

sudo curl -L https://github.com/docker/compose/releases/download/1.21.2/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

sudo groupadd docker

sudo usermod -aG docker $USER

Restart komputera. Od tej chwili mają Państwo dostępne komendy **docker** oraz **docker-compose**.

Po restarcie wykonujemy ponownie w **terminalu**:

cd ~

git clone https://github.com/cackoarek/docker-big-data.git

cd docker-big-data/jupyter-spark/

docker-compose build

Tu nastąpi faza ściągania środowiska. Może potrwać kilkanaście minut (zależnie od prędkości Internetu).

## 3 - Użycie VB z którego korzystam na swoich szkoleniach

Zamiast VB z którego korzystacie na swoich zajęciach możecie ściągnąć VB, który jest możliwy do ściągnięcia pod linkiem: [LINK](https://mega.nz/" \l "!oEpiCSZQ!JTUARDpidbSU_TGuq_iwxje1m5ANykN7Jn2pss8CbLM)

login/pass: szkolenie/11

Po starcie wykonujemy w **terminalu**:

cd ~

git clone https://github.com/cackoarek/docker-big-data.git

cd docker-big-data/jupyter-spark/

docker-compose build

Dziękuję :)

pytania: cackoarek@gmail.com

Poniższą instrukcję wykonamy już na zajęciach. Zapisuję ją na potrzeby udokumentowania instrukcji:

1. Terminal: anaconda-navigator
2. W Anaconda:
   1. Application on: python 3.5 -> Jupyter -> install
   2. ~~Zakładka Enviroments -> Python 3.5 -> Open terminal -> pip install findspark~~

Jeśli chcemy używać wersji Python 3.6 należy ściągnąć nowszą wersję spark niż 2.1.0.

1. Ściągamy ze strony: <http://ftp.man.poznan.pl/apache/spark/spark-2.3.2/spark-2.3.2-bin-hadoop2.7.tgz>
2. Rozpakowujemy do ~/spark-2.3.2-bin-hadoop2.7
3. Usuwamy linka sympolicznego i wskazujemy na nowy:
   1. cd ~
   2. rm spark
   3. ln -s spark-2.3.2-bin-hadoop2.7 spark
4. Zmieniamy ostatnią linijkę w pliku ~/.bashrc
   1. nano ~/.bashrc
   2. linijkę:   
      export PYTHONPATH=$SPARK\_HOME/python:$SPARK\_HOME/python/lib/py4j-0.10.4-src.zip:$PYTHONPATH
   3. zamieniamy na:  
      export PYTHONPATH=$SPARK\_HOME/python:$SPARK\_HOME/python/lib/py4j-0.10.**7**-src.zip:$PYTHONPATH
5. Restart maszyny