

CLOUD COMPUTING WITH GOOGLE CLOUD & DOCKER

AI FRAMEWORKS

Brendan Guillouet

November 2th, 2019

Institut National des Sciences Appliquées

INTRODUCTION

IA FRAMEWORKS - TOOLS

ML Python
Libraries



Viz' Python
Libraries



seaborn



Python
Environment



Other Tools &
Frameworks



IA FRAMEWORKS - TOOLS

ML Python
Libraries



Viz' Python
Libraries



seaborn



Python
Environment



Other Tools &
Frameworks



TABLE OF CONTENTS

Introduction

Script Python

Google Cloud

Pipeline d'exécution

Docker

SCRIPT PYTHON

WHY USING SCRIPT?

Jupyter limits :

- it's an exploration tool
 - but cloud machine are accounted on an hourly base.
- Non-linear workflow.
 - Easy to write messy code.
- Not designed to handle large-scale experiment.
- Not designed for production.
 - Can't be run from terminal, no test procedure.

WHY USING SCRIPT?

Jupyter limits :

- it's an exploration tool
 - but cloud machine are accounted on an hourly base.
- Non-linear workflow.
 - Easy to write messy code.
- Not designed to handle large-scale experiment.
- Not designed for production.
 - Can't be run from terminal, no test procedure.

⇒ Exploration work : *Jupyter*.

WHY USING SCRIPT?

Jupyter limits :

- it's an exploration tool
 - but cloud machine are accounted on an hourly base.
- Non-linear workflow.
 - Easy to write messy code.
- Not designed to handle large-scale experiment.
- Not designed for production.
 - Can't be run from terminal, no test procedure.

⇒ Exploration work : *Jupyter*.

⇒ Large-scale/production work : Write *Script*!

SCRIPT EXECUTION

File *script.py*

```
a = 5
b = 3
c = a + b
print("The answer is %d" %c)
```

Terminal

```
bguillou $> python script.py
bguillou $> The answer is 8
```

Write two scripts :

- **learning.py** : to learn a model, save it in the *model* directory, save results in the *results* directory and
- **prediction.py** : to generate prediction and save it in the *results* directory

on *CatsVsDogs* data.

⇒ Ensure that the **complete workflow is working** locally before pushing the code on the instance.

LIBRAIRIE ARGPARSE

File *script.py*

```
import argparse

parser = argparse.ArgumentParser()
parser.add_argument('--a', type=int, default=5)
parser.add_argument('--b', type=int, default=3)

args = parser.parse_args()

c= args.a + args.b
print("The answer is %d" %c)
```

Terminal

```
bguillou $> python script.py
bguillou $> The answer is 8
bguillou $> python script.py --a 4
bguillou $> The answer is 7
bguillou $> python script.py --a 4 --b 2
bguillou $> The answer is 7
```

LIBRAIRIE PICKLE

File *learning.py*

```
import pickle
...
results = {"learning_time" : lt, "accuracy" : acc}
pickle.dump(results, open("/User/bguillouet/data/results.pkl", "wb"))
```

File *explore_results.py*

```
import pickle
results = pickle.load(open("/User/bguillouet/data/results.pkl", "rb"))
print(results)
```

Terminal

```
bguillou $> ls data/
bguillou $>
bguillou $> python learning.py
bguillou $> ls data/
bguillou $> results.pkl
bguillou $> python explore_results.py
bguillou $> {"learning_time" : lt, "accuracy" : acc}
```

LIBRAIRIE HASHLIB

python script.py

```
import argparse
import hashlib
import pickle

parser = argparse.ArgumentParser()
parser.add_argument('--a', type=int, default=5)
parser.add_argument('--b', type=int, default=3)
parser.add_argument('--type_op', type=str, default="addition")

args = parser.parse_args()

if args.type_op == "addition":
    c = args.a + args.b
else:
    c = args.a * args.b

results = vars(args)
print("Argument dictionary: " + str(results))
results.update({"results": c})
print("Argument dictionary with score: " + str(results))

args_str = "_".join([k + ":" + str(v) for k, v in sorted(results.items(), key=lambda x : x[0])])
print("Argument string: " + args_str)

id_str = hashlib.md5(args_str.encode("utf8")).hexdigest()
print(id_str)

result_file = "/User/bguillouet/data/" + id_str + ".pkl"

pickle.dump(results, open(result_file, "wb"))
```

```
bguillou $> python script.py
bguillou $> {"a":5, "b":3, "type_op":"addition"}
bguillou $> {"a":5, "b":3, "type_op":"addition", "results":8}
bguillou $> a:5_b:3_type_op:addition_results:8
bguillou $> aZezEzj7jhZ8793DeefdjZ9
bguillou $> ls data/
bguillou $> aZezEzj7jhZ8793DeefdjZ9.pkl
```

GOOGLE CLOUD

Suite of cloud computing services with more than 90 products.

- Power computation , Database, AI, Networking, Security *etc..*

Suite of cloud computing services with more than 90 products.

- Power computation, Database, AI, Networking, Security etc..

Suite of cloud computing services with more than 90 products.

- **Power computation**, Database, AI, Networking, Security etc..
 - \implies Google Cloud Engine

Suite of cloud computing services with more than 90 products.

- **Power computation**, Database, AI, Networking, Security etc..
 - \implies Google Cloud Engine

Why not IA tools?

- *AutoML* or *ML Engine* are tools that provide solution for non-expert.

Suite of cloud computing services with more than 90 products.

- **Power computation**, Database, AI, Networking, Security etc..
 - \implies Google Cloud Engine

Why not IA tools?

- *AutoML* or *ML Engine* are tools that provide solution for non-expert.

What do I need to know to use Google CCloud Engine?

- Basic knowledge of Google cloud interface.
- Basic use of terminal command (no graphic interface).
- *gcloud SDK*.

<https://cloud.google.com/sdk/docs/quickstart-debian-ubuntu>

Command line tool which allow to manage VM instance.

- *gcloud init*. To be used at first utilization.
- *gcloud compute instances start/stop/delete instance_name*. To start/stop/remove instance.
- *gcloud compute scp --recurse CopyFrom CopyTo*
 - *--recurse* (optional) : To be used if directory is copied.
 - *CopyFrom* : Location of the file or directory to be copied.
 - *CopyTo* : Location of the directory where the file or directory will be copied.
 - *Syntax* : `[[[USER@]INSTANCE :]DIR]` . To send file on the instance.

- *gcloud compute ssh --ssh-key-file LocationOfSSHKey --zone europe-west1-b*. To set ssh-connection to the instance.
- *gcloud compute ssh --command 'COMMAND'*. To execute command on the instance.
 - *gcloud compute ssh --command 'mkdir data'*
 - *gcloud compute ssh --command 'python learning.py'*

The screenshot shows the Google Cloud Platform (GCP) console dashboard for a project named 'TestTPGCE'. The interface is in French and features a blue header bar with the GCP logo, project name, and navigation icons. Below the header, there are tabs for 'TABLEAU DE BORD' (Dashboard) and 'ACTIVITÉ' (Activity). The dashboard is divided into several sections:

- Informations sur le projet** (Project Information): Displays project details such as 'Nom du projet' (TestTPGCE), 'ID du projet' (testtpgce), and 'Numéro du projet' (341748641608). It includes a link to 'Accéder aux paramètres du projet'.
- Ressources** (Resources): Lists resources, including 'Compute Engine' with 1 instance.
- Trace**: Shows 'Aucune donnée Trace des sept derniers jours' and a link to 'Premiers pas avec Stackdriver Trace'.
- Compute Engine**: Displays a graph for 'Processeur (%)' (CPU usage) over time. The graph shows a flat line at 0% with a warning icon and the text 'Aucune donnée disponible pour la période sélectionnée'. It includes a link to 'Accéder au tableau de bord Compute Engine'.
- API**: Displays a graph for 'Requêtes (requêtes/s)' (Requests per second) over time, showing a flat line at 0.0175.
- État de Google Cloud Platform**: Shows 'Fonctionnement normal de tous les services' and a link to 'Accéder à Cloud Status Dashboard'.
- Facturation**: Displays 'Frais estimés' (Estimated costs) for the period 1-5 Nov. 2018, showing '0,00 USD \$'. It includes a link to 'Afficher les frais détaillés'.
- Error Reporting**: Shows 'Aucun signe d'erreur. Avez-vous configuré Error Reporting ?' and a link to 'Découvrir comment configurer Error Reporting'.
- Actualités**: Displays 'Serverless from the ground up: Connecting Cloud Functions with a'.

At the bottom of the dashboard, there is a status bar showing 'onglet_creer_pr_...png' and a 'Tout afficher' button.

GCE : SÉLECTION PROJET

The screenshot shows the Google Cloud Platform interface with a modal dialog titled "Sélectionner un projet". The dialog has a search bar at the top with the placeholder text "Rechercher des projets et des dossiers". Below the search bar, there are two tabs: "PROJETS RÉCENTS" (selected) and "TOUS". Under the "PROJETS RÉCENTS" tab, there is a table with two columns: "Nom" and "Identifiant". The table contains one entry: "TestTPGCE" with the identifier "testtpgce". At the bottom of the dialog, there are two buttons: "ANNULER" and "OUVRIR".

Google Cloud Platform

TestTPGCE

TABLEAU DE BORD ACTIVITÉ

Informations sur le projet

- Nom du projet: TestTPGCE
- ID du projet: testtpgce
- Numéro du projet: 341748641608

Accéder aux paramètres du projet

Ressources

- Compute Engine: 1 instance

Trace

Aucune donnée Trace des sept derniers jours

Premiers pas avec Stackdriver Trace

NOUVEAU PROJET

Rechercher des projets et des dossiers



PROJETS RÉCENTS TOUS

| Nom | Identifiant |
|-------------|-------------|
| ✓ TestTPGCE | testtpgce |


ANNULER OUVRIR


Actualités

Serverless from the ground up: Connecting Cloud Functions with a

 Google Cloud Platform 


Nouveau projet

 Il vous reste 22 projets dans votre quota. Demandez une augmentation ou supprimez des projets.
[En savoir plus](#)
[MANAGE QUOTAS](#)


Nom du projet *
My Project 60149 

ID du projet : atomic-griffin-221614. Vous ne pourrez pas le modifier par la suite.

[MODIFIER](#)

Compte de facturation *
High dimensional learning 

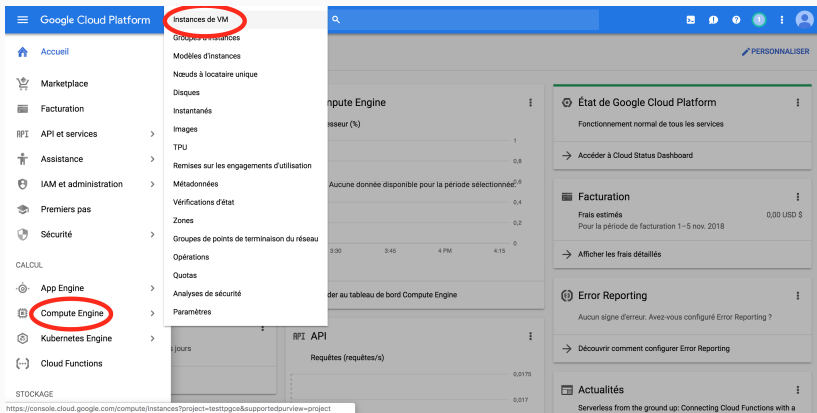
Tous les frais relatifs à ce projet seront facturés sur le compte que vous sélectionnez ici.

Zone *
 Aucune organisation [PARCOURIR](#)

Organisation ou dossier parent

[CRÉER](#) [ANNULER](#)

GCE : MENU PRINCIPAL (HAMBURGER)



GCE : INSTANCE VM

The screenshot shows the Google Cloud Platform console interface. At the top, the header includes 'Google Cloud Platform', a project selector 'TestTPGCE', and a search bar. The left sidebar displays the 'Compute Engine' menu with various options like 'Instances de VM', 'Groupes d'instances', 'Modèles d'instances', etc. The main content area is titled 'Instanc... de VM' and features a red circle around the 'CRÉER UNE INSTANCE' button. Below this, there is a table of VM instances. The table has columns for 'Nom', 'Zone', 'Recommandation', 'Adresse IP interne', 'Adresse IP externe', and 'Se connecter'. One instance, 'Instance-2', is listed in the 'europe-west1-b' zone with an internal IP of '10.132.0.2 (nic0)' and no external IP. To the right of the table, a panel titled 'Sélectionner une instance' shows tabs for 'AUTORISATIONS', 'SURVEILLANCE', and 'LIBELLÉS'. A message at the bottom of this panel states 'Veuillez sélectionner au moins une ressource.'

Google Cloud Platform TestTPGCE

Compute Engine

Instances de VM

Groupes d'instances

Modèles d'instances

Nœuds à locataire unique

Disques

Instantanés

Images

TPU

Remises sur les engagements...

Métadonnées

Vérifications d'état

Zones

Groupes de points de termin...

Opérations

Marketplace

◀

Instanc... de VM

CRÉER UNE INSTANCE

IMPORTER LA VM

MASQUER LE PANNEAU D'INFORMATIONS

Filtrer les instances VM

Colonnes

| <input type="checkbox"/> Nom | Zone | Recommandation | Adresse IP interne | Adresse IP externe | Se connecter |
|-------------------------------------|----------------|----------------|--------------------|--------------------|--------------|
| <input type="checkbox"/> Instance-2 | europe-west1-b | | 10.132.0.2 (nic0) | Aucune | SSH |

Sélectionner une instance


AUTORISATIONS SURVEILLANCE LIBELLÉS


Veuillez sélectionner au moins une ressource.


GCE : CRÉER UNE INSTANCE VM

[←](#) Créer une instance

Pour créer une instance de VM, sélectionnez l'une de ces options :

**Nouvelle instance de VM**
Créer entièrement une instance de VM unique

**Nouvelle instance de VM à partir d'un modèle**
Créer une instance de VM unique à partir d'un modèle existant

**Marketplace**
Déployer une solution prête à l'emploi sur une instance de VM

nom

instance-3

région

us-east1 (Caroline du Sud)

zone

us-east1-b

Type de machine
Cliquez sur "Personnaliser" pour sélectionner des cœurs, la mémoire et des GPU

1 vCPU


3,75 Go de mémoire

Personnaliser

Conteneur

☐ Déployer une image de conteneur dans cette instance de VM. [En savoir plus](#)

Disque de démarrage



Nouveau disque persistant standard de 10 Go
image
Debian GNU/Linux 9 (stretch)

Modifier

Identité et accès à l'API

Compte de service

Compute Engine default service account

Charges d'application de l'accès

☒ Autoriser l'accès par défaut
☐ Autoriser l'accès complet à l'ensemble des API Cloud
☐ Définir l'accès pour chaque API

Pare-feu

Ajouter des tags et des règles de pare-feu pour autoriser un trafic réseau spécifique provenant d'internet

☐ Autoriser le trafic HTTP

☐ Autoriser le trafic HTTPS

[Gestion, sécurité, disques, mise en réseau et location unique](#)

Estimation mensuelle : 24,67 \$
Soit un coût horaire d'environ 0,034 \$
Vous payez ce que vous consommez : facturation à la seconde, sans frais initiaux à supporter
[Détails](#)

18/40

← Créer une instance

Pour créer une instance de VM, sélectionnez l'une de ces options :



Nouvelle instance de VM

Créer entièrement une instance de VM unique



Nouvelle instance de VM à partir d'un modèle

Créer une instance de VM unique à partir d'un modèle existant



Marketplace

Déployer une solution prête à l'emploi sur une instance de VM



Région ?

europa-west1 (Belgique)

Zone ?

europa-west1-b

Type de machine

Cliquez sur "Personnaliser" pour sélectionner des cœurs, la mémoire et des GPU.

Cœurs

Affichage standard



1

vCPU

1 - 96

Mémoire



3,75

Go

1 - 6,5

☐ Plus de mémoire ?

Plate-forme du processeur ?

Automatique

GPU

Le nombre de GPU est lié au nombre de cœurs de processeur et de mémoires sélectionnés pour cette instance. Pour ce type de machine, vous devez sélectionner au moins 1 GPU. [En savoir plus](#)

Nombre de GPU

1

Type de GPU

NVIDIA Tesla K80



Vous ne pouvez pas transférer les machines dotées de GPU pendant la maintenance de l'hôte

[Choisir un type de machine](#)

GCE : DISQUE DÉMARRAGE

Google Cloud Platform

Créer une instance

Pour créer une instance de VM, sélectionnez l'une des options :

Nouvelle instance de VM

Créer entièrement une instance de VM unique

Nouvelle instance de VM à partir d'un modèle

Créer une instance de VM unique à partir d'un modèle existant

Marketplace

Deployer une solution prête à l'emploi sur une instance de VM

Disque de démarrage

Sélectionnez une image ou un instantané pour créer un disque de démarrage, ou associez un disque existant

Images d'OSImages d'applicationImages personnaliséesInstantanésDisques existants

Les VM protégées sont en phase bêta. En savoir plusDismiss

☐ Afficher les images disposant des fonctionnalités des VM protégées

☐ Debian GNU/Linux 9 (stretch)
amd64 built on 20181011

☐ CentOS 6
x86_64 built on 20181011

☐ CentOS 7
x86_64 built on 20181011

☐ CoreOS alpha 1939.0.0
amd64-usr published on 2018-10-24

☐ CoreOS beta 1911.2.0
amd64-usr published on 2018-10-24

☐ CoreOS stable 1855.5.0
amd64-usr published on 2018-10-24

☐ Ubuntu 14.04 LTS
amd64 trusty image built on 2018-10-02

☒ Ubuntu 16.04 LTS
amd64 xenial image built on 2018-10-30

☐ Ubuntu 18.04 LTS
amd64 bionic image built on 2018-10-29

☐ Ubuntu 18.10
amd64 cosmic image built on 2018-10-18

☐ Ubuntu 16.04 LTS Minimal
amd64 xenial minimal image built on 2018-10-29

☐ Ubuntu 18.04 LTS Minimal
amd64 bionic minimal image built on 2018-10-30

☐ Ubuntu 18.10 Minimal
amd64 cosmic minimal image built on 2018-10-18

☐ Container-Optimized OS 69-10095.91.0 stable
Kernel: ChromiumOS-4.14.65 Kubernetes: 1.11.1 Docker: 17.03.2 Family: cos-69-lts

☐ Container-Optimized OS 71-11151.16.0 beta
Kernel: ChromiumOS-4.14.65 Kubernetes: 1.11.1 Docker: 17.03.2 Family: cos-71-beta

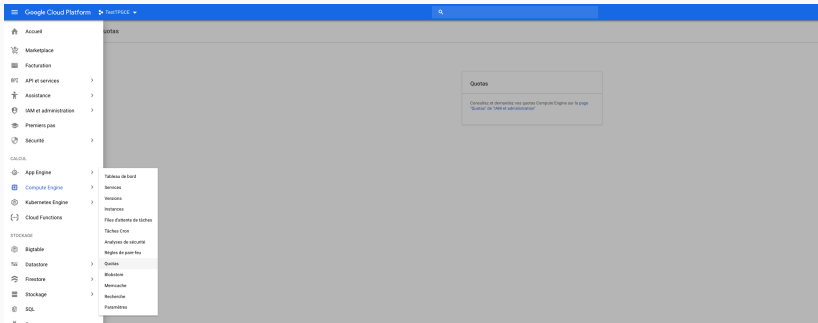
Can't find what you're looking for? Explore hundreds of VM solutions in Marketplace

Type de disque de démarrageTaille (Go)

Disque persistant standard10

SélectionnerAnnuler

GCE : QUOTAS



Google Cloud Platform

TestTPGCE

IAM et administration

Quotas

MODIFIER LES QUOTAS

IAM

Identité et organisation

Règles de l'organisation

Quotas

Comptes de service

Libellés

Confidentialité et sécurité

Paramètres

Clés de chiffrement

Proxy sensible à l'identité (IA...)

Rôles

Journaux d'audit

Type de quotaTous les quotasServiceCompute Engine APIMétriqueGPUs (all regions)ZoneToutes les zonesEffacer

☐ Service

☒ Compute Engine API GPUs (all regions)

ZoneMondial

Utilisation actuelle

Pic d'utilisation sur 7 jours ^

Limite1

GCE : FACTURATION

Google Cloud Platform TestTPGCE

Facturation


instances de VM


CRÉER UNE INSTANCE IMPORTER LA VM ACTUALISER DÉMARRER ARRÊTER RÉINITIALISER SUPPRIMER


Filtrer les instances VM


| Nom | Zone | Recommandation | Adresse IP interne | Adresse IP externe | Se connecter |
|------------|----------------|----------------|--------------------|--------------------|--------------|
| instance-2 | europe-west1-b | | 10.132.0.2 (nic0) | 35.233.12.224 1.7 | SSH |


GCE : FACTURATION


 Google Cloud Platform

 Facturation


 Vue d'ensemble

 Budgets et alertes

 Exportation de la facturation


 Rapports

Vue d'ensemble


High dimensional learning  [RENOMMER LE COMPTE DE FACTURATION](#)

ID du compte de facturation : 018EC2-9A109B-882A89

Crédits



31,55 \$
Crédits restants
Sur 50,00 \$



304
Jours restants
Expire le 1 oct. 2019

Projets associés à ce compte de facturation

| Nom du projet | Identifiant du projet | |
|---------------|-----------------------|---|
| TestTPGCE | testtpgce | ⋮ |
| Second | second-221614 | ⋮ |

- Facturation arrive en différé.
- 1 projet par compte de facturation.
- 1 machine GPU par projet.
- Capacité de la machine illimité.

- Python
- Cuda
- Docker
- Nvidia-docker

bash_script for installation in *utils/bash_utils_on_gpu*.

PIPELINE D'EXECUTION

1. Write script in your *local* machine.
2. Turn you *instance on*.
3. Build environment(if first used).
4. Send latest version of your code to the instance.
5. Send data to the instance.
7. Run the script on the instance.
8. Copy the results you want to analyze from the instance to your local machine.
10. Turn your instance off.

1. Write script in your *local* machine.
2. Turn you *instance on*.
3. Build environment(if first used).
4. Send latest version of your code to the instance.
5. Send data to the instance.
6. Run container.
7. Run the script on the instance.
8. Copy the results you want to analyze from the instance to your local machine.
9. Stop and remove container.
10. Turn your instance off.

PIPELINE - COMMAND

1. Write script in your *local* machine.
2. `gcloud compute instances start ..`
3. `gcloud compute ssh --command 'mkdir data'`
4. `gcloud compute scp script.py bguillou@instance-gpu :/home/`
5. `gcloud compute scp --recurse data bguillou@instance-gpu :/home/`
6. `gcloud compute scp --recurse data bguillou@instance-gpu :/home/`
7. `gcloud compute ssh --command 'python script.py -a 3'`
8. `gcloud compute scp --recurse bguillou@instance-gpu :/home/results/
/home/`
9. `gcloud compute instances start ..`
10. `gcloud compute instances stop ..`

PIPELINE - COMMAND

1. Write script in your *local* machine.
2. `gcloud compute instances start ..`
3. `gcloud compute ssh --command 'mkdir data'`
4. `gcloud compute scp script.py bguillou@instance-gpu :/home/`
5. `gcloud compute scp --recurse data bguillou@instance-gpu :/home/`
6. `gcloud compute ssh --command 'sudo nvidia-docker run ...'`
7. `gcloud compute ssh --command 'python script.py -a 3'`
8. `gcloud compute scp --recurse bguillou@instance-gpu :/home/results/
/home/`
9. `gcloud compute ssh --command 'sudo nvidia-docker container stop ...'`
10. `gcloud compute instances stop ..`

PIPELINE - COMMAND

1. Write script in your *local* machine.
2. `gcloud compute instances start ..`
3. `gcloud compute ssh --command 'mkdir data'`
4. `gcloud compute scp script.py bguillou@instance-gpu :/home/`
5. `gcloud compute scp --recurse data bguillou@instance-gpu :/home/`
6. `gcloud compute ssh --command 'sudo nvidia-docker run ...'`
7. `gcloud compute ssh --command 'python script.py -a 3'`
8. `gcloud compute scp --recurse bguillou@instance-gpu :/home/results/
/home/`
9. `gcloud compute ssh --command 'sudo nvidia-docker container stop ...'`
10. `gcloud compute instances stop ..`

Too many possibility to make a mistake.

PIPELINE - COMMAND

1. Write script in your *local* machine.
2. `gcloud compute instances start ..`
3. `gcloud compute ssh --command 'mkdir data'`
4. `gcloud compute scp script.py bguillou@instance-gpu :/home/`
5. `gcloud compute scp --recurse data bguillou@instance-gpu :/home/`
6. `gcloud compute ssh --command 'sudo nvidia-docker run ...'`
7. `gcloud compute ssh --command 'python script.py -a 3'`
8. `gcloud compute scp --recurse bguillou@instance-gpu :/home/results/
/home/`
9. `gcloud compute ssh --command 'sudo nvidia-docker container stop ...'`
10. `gcloud compute instances stop ..`

Too many possibility to make a mistake.

⇒ Pipeline

We will use a tool composed of 3 *python script* and a *.yaml* file :

- **conf.yaml** : This file works like a dictionary which contains global variables such as location of directory.
- **instances.py** : This script defines a Python class *InstanceManager* which encapsulates calls to *gcloud*. For example *list()* function of this class calls this command in terminal :

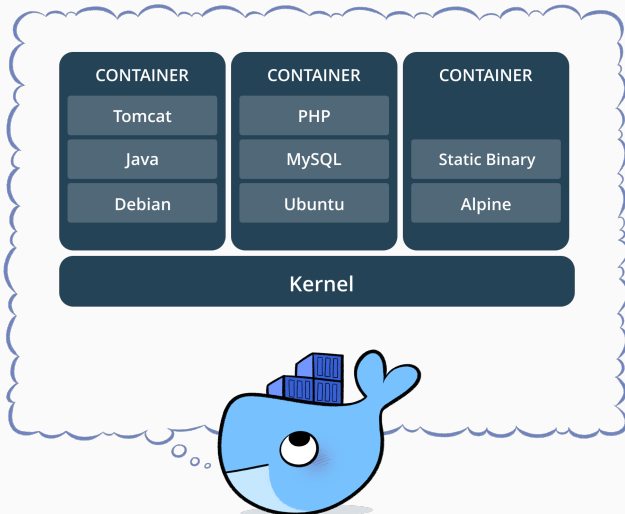
```
gcloud compute instances list
```

- **project.py** This script defines a Python class *ProjectManager* which contains specific functions to manage your project.
For example, the *update_data(self, zip_file)* function allows to send a *zip_file* from your *data* directory from your local machine to the *data* directory on your instance.
- **main.py** This is a python script which contains all the command to execute your pipeline.

DOCKER

WHAT IS DOCKER?

Docker enables to build Virtual machines which are easy to re-create on different compute environment.



WHY SHOULD I USE DOCKER AS A DATA SCIENTIST?

- **Reproducibility** : wrapping all you environment in a Docker container ensure the possibility to recreate your environment and makes your work more accessible.
- **Portability of your compute environment** To move your code and your model easily on machine with more computational power.
- **Enlarge your possibility** : being comfortable with Docker can allow you to use various solution available with docker.

- **Image** : Its like a turned-off VM which contains the tools you want. Ex : Ubuntu + TensorFlow with Nvidia Drivers and a running Jupyter Server.
- **Container** : Is an instantiation of an image. You can have multiple copies of the same image running.
- **Dockerfile** : Recipe for creating an Image.
- **DockerHub / Image Registry** : Place where people/organization can post public (or private) docker images to facilitate collaboration and sharing.

<https://hub.docker.com/>

WHAT WE'LL DO IN THIS TP

- Write a *Dockerfile* which is based on the official *Tensorflow Dockerfile* available on *DockerHub*.
- Use the *Dockerfile* to build a *image*.
- Launch *container* with different option from the build *image*.

DOCKERFILE

Dockerfile of the image we'll build :

```
FROM tensorflow/tensorflow:latest-devel-gpu-py3
RUN apt-get update && apt-get install -y
python-opencv python-tk vim
RUN pip install h5py keras pytest scikit-image
seaborn tqdm gensim
```

- **FROM** : Specifies the base image you want to build on top of. Docker will look in your local environment for the image you called and if it cannot find it locally it will search it in *DockerHub*.
- **RUN** : Is followed by normal commands that would be directly run on terminal to install librairies or framework.

Run the BUILD command in order to build your **IMAGE**.

```
docker build -t ImageName -f /Docker/Dockerfile /Docker/
```

Run the BUILD command in order to build your **IMAGE**.

Name of
the image



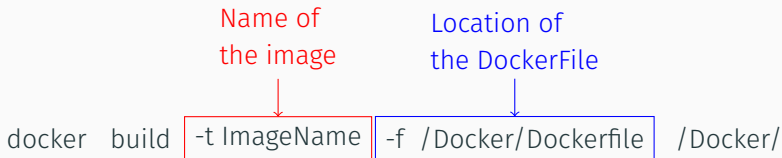
```
docker build -t ImageName -f /Docker/Dockerfile /Docker/
```

BUILD IMAGE

Run the BUILD command in order to build your **IMAGE**.

docker build -t ImageName -f /Docker/Dockerfile /Docker/

Name of the image Location of the DockerFile



BUILD IMAGE

Run the BUILD command in order to build your **IMAGE**.

Diagram illustrating the components of the `docker build` command:

```
docker build -t ImageName -f /Docker/Dockerfile /Docker/
```

The components are labeled as follows:

- Name of the image** (red text) points to `-t ImageName`.
- Location of the DockerFile** (blue text) points to `-f /Docker/Dockerfile`.
- Location of the build context** (orange text) points to `/Docker/`.

The build context is the location of the folder to which the **ADD** statement will reference. This means that all external files required by the *Dockerfile* will be located here.

RUN CONTAINER


Run the RUN command in order to run your CONTAINER.

```
docker run -it --gpus all --name ContainerName  
-v ~/CatsVsDogs/ :/root/CatsVsDogs ImageName
```

RUN CONTAINER

Run the RUN command in order to run your CONTAINER.

Mode in which
the container
will be launched

```
docker run  -it -gpu all -name ContainerName  
-v ~/CatsVsDogs/ :/root/CatsVsDogs ImageName
```

Mode of the container could be :

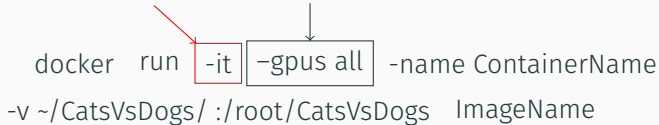
- -it : interactive mode,
- -dt : detached mode.

RUN CONTAINER

Run the RUN command in order to run your CONTAINER.

Mode in which
the container
will be launched

gpus
option



```
docker run -it -gpus all -name ContainerName  
-v ~/CatsVsDogs/ :/root/CatsVsDogs ImageName
```

Mode of the container could be :

- -it : interactive mode,
- -dt : detached mode.

RUN CONTAINER

Run the RUN command in order to run your CONTAINER.

Mode in which
the container
will be launched

gpus
option

Name of
the container

```
docker run -it -gpus all -name ContainerName  
-v ~/CatsVsDogs/ :/root/CatsVsDogs ImageName
```

Mode of the container could be :

- -it : interactive mode,
- -dt : detached mode.

RUN CONTAINER

Run the RUN command in order to run your CONTAINER.

Mode in which
the container
will be launched

gpus
option

Name of
the container

```
docker run -it -gpus all -name ContainerName  
-v ~/CatsVsDogs/ :/root/CatsVsDogs ImageName
```

Diagram illustrating the components of the `docker run` command:

- `-it` is highlighted with a red box and labeled "Mode in which the container will be launched".
- `-gpus all` is highlighted with a grey box and labeled "gpus option".
- `-name ContainerName` is highlighted with a blue box and labeled "Name of the container".
- `ImageName` is highlighted with a red box and labeled "Name of the image".

Mode of the container could be :

- `-it` : interactive mode,
- `-dt` : detached mode.

RUN CONTAINER

Run the RUN command in order to run your CONTAINER.

Mode in which
the container
will be launched

gpus
option

Name of
the container

docker run

-it

-gpus all

-name ContainerName

-v ~/CatsVsDogs/ :/root/CatsVsDogs

ImageName

Volume Option

Name of
the image

Mode of the container could be :

- -it : interactive mode,
- -dt : detached mode.

The `-v` option allow you to use some data you have in your machine within a container.

```
-v ~/CatsVsDogs/ :/root/CatsVsDogs
```

The `-v` option allow you to use some data you have in your machine within a container.

Declare
Option



`-v` `~/CatsVsDogs/ :/root/CatsVsDogs`

The `-v` option allow you to use some data you have in your machine within a container.

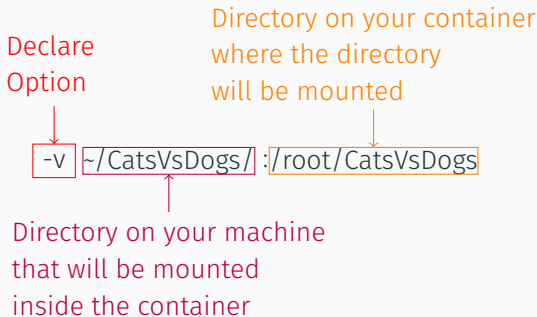
Declare
Option

`-v` `~/CatsVsDogs/` `:/root/CatsVsDogs`

Directory on your machine
that will be mounted
inside the container

MOUNTED VOLUME

The `-v` option allow you to use some data you have in your machine within a container.



- `sudo docker image ls -a`
- `sudo docker container ls -a`
- `sudo docker start/stop/rm container_name -a`
- `sudo docker exec container_name 'Command to execute in container' -a`

- Use Jupyter on the instance (via ssh connection)
- Google Colab
- Image Gcloud

<https://towardsdatascience.com/how-docker-can-help-you-become-a-more-effective-data-scientist-7fc048ef91d5>

RÉFÉRENCES
