

Déploiement de l'API de détection des objets sur Ubuntu VPS 18.04 LTS

Etape1 : installation des prérequis

A l'instar des projets de machine learning, le déploiement du code sur un serveur dédié est indispensable pour le rendre accessible depuis tous les dispositifs informatiques . Ainsi, nous allons utiliser le serveur web Apache2 et le module `mod_wsgi` qui constitue une interface facilitant l'hébergement des applications développées avec python sur le serveur web.

Tout d'abord, on commence par mettre à jour le serveur Ubuntu

`sudo apt-get update`

`sudo apt-get upgrade`

Ainsi, on va installer Apache 2.4.29 et vérifier son état.

`sudo apt install apache2`

On vérifie l'état du serveur à l'aide de la ligne de commande suivante :

`sudo service apache2 status`

```
ubuntu@vps-0689a54b:~$ sudo service apache2 status
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
   Active: active (running) since Mon 2020-09-28 22:26:39 UTC; 21min ago
   Process: 12694 ExecStop=/usr/sbin/apachectl stop (code=exited, status=0/SUCCESS)
   Process: 31881 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
   Process: 12699 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 12714 (apache2)
    Tasks: 55 (limit: 2260)
   CGroup: /system.slice/apache2.service
           └─12714 /usr/sbin/apache2 -k start
             └─12717 /usr/sbin/apache2 -k start
               └─12718 /usr/sbin/apache2 -k start

Sep 28 22:26:39 vps-0689a54b systemd[1]: Starting The Apache HTTP Server...
Sep 28 22:26:39 vps-0689a54b systemd[1]: Started The Apache HTTP Server.
ubuntu@vps-0689a54b:~$
```

On installe `mod_wsgi` en utilisant :

`sudo apt-get install libapache2-mod-wsgi-py3`

Et on l'active en utilisant : *`sudo a2enmod wsgi`*

On vérifie l'état de `mod_Wsgi` en accédant au dossier **`/usr/lib/apache2/modules`** via la commande :

ldd mod_wsgi.so

```
ubuntu@vps-0689a54b:~$ cd /usr/lib/apache2/modules/
ubuntu@vps-0689a54b:/usr/lib/apache2/modules$ ldd mod_wsgi.so
        linux-vdso.so.1 (0x00007fff577d4000)
        libpython3.6m.so.1.0 => /usr/lib/x86_64-linux-gnu/libpython3.6m.so.1.0 (0x00007flf8d465000)
        libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007flf8d246000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007flf8ce55000)
        libexpat.so.1 => /lib/x86_64-linux-gnu/libexpat.so.1 (0x00007flf8cc23000)
        libz.so.1 => /lib/x86_64-linux-gnu/libz.so.1 (0x00007flf8ca06000)
        libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007flf8c802000)
        libutil.so.1 => /lib/x86_64-linux-gnu/libutil.so.1 (0x00007flf8c5ff000)
        libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x00007flf8c261000)
        /lib64/ld-linux-x86-64.so.2 (0x00007flf8dd4a000)
ubuntu@vps-0689a54b:/usr/lib/apache2/modules$
```

Comme mentionné dans la figure, mod_wsgi utilise la version 3.6.9 de python qui est la version appliquée par défaut sur Ubuntu 18.04, ainsi, nous allons installer les bibliothèques nécessaires pour cette version spécifique.

A cet effet, on commence par installer pip3 qui va nous permettre d'installer les packages pour python3.6 :

sudo apt install python3-pip

on installe aussi l'environnement de développement pour python3 via

sudo apt install python3-dev

maintenant on va installer Flask, numpy, opencv et pytesseract

Flask 0.12.2: *pip3 install flask*

Numpy 1.13.3: *pip3 install numpy*

Opencv 4.4.0.42: *sudo apt install libopencv-dev*

Pip3 install opencv-python

Pytesseract 4.0.0: *pip3 install pytesseract*

Installation de Tesseract-ocr pour la langue arabe

sudo apt install tesseract-ocr-ara

Etape 2 : Création de l'API sur le serveur

Dans une deuxième étape, nous allons transférer le projet vers le serveur en utilisant FTP.

Tout d'abord, nous allons placer l'API dans le répertoire **/var/www** à l'aide de la commande suivante :

cd /var/www

sudo mkdir DetectionAPP

```
cd DetectionAPP
```

```
sudo mkdir APP_code
```

```
cd APP_code
```

```
sudo mkdir static templates YOLO
```

```
cd static
```

```
sudo mkdir uploads detections plaques
```

Avant de transférer le projet vers Ubuntu, on va modifier les permissions pour donner l'accès à FTP pour modifier dans le serveur :

```
sudo chmod -R a+rwX /var/www/DetectionAPP/APP_Code
```

Ainsi, la structure du répertoire est la suivante :

```
tree /var/www/DetectionAPP/APP_Code
```

```
ubuntu@vps-0689a54b:/var/www/DetectionAPP/APP_Code$ tree
.
├── API.py
├── __pycache__
│   └── Training_model.cpython-36.pyc
├── static
│   ├── detections
│   ├── plaques
│   └── uploads
├── templates
│   ├── about.html
│   ├── index.html
│   └── output.html
├── Training_code.py
└── YOLO
    ├── coco.names
    ├── yolov3.cfg
    └── yolov3.weights

7 directories, 9 files
```

On va tester l'installation du code en exécutant l'API sur le serveur local :

```
sudo python3 API.py
```

```
ubuntu@vps-0689a54b:/var/www/DetectionAPP/APP_Code$ sudo python3 API.py
* Running on http://127.0.0.1:4000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 382-573-204
```

Etape3 : configuration de Apache2 et mod wsgi

Nous allons créer un nouvel hôte virtuel pour héberger l'API sur le serveur web.

Tout d'abord, on va accéder au répertoire **/etc/apache2/sites-available** pour créer le fichier de configuration:

sudo nano /etc/apache2/sites-available/APP.conf

Dans le fichier APP.conf nous allons définir le virtual host

```
<VirtualHost *:80>
    ServerName 51.210.45.30
    ServerAdmin admin
    WSGIScriptAlias / /var/www/DetectionAPP/APP.wsgi
    <Directory /var/www/DetectionAPP/APP_Code/>
        Order allow,deny
        Allow from all
    </Directory>
    Alias /static /var/www/DetectionAPP/APP_Code/static
    <Directory /var/www/DetectionAPP/APP_code/static/>
        Order allow,deny
        Allow from all
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/error.log
    LogLevel warn
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

On va activer l'hôte virtuel en utilisant la commande suivante :

Sudo a2ensite APP.conf

On passe maintenant à créer le fichier APP.wsgi dans le répertoire **/var/www/DetectionAPP** pour exécuter l'API sur Apache2

sudo nano /var/www/DetectionAPP/APP.wsgi

```
#!/usr/bin/python3
import sys
import logging
logging.basicConfig(stream=sys.stderr)
sys.path.insert(0, "/var/www/DetectionAPP/APP_Code/")
sys.path.insert(0, "/var/www/DetectionAPP/APP_Code/static")
from API import app as application
```

Finalement, on va redémarrer le serveur pour appliquer les modifications .

sudo service apache2 restart

Output



Détection des objets

La détection des objets dans les images est une problématique qui a affronté l'Humain depuis une longue période.

Aujourd'hui grâce aux algorithmes de machine learning et à la vision par ordinateur, la reconnaissance des objets est désormais possible et réalisable.

Ainsi, dans le contexte de la photo360, nous avons développé une application web capable de reconnaître les voitures, leurs plaques d'immatriculation ainsi que déchiffrer le matricule à l'aide de l'OCR.

D'où l'application de l'algorithme YOLO et les modules de la bibliothèque OpenCV pour détecter, reconnaître et localiser les voitures dans les images.

Continuer



API DE DETECTION D'OBJETS

Veuillez insérer une image

Browse...

No file selected.

Envoyer



Processus de détection des objets

Image d'origine



Image après détection



Detection de la plaque d'immatriculation



le contenu de la plaque d'immatriculation est : 68858 FT 172)

NB: la détection de texte avec Tesseract OCR n'est pas optimale vu que l'algorithme n'arrive pas à détecter les caractères spéciaux tels que la barre '|'