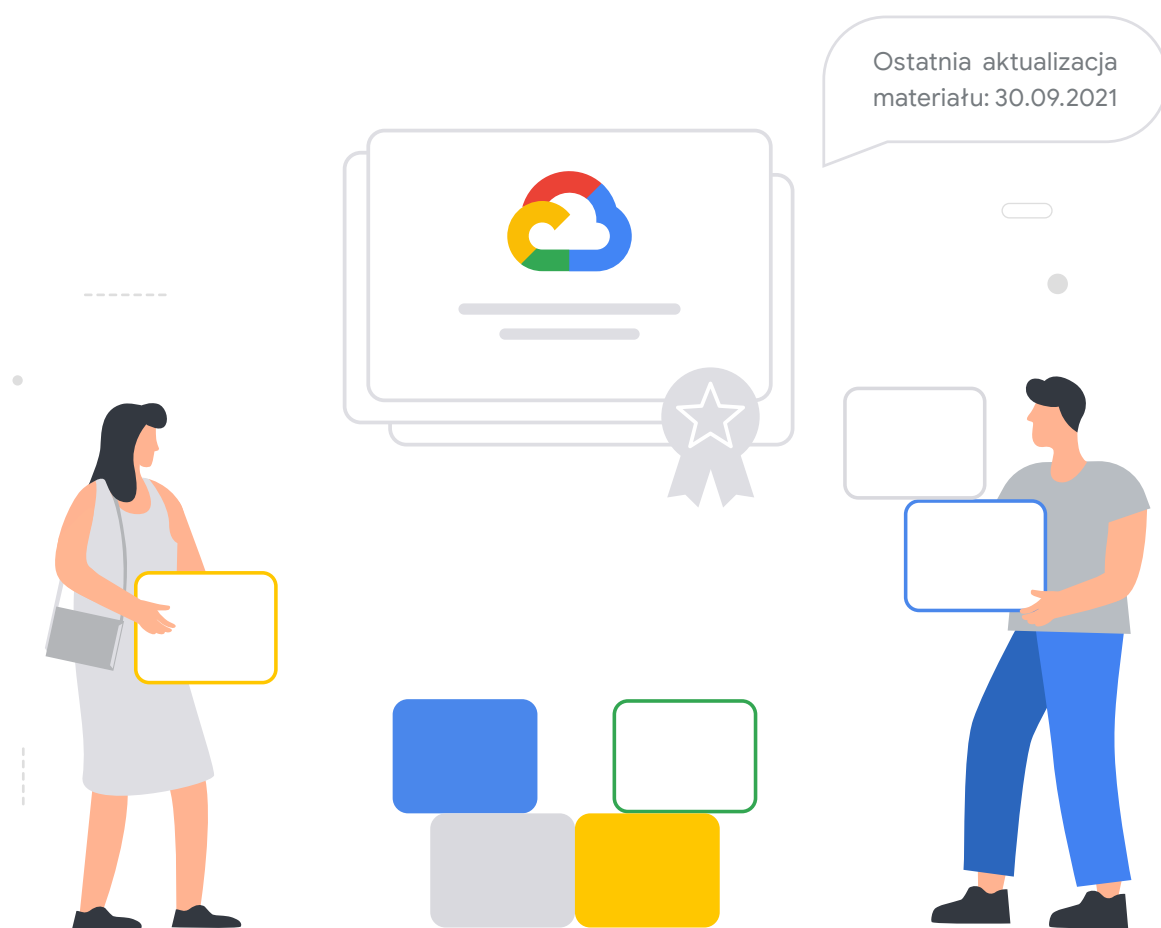




Pomoce do programu:

Rozwiń swoją karierę z Google Cloud.

Ostatnia aktualizacja
materiału: 30.09.2021



SPIS TREŚCI

Wstęp	3
1. Tworzenie VM	5
2. Zmienna środowiskowa	6
3. Tworzenie klastra	7
4. Tworzenie zasobnika/koszyeczka	8
5. Role na poziomie konsoli Uprawnień i projektu	8
6. Tworzenie kontroli dostępności (UPTIME CHECK)	9
7. Wykorzystanie Pythona w GCP na przykładzie Pub/Sub	10
8. Tworzenie sieci VPC	11
9. Obiekty deployment	12
10. TensorFlow	13
11. Inicjowanie Cloud Dataprep	14
12. Otwarcie konsoli BigQuery	14
13. Tworzenie instancji Cloud SQL	14
14. Docker	15
15. Biblioteka interfejsów API	15
16. Generowanie klucza API	16

Rozwiń swoją karierę z Google Cloud.



Sprawdź

Proponowane ścieżki rozwoju

Zobacz nagranie



Komendy w systemie Linux

curl z argumentami : np **-u**

łączenie komend linux ' | ' oraz **&**

cat, grep, ls, cd

uruchamianie skryptu: **python <skrypt.py>**

Tworzenie plików

touch <nazwa_pliku>

Otwieranie za pomocą edytora np. vi/nano

vi <nazwa_pliku>

Przydatne frazy z SQL

**SELECT, FROM, WHERE, AS, SUM(), COUNT, GROUP BY,
ORDER BY, DESC, IN, JOIN, MAX, MIN, UNNEST**

1. Tworzenie VM

W menu nawigacyjnym kliknij Compute Engine > Instancje maszyn wirtualnych.

1. Aby utworzyć nową instancję maszyny wirtualnej kliknij: **Utwórz instancję**.
2. Przy tworzeniu **nowej instancji** możliwe jest skonfigurowanie różnych parametrów.
3. Aby połączyć się z tą maszyną wirtualną przez **SSH**, w wierszu odpowiadającym maszynie kliknij **SSH**.

\$STREFA - zmienna środowiskowa zawierająca nazwę strefy (ang. zone), np. europe-west1-b

```
gcloud compute ssh gcelab2 --zone europe-west1-b
```

Komenda nawiązuje połączenie z maszyną gcelab2 w strefie europe-west1-b

Częste komendy

```
sudo apt-get update
```

 aktualizowanie informacji systemowych o repozytoriach pakietów Debiana Linux

```
gcloud compute instances list
```

 podgląd listy instancji VM-ów

Korzystanie z **wiersza poleceń** (gcloud):

```
gcloud compute instances create gcelab2 \
```

```
--machine-type n1-standard-2 \
```

```
--zone us-central1-f
```

Domyślne wartości nowej instancji:

- Najnowszy obraz Debian 10 (buster).
- Typ maszyny n1-standard-2.
- Główny dysk stały o takiej samej nazwie jak instancja.
Dysk jest automatycznie podłączany do instancji.

2. Zmienna środowiskowa

```
export PROJECT_ID=<ID_projektu>
```

Definiowanie zmiennej środowiskowej z nazwą projektu.

Użyj \$PROJECT_ID zamiast wpisywania nazwy projektu w linii poleceń.

```
export ZONE=<strefa>
```

Definiowanie zmiennej środowiskowej z nazwą strefy.

Praktyczne użycie export do stworzenia zasobnika:

```
export BUCKET_NAME=<your-unique-name>
```

```
gsutil mb gs://$BUCKET_NAME/
```

Często wykorzystywane w laboratoriach

Użycie flagi `-zone`.

Jeżeli jednak zawsze pracujesz w obrębie jednego regionu lub strefy, możesz skonfigurować domyślny region i domyślne strefy w gcloud:

Aby **ustawić domyślną strefę** na np.: `us-central1-a`

```
gcloud config set compute/zone us-central1-a
```

oraz region na: `us-central1`

```
gcloud config set compute/region us-central1
```

3. Tworzenie klastra

Tworzenie klastra GKE `gcloud container clusters create [NAZWA-KLASTRA]`

Uwierzytelniające dane klastra `gcloud container clusters get-credentials [NAZWA-KLASTRA]`

Usunięcie klastra `gcloud container clusters delete [NAZWA-KLASTRA]`

Aby utworzyć klaster Cloud Dataproc o nazwie example-cluster, z domyślnymi ustawieniami uruchom polecenie `gcloud dataproc clusters create example-cluster`

Aktualizacja klastra

Aby zmienić liczbę instancji roboczych w klastrze na 4, uruchom następujące polecenie:

```
gcloud dataproc clusters update example-cluster --num-workers 4
```

Możliwe jest zmniejszenie liczby instancji roboczych, używając tego samego polecenia:

```
gcloud dataproc clusters update example-cluster --num-workers 2
```

Tworzenie VM i opisy tych różnych parametrów

```
gcloud compute instances create www1 \
```

```
--image-family debian-9 \
```

```
--image-project debian-cloud \
```

```
--zone us-central1-a \
```

```
--tags network-lb-tag \
```

```
--metadata startup-script="#!/bin/bash
```

```
sudo apt-get update
```

```
sudo apt-get install apache2 -y
```

```
sudo service apache2 restart
```

```
echo '<!doctype html><html><body><h1>www1</h1></body></html>' | tee /var/www/html/index.html"
```

4. Tworzenie zasobnika/koszyczka

Tworzenie zasobnika z linii poleceń

```
gsutil mb -p [PROJECT_ID] gs://[BUCKET_NAME]
```

Dla wygody, ustaw zmienne środowiskowe \$PROJECT_ID i \$BUCKET_NAME za pomocą komendy export.

Alternatywny sposób:

1. Wybierz **menu nawigacyjne > Cloud Storage > Przeglądarka**.
2. Kliknij **Utwórz zasobnik**, ustaw nazwę i wybierz parametry.

Cały opis znajduje się [tutaj](#).

Przesyłanie obiektu do zasobnika

1. Na stronie przeglądarki Cloud Storage kliknij nazwę utworzonego zasobnika.
2. Na karcie **Obiekty** kliknij **Prześlij pliki** (np.jpg).
3. Sprawdzenie czy plik został przesłany [tutaj](#).

5. Role na poziomie konsoli Uprawnień i projektu

1. Wybierz **menu nawigacyjne > Administracja > Uprawnienia**.
Jesteś teraz w konsoli „Administracja”.
2. U góry strony kliknij przycisk **+DODAJ** i przejrzyj role w projekcie, klikając menu „Wybierz rolę”:

role/wyświetlający, role/edytujący, role/właściciel, role/przeglądający (beta)

6. Tworzenie kontroli dostępności (UPTIME CHECK)

1. W menu Cloud Console po lewej kliknij **Kontrola dostępności**, a następnie kliknij **Utwórz kontrolę dostępności**.

Zastosowanie

Możliwe jest utworzenie kontroli dostępności sprawdzającej czy maszyna wirtualna działa.

Wdrożenie funkcji na przykładzie helloWorld

```
cloud functions deploy helloWorld -- <parametry>
```

Sprawdzenie stanu funkcji

```
gcloud functions describe helloWorld
```

Sprawdzanie dzienników funkcji

```
gcloud functions logs read helloWorld
```

7. Wykorzystane Pythona w GCP na przykładzie Pub/Sub

Otwórz wiersz poleceń `sudo apt-get update`

Zainstaluj wirtualne środowisko Pythona `sudo apt-get install virtualenv`

`virtualenv -p python3 venv`

Aktywuj `source venv/bin/activate`

Po zainstalowaniu środowiska możesz korzystać z Pythona.

Instalacja biblioteki Pub/Sub `pip install --upgrade google-cloud-pubsub`

Klonowanie repozytoriów z Github: `git clone`

Po pobraniu przejdź komenda linuxową `cd python-pubsub/samples/snippets`

Aby wyświetlić pomoc jak obchodzić się ze skryptem (w tym przypadku) `publisher.py`

wpisz `python publisher.py -h`

8. Tworzenie sieci VPC

Rozdział Google Cloud Skills Boost poświęcony VPC

Multiple VPC Networks

1. Aby móc utworzyć **sieć VCP** otwórz **Menu nawigacyjne** i kliknij **Sieć VPC > Sieci VPC**.
2. **Utwórz sieć VPC**.
3. Podaj wartości i parametry.
4. **Utwórz**.

Tworzenie sieci z wiersza poleceń

Np. sieć privatenet

```
gcloud compute networks create privatenet \
```

```
--<ew parametry>
```

Tworzenie podsieci:

```
gcloud compute networks subnets create privatesubnet-us \
```

```
--<ew parametry>
```

Sprawdzenie dostępnych sieci VPC

```
gcloud compute networks list
```

Lista dostępnych podsieci VPC

```
gcloud compute networks subnets list
```

Tworzenie reguły zapory sieciowej

1. Otwórz **menu nawigacyjne** i kliknij **Sieć VPC > Zapora sieciowa**.
2. Kliknij **+** **Utwórz regułę zapory sieciowej**.
3. Ustaw wartości i parametry.
4. **Utwórz**.

Sprawdzenie czy reguła zapory sieciowej jest gotowa

```
gcloud compute firewall-rules list
```

9. Obiekty deployment

Polecenie `explain` w `kubectl` opisze istniejący obiekt Deployment (wdrożenie)

```
kubectl explain deployment
```

Możliwe jest użycie opcji rekursywnej `kubectl explain deployment --recursive`

Możliwe jest użycie polecenia `explain` podczas przechodzenia przez laboratorium, aby zrozumieć strukturę obiektu Deployment i zrozumieć, co robią poszczególne pola.

```
kubectl explain deployment.metadata.name
```

Stworzenie obiektu wdrażania za pomocą `kubectl create`:

```
kubectl create -f deployments/auth.yaml
```

Sprawdzenie czy obiekt został utworzony:

```
kubectl get deployments
```

Po utworzeniu wdrożenia Kubernetes utworzy zestaw replik dla wdrożenia.

Możliwe jest zweryfikowanie czy dla wdrożenia utworzono ReplicaSet `kubectl get replicaset`

Zestaw replik będzie nosił nazwę `auth-xxxxxxx`

Możliwe jest wyświetlenie **podów**, które zostały utworzone w ramach wdrożenia.

Pojedynczy pod jest tworzony przez Kubernetes podczas tworzenia zestawu ReplicaSet.

Usługa uwierzytelniania

Użyj polecenia `kubectl create` `kubectl create -f services/auth.yaml`

Po utworzeniu wdrożenia możliwe jest skalowanie, uruchamianie aktualizacji, wstrzymywanie, wycofywanie.

Inne rodzaje wdrożeń: **Canary** deployments, **Blue-green** deployments.

10. TensorFlow

Najczęściej używany model do AI w laboratoriach to TensorFlow. W laboratoriach dowiesz się jak wdrażać aktywny model (do prognozowania) w AI Platform.

1. Kliknij **menu nawigacyjne**. Otwórz **AI Platform** i wybierz **Notatki**.
2. Na stronie Instancje Notebook kliknij **Nowa instancja**. Wybierz najnowszą wersję TensorFlow 2.x bez procesorów graficznych.
3. Otwórz **JupyterLab**. Pojawią się do wyboru Notebook, Console i other. W laboratoriach najczęściej wykorzystuje się terminal (Console).
4. Zaimportuj dane

```
git clone https://github.com/GoogleCloudPlatform/training-data-analyst
```

W Notatkach w AI Platform możliwe jest przejście do

```
training-data-analyst/self-paced-labs/ai-platform-qwikstart
```

i otworzenie

```
ai_platform_qwik_start.ipynb
```

Należy wyczyścić wszystkie komórki w notatkach. Aby to zrobić, na pasku narzędzi notatek znajdź przycisk **Wyczyść**, a następnie wybierz **Uruchom**.

Do przesuwania się w notebooku użyj Run lub **SHIFT + ENTER**.

11. Inicjowanie Cloud Dataprep

1. Wybierz **menu nawigacyjne** > **Dataprep**.
2. **AKCEPTUJĘ** > **Zgadzam się i chcę przejść dalej** - > **Continue**

Aby utworzyć **zbiór danych** o nazwie taxirides (przejazdy_taksówką)

z poziomu wiersza poleceń, należy uruchomić polecenie `bq mk taxirides`

`mk` make(stwórz) `bq` bigquery

12. Otwarcie konsoli BigQuery

1. **Menu nawigacyjne** > **BigQuery:** > **DONE**

Stworzenie zbioru danych (dataset):

1. **Menu nawigacyjne** > **BigQuery**.
2. W panelu po prawej kliknij **View actions**, a następnie **Create dataset**.
3. Podaj ID i dalsze parametry. -> **Create Dataset**.

Przejdź do **EDYTORA BigQuery** i wpisz lub wklej zapytania, aby utworzyć np. model.

13. Tworzenie instancji Cloud SQL

1. **Menu nawigacyjne** > **SQL**.
2. **UTWÓRZ INSTANCJĘ**.
3. Wybranie silnika bazy danych. Wybierz **MySQL** > **CREATE**.

Przesyłanie plików CSV do tabel z Cloud Storage

1. Na stronie instancji SQL kliknij **IMPORTUJ**.
2. W polu pliku w Cloud Storage kliknij **Przeglądaj**.
3. Wybierz z listy w Database.
4. W polu „Table” podaj nazwę tabeli.

14. Docker

Uruchomienie kontenera Hello world `docker run hello-world`

Podgląd na obraz kontenera pobrany z Docker Hub `docker images`

Start po ponownym uruchomieniu `docker run hello-world`

Sprawdzenie aktualnie działających kontenerów `docker ps`

Wpisanie z parametrem `-a` pokazuje także te, które zakończyły działanie `docker ps -a`

Podgląd zbudowanych obrazów `docker images`

Zatrzymanie i usunięcie wszystkich kontenerów:

`docker stop $(docker ps -q)``docker rm $(docker ps -aq)`

15. Biblioteka interfejsów API

Włączenie interfejsu API:

1. Otwórz **Navigation menu** > **APIs & Services** > **Dashboard** > **Enable APIs and Services**
2. Wpisz np. Dialogflow i kliknij “enable”.
3. Otwórz <https://dialogflow.cloud.google.com/>
4. **Zaakceptuj** > **create agent** i podaj parametry.

Dodawanie plików do katalogu

W sesji Cloud Shell, w prawym górnym rogu strony kliknij ikonę menu z trzema kropkami, a następnie kliknij **Upload file** (Prześlij plik).

Wybierz plik np. demo-image.png. Spowoduje to dodanie obrazu do Twojego katalogu.

Aby otrzymać konkretną ścieżkę `realpath demo-image.png`

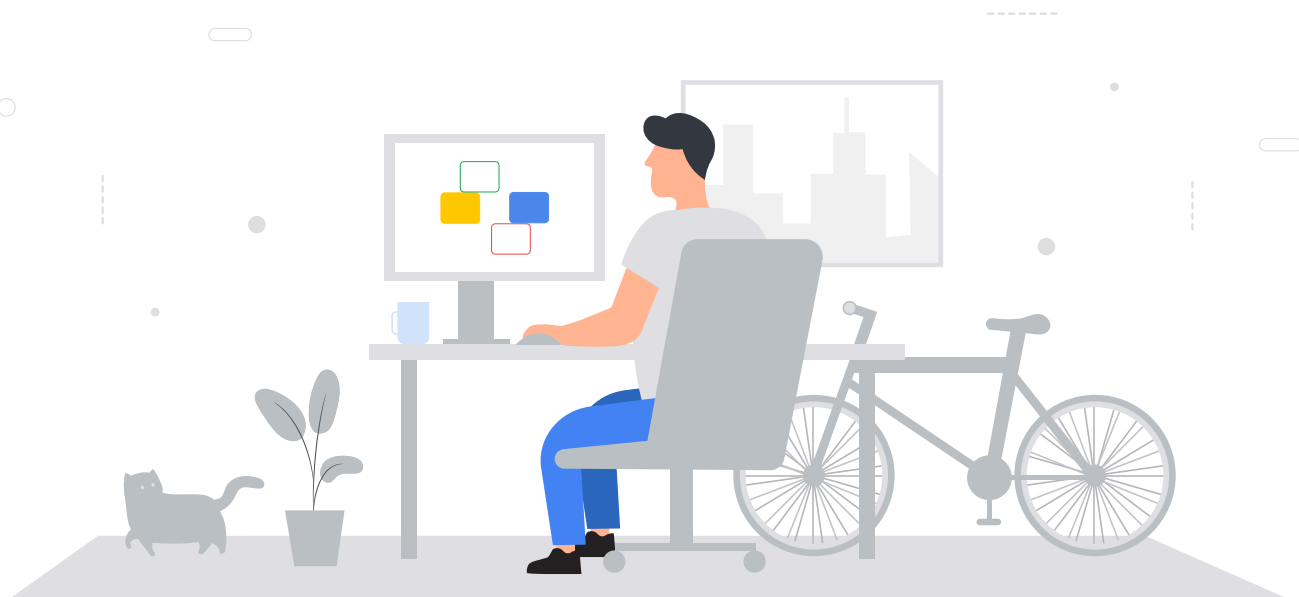
Sprawdzanie folderów komenda `ls` !

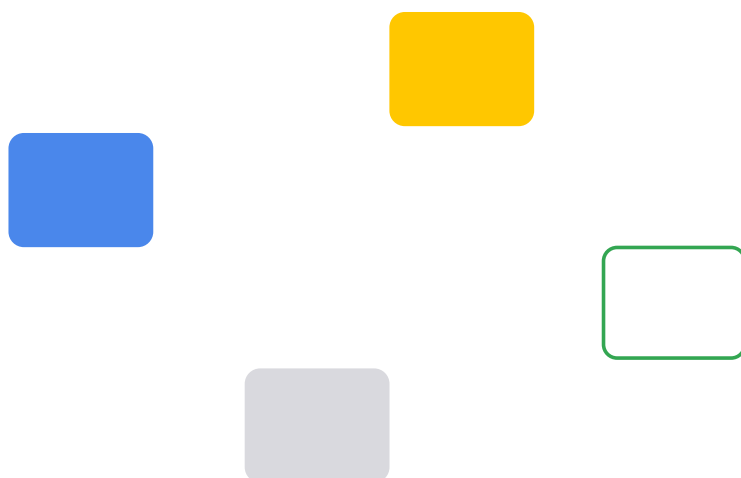
16. Generowanie klucza API

1. **Navigation Menu > APIs & services > Credentials.**
2. Kliknij **+ Create Credentials.**
3. Wybierz **API key.**
4. Po skopiowaniu możliwe jest wyeksportowanie do **zmiennej środowiskowej.**

Zapisanie zdjęcia na Cloud Storage

1. **Navigation menu > Cloud Storage.**
2. Stwórz zasobnik: **CREATE BUCKET.**
3. Po podaniu nazwy i stworzeniu bucketu możesz nadać mu uprawnienia/role.
4. Wybierz **> UPLOAD FILES** aby wgrać pliki.





Materiał przygotowany przez zespół Google Cloud for Higher Education.

Niniejszy materiał stanowi jedynie pomoc dydaktyczną do programu Rozwiń karierę z Google Cloud. Podstawowym źródłem wiedzy w ramach programu Rozwiń karierę z Google Cloud pozostają laboratoria dostępne na platformie szkoleniowej Google Cloud Skills Boost.

Ostatnia aktualizacja materiału: 30.09.2021