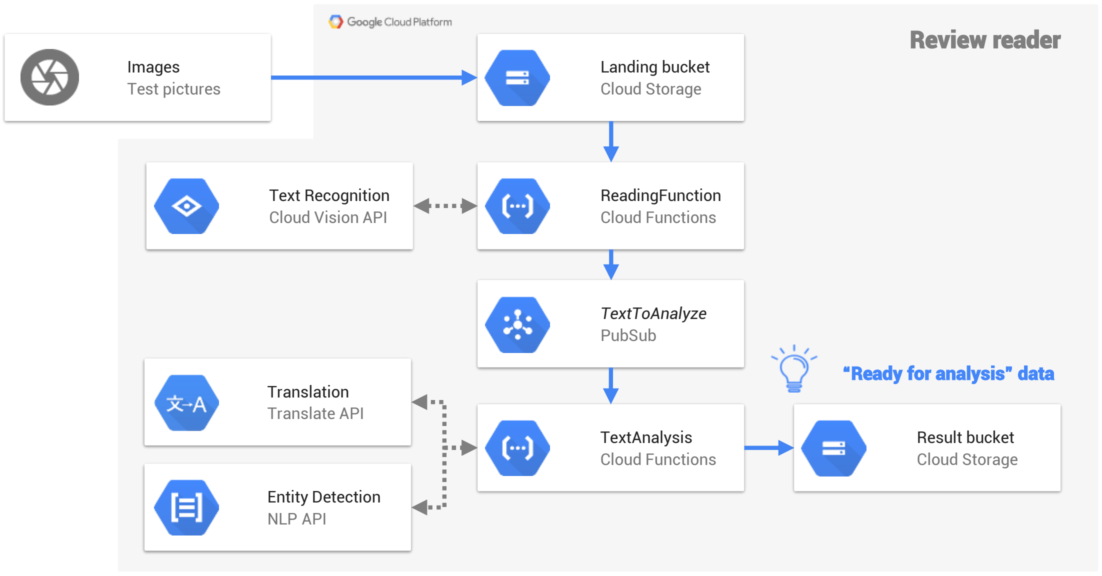
Google Cloud Platform
Practical Lab for students

accenture

Exercice 1 – Review reader



In this exercise, you are going to deploy a Cloud based review reader to extract meaningful information from images. The architecture of the system to build is available above. To reach your goal, you'll need to realize the following task :

Storage (Google Cloud Storage)
- Create a **bucket storage** that will hosts your base pictures

- Create a **bucket storage** that will hosts the analysis resulting files

Google Cloud APIs
- In the API catalog, make sure that Cloud Vision API, Translate API and Natural Language Processing API are activated for your project (if not, you won't be able to call them later)

Data pipelining (Google Cloud Functions and PubSub)
- Create a PubSub topic named "*TextToAnalyze*". This will be used to send messages from the OCR cloud function to the translation/NLP cloud function.

Data processing (Google Cloud Functions and PubSub)
- Create a first Cloud Function that **will OCR the image thanks to the Google Cloud Vision API**. Trigger this cloud function with the creation of a new object in the bucket storage related to the base pictures. This function will have to send the resulting text to the "*TextToAnalyze*" PubSub topic in order to communicate with the second function (cf. below).

- Create a second Cloud Function **that will detect the language of the detected text, translate it to English and, finally, extract information from it**. Trigger this cloud function with the reception of a new message to the "*TextToAnalyze*" PubSub topic. This function will have to use the Translation API and the NLP API to extract the following information :

    - Text detected with OCR
    - Language
    - English translation
    - Overall sentiment of the text
    - Entities detected in the text
    - Timestamp (date – hint : check the filename)

Finally, this function will have to **write an analysis resulting file** (format is up to you) that will contains all the previously extracted information in the bucket related to resulting files.

Google Cloud Platform

Practical Lab for students

Key resources & snippets

Make sure you check those resources and documentation while developing to get to know Google Cloud Platform products as well as the associated Python library and clients :

Google Cloud Platform – Python SDK Library :
https://cloud.google.com/python/docs/reference/

Google Cloud Platform & APIs – Product pages :

Natural Language API :

- **Product page :** https://cloud.google.com/natural-language/
- **API :** https://googleapis.dev/python/language/latest/index.html

Translate API :

- **Product page :** https://cloud.google.com/translate/
- **API :** https://googleapis.dev/python/translation/latest/client.html

Vision API :

- **Product page :** https://cloud.google.com/vision/?hl=fr
- **API :** https://googleapis.dev/python/vision/latest/index.html

Cloud Storage :

- **Product page :** https://cloud.google.com/storage/
- **API :** https://googleapis.dev/python/storage/latest/client.html

PubSub :

- **Product page :** https://cloud.google.com/pubsub/docs/overview?hl=fr
- **API :** https://googleapis.dev/python/pubsub/latest/index.html

Cloud Functions :

- **Product page :** https://cloud.google.com/functions/docs/

PubSub – Send a message to a topic

```
#Generate PubSub client associated to a topic
publisher = pubsub_v1.PublisherClient()
topic_path = publisher.topic_path(PROJECT_ID,
PUB_SUB_TOPIC_NAME)

#Publish to the PubSub topic
# detected_text is the core of the message
# image_name is a parameter (name is its the value)
publisher.publish(topic_path,
bytes(detected_text,encoding="utf-8"),
image_name=name)
```

Translate – Detect source language and translate to a target language

```
# Translate - Get the language of a text and translate it
client_translation = translate.Client()

# Detected language
response_languageDetection=client_translation.detect_language(text)
detectedLanguage=response_languageDetection["language"]

# Translate to English
response_Translation=client_translation.translate(text,
target_language="en")
translatedText=response_Translation["translatedText"]
```

Storage – Retrieve bucket and copy file into it

```
# Storage - Get bucket and write file to it
client_storage = storage.Client()

bucket = client_storage.get_bucket("mon_bucket_name")
blob = bucket.blob(original_image_export_name)
blob.upload_from_filename(filename="/tmp/myJson.json")
```

Google Cloud Platform
Practical Lab for students

accenture

Key resources & snippets

Cloud Vision – Make an OCR request

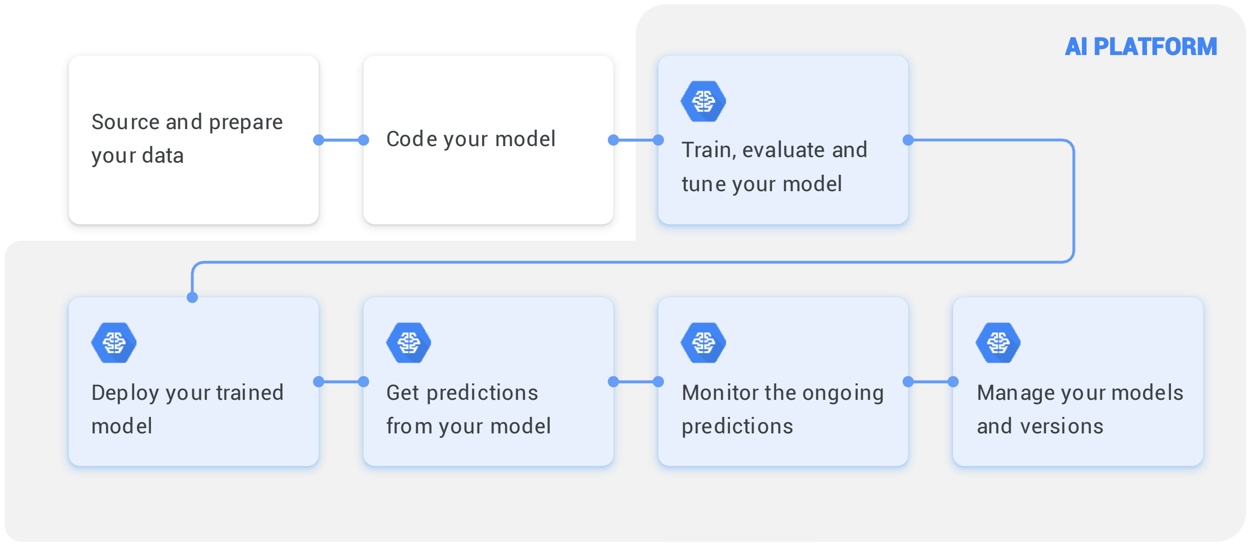
```
#Generate Cloud Vision client
vision_client = vision.ImageAnnotatorClient()

text_detection_response = vision_client.text_detection({
'source': {'image_uri': 'gs://{}/{}'.format(bucket,
filename)}
})

annotations = text_detection_response.text_annotations
rawResult = annotations[0].description
```

NLP – Create a document and request for analysis

```
client_nlp = language.LanguageServiceClient()

# Create Document object for NLP API
document = language.types.Document(
content=translatedText,
type='PLAIN_TEXT'
)

# Call the NLP API for sentiment and entities
analysis
response_sentiment = client_nlp.analyze_sentiment(
document=document,
encoding_type='UTF32'
)

response_entities = client_nlp.analyze_entities(
document=document,
encoding_type='UTF32'
)
```

Exercice 2 – Deploy a production model

In this exercise, you are going to deploy a TensorFlow saved model to production.
To do so, we are going to use GCP AI Platform and test it on real images.


AI PLATFORM
Source and prepare your data
Code your model
Train, evaluate and tune your model
Deploy your trained model
Get predictions from your model
Monitor the ongoing predictions
Manage your models and versions

Google Cloud Platform

Practical Lab for students

accenture

Exercice 2 – Deploy a production model

Model creation (Google Cloud AI Platform)

- In AI Platform, go to the "Models" section
- Create a first model named "digit_reader" (or the name you want)
- In digit_reader, create a first version of the model that will have the following characteristics :
  - Name: v1
  - Python version: 3.5
  - Framework: TensorFlow
  - Framework version : 1.14.0
  - ML execution version : 1.14
  - Model URI : point to the "saved_model" folder in your bucket, probably located at :
  your_bucket > Exo2_ObjectDetectionModel > trained_models > digit_reader > saved_model

- Redo the operation with the same information but with a model named "digit_line_detector" (or the name you want). This time, the model "saved_model" folder is probably located at :
  your_bucket > Exo2_ObjectDetectionModel > trained_models > digit_line_detector > saved_model

Model utilization (Google Cloud AI Platform)

- In AI Platform, go to the "Notebooks" and start the pre-built instance which is there for you. Then click on "Open JupyterLab".

- Once in JupyterLab, go to the "Exo2_ObjectDetectionModel" folder and open the "use_models_on_images.ipynb" notebook.

- There, complete the code with your credentials (project ID, models names, etc.) and play with it to observe the results. You can change the image_path to apply the process to the different images stored in the "images" folder

Curious about life at Accenture ? Reach out !

Nicolas Auricchio

Technology Architect
Industrial Engineer – ISIMs 2016

nicolas.auricchio@accenture.com

LinkedIn Profile

Nathan Derave

Machine Learning Engineer
Civil Engineer – FPMs 174 | 2018

nathan.derave@accenture.com

LinkedIn Profile

Look for your future job >

https://www.accenture.com/be-en/careers/jobsearch?jk=Belgium+Young+Graduates