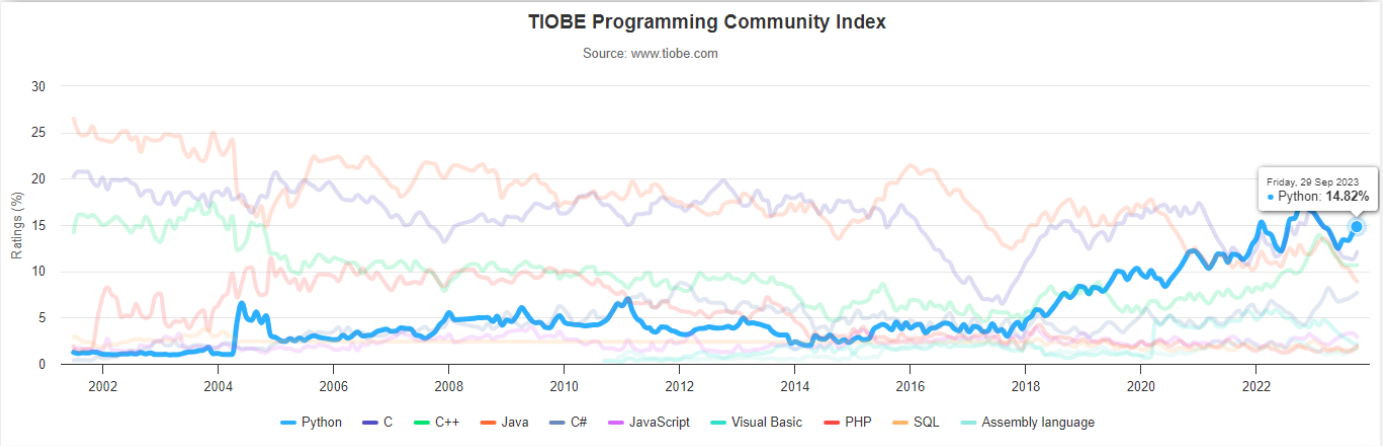
Python

Wprowadzenie

Python jest językiem nie tylko bardzo popularnym. W październiku 2023 zarówno wg.

- PYPL PopularitY of Programming Language `https://pypl.github.io/PYPL.html` jak i
- TIOBE Index `https://www.tiobe.com/tiobe-index/`

zajmował on pierwsze miejsce wśród języków programowania. Co więcej, dynamika wzrostu począwszy od 2018 roku jest największa wśród wszystkich pozostałych języków programowania.



Python jest przede wszystkim językiem bardzo uniwersalnym, a jego rola w ogólnie rozumianej analizie i przetwarzaniu danych z roku na rok jest coraz większa.

Nic więc dziwnego, że świat Big Data, który przez lata był zarezerwowany dla Javy oraz języków, które dają się kompilować do kodu bajtowego Javy, otwiera się na Pythona (patrz Spark czy Flink).

W ramach warsztatu postaramy się dokonać krótkiego przeglądu niektórych typów i bibliotek, które wykorzystywane są w Pythonie do przetwarzania danych.

Dane

Aby przetwarzać jakieś dane, przydałoby się je mieć. Python zawiera niezliczone biblioteki, przydatne praktycznie w każdym przypadku. Dotyczy to również dostępu do przykładowych danych.

Przykładowe biblioteki, w których można takie dane znaleźć to `PyDataset`, `seaborn` czy `sklearn`. Te dwie ostatnie główny cel mają oczywiście inny (wizualizacja danych, ML).

Przygotowanie środowiska

Treść zadań realizowanych w ramach warsztatów zawarta jest w notatniku Jupyter. Aby móc je wykonać powinniśmy mieć dostęp do środowiska Jupter Notebook lub JupterLab (wariant preferowany).

Zainstalować to środowisko można na wiele sposobów.

- Lokalnie: `https://jupyter.org/install`
- W kontenerze dockerowym, dla przykładu:

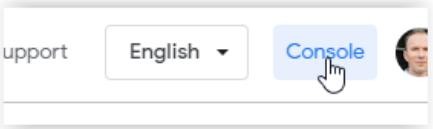
```
docker run -p 8888:8888 \
           -e JUPYTER_ENABLE_LAB=yes \
           -e JUPYTER_TOKEN=pbd \
           --name jupyter \
           -d jupyter/datascience-notebook:latest
```

Możemy także skorzystać z klastra *Dataproc*. Nie jest to „naturalna" metoda, ale jeśli mamy do niego dostęp, to dlaczego tego nie zrobić.

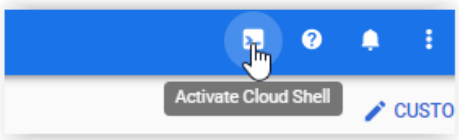
Uruchomienie klastra Dataproc

1. Zaloguj się do platformy *Google Cloud Platform*: `https://cloud.google.com/gcp/`

2. Przejdź do konsoli tej platformy



3. Aktywuj *Cloud Shell* pozwalający zarządzać tę platformą za pomocą poleceń linii komend.

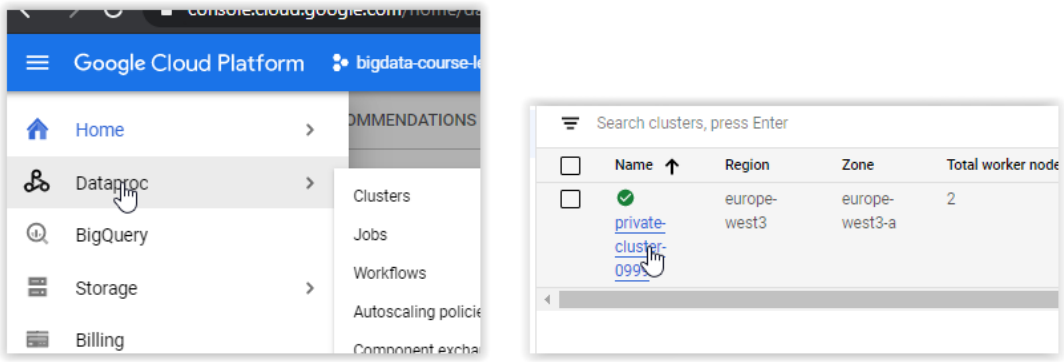


4. Uruchom klaster *Dataproc* korzystając z poniższych poleceń *Cloud Shell*. Ustaw właściwe wartości zmiennych (możesz to zrobić raz, w pliku `.bashrc`).
   W rzeczywistości do naszych zadań nie potrzebujemy klastra. Dlatego uruchomimy klaster z jednym węzłem.

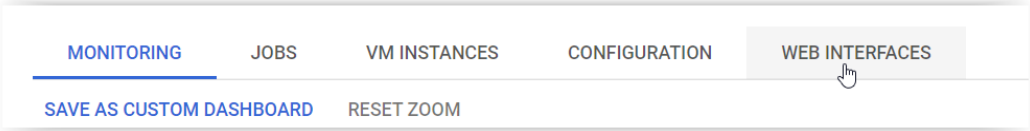
```
gcloud dataproc clusters create ${CLUSTER_NAME} \
 --enable-component-gateway --bucket ${BUCKET_NAME} \
 --region ${REGION} --subnet default \
 --single-node \
 --master-machine-type n1-standard-4 --master-boot-disk-size 50 \
 --image-version 2.1-debian11 \
 --optional-components=DOCKER,JUPYTER \
 --project ${PROJECT_ID} --max-age=3h
```

JypyterLab

5. Wybierając z lewego menu pozycję *Dataproc* wyświetl listę klastrów. Zaczekaj aż klaster zostanie utworzony. Odśwież stronę i wybierz klaster utworzony przed chwilą

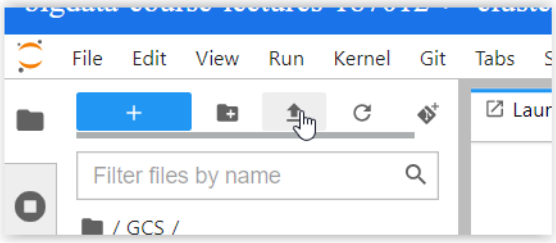


6. Wybierz zakładkę *Web Interfaces*



7. A następnie *JupyterLab*

8. Zaimportować notatkę (notatnik) `SP01_w1_23-Python-zadania.ipynb` możesz za pomocą stosownego przycisku nad listą dostępnych notatników



Dalsze instrukcje znajdziesz we wnętrzu notatki