Fiche Complète : Configuration d'un Serveur VPS pour l'API de Traitement d'Images

MAECKELBERG Julien

19 janvier 2026

Table des matières

1 Objectif

Mettre en place un serveur VPS distant pour héberger l'API FastAPI de traitement d'images de fusées. Ce serveur permettra de recevoir des images depuis l'application mobile, d'effectuer l'analyse avec les modèles de deep learning, et de retourner les dimensions calculées.

Attention

Ce serveur fonctionnera 24h/24 et 7j/7 sans nécessiter qu'un ordinateur personnel reste allumé. Il sera accessible depuis Internet via son adresse IP publique.

2 Choix du VPS

2.1 Caractéristiques recommandées

Pour ce projet, un VPS d'entrée de gamme est largement suffisant. Les caractéristiques minimales recommandées sont :

— **RAM :** 4 Go

— **vCores :** 4 cœurs CPU

— **Stockage :** 30 Go

— **CPU :** Processeur standard (pas besoin de GPU)

— **Bande passante :** Illimitée ou suffisante pour les transferts d'images

— **Système d'exploitation :** Ubuntu 22.04 LTS **(TRÈS IMPORTANT !)**

Information

Ces caractéristiques correspondent au minimum testé et fonctionnel. Un VPS plus puissant n'apportera pas d'amélioration significative pour ce projet, car le traitement d'images avec les modèles PyTorch reste raisonnablement rapide sur CPU.

2.2 Fournisseur recommandé

Le fournisseur **OVHcloud** a été testé avec succès pour ce projet. D'autres fournisseurs comme DigitalOcean, Scaleway ou Hetzner peuvent également convenir.

— **Prix indicatif :** ∼10 €/mois pour un VPS avec ces caractéristiques

— **Disponibilité :** 99.9% de uptime garanti

— **Support :** Documentation complète et support technique disponible

2.3 Système d'exploitation

Attention

Choisissez impérativement **Ubuntu 22.04 LTS** lors de la commande du VPS. Cette version est stable, supportée jusqu'en 2027, et compatible avec toutes les dépendances du projet.

3 Configuration du serveur

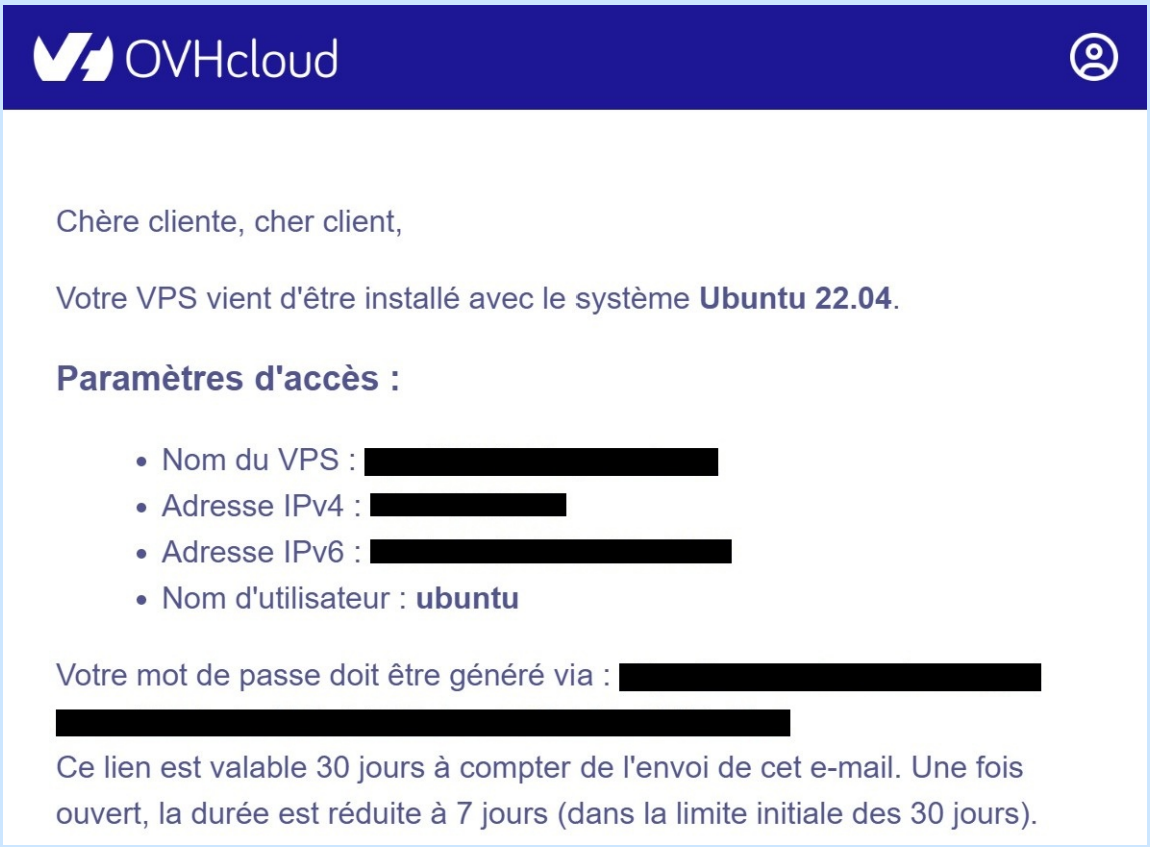
3.1 Réception des informations de connexion

Après la commande du VPS, vous recevrez un email de votre fournisseur contenant les informations de connexion :

- **Nom du VPS :** Par exemple `vps-xxxxxxxx.vps.ovh.net`
- **Adresse IPv4 :** Par exemple `xxx.xxx.xxx.xxx`
- **Adresse IPv6 :** Par exemple `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`
- **Nom d'utilisateur :** Généralement `ubuntu` ou `root`
- **Mot de passe :** Lien sécurisé pour générer/récupérer le mot de passe

Information

Voir l'image ci-dessous pour un exemple d'email de configuration OVHcloud :



Attention

Le lien pour générer le mot de passe est généralement valable 30 jours. Une fois ouvert, sa durée de validité est réduite à 7 jours. Générez et notez votre mot de passe rapidement.

3.2 Première connexion SSH

Depuis votre ordinateur Windows, ouvrez PowerShell ou le terminal Windows et connectez-vous au serveur :

```
ssh ubuntu@xxx.xxx.xxx.xxx
```

Note

Remplacez `ubuntu` par votre nom d'utilisateur et `xxx.xxx.xxx.xxx` par l'adresse IP de votre VPS. Au premier connexion, il vous sera demandé de confirmer l'empreinte du serveur (tapez `yes`).

Entrez le mot de passe reçu par email. Vous devriez maintenant être connecté au serveur.

3.3 Mise à jour initiale du système

PREMIÈRE FOIS UNIQUEMENT

Avant toute installation, mettez à jour le système :

```
sudo apt update
sudo apt upgrade -y
```

Cette opération peut prendre plusieurs minutes.

4 Installation des dépendances

4.1 Installation de Python et des outils de base

PREMIÈRE FOIS UNIQUEMENT

Installez Python 3.10 et les outils nécessaires :

```
sudo apt install -y python3.10 python3.10-venv python3-pip
sudo apt install -y git curl wget nano
```

Vérifiez l'installation :

```
python3.10 --version
```

Résultat attendu : `Python 3.10.x`

4.2 Création du répertoire de travail

Créez le dossier qui contiendra l'API :

```
mkdir -p ~/ml_server
cd ~/ml_server
```

Note

Remplacez `ubuntu` par votre nom d'utilisateur.

4.3 Création de l'environnement virtuel Python

PREMIÈRE FOIS UNIQUEMENT

Créez un environnement virtuel isolé pour le projet :

```
python3.10 -m venv .venv
```

Activez l'environnement virtuel :

```
source .venv/bin/activate
```

Le terminal doit afficher : `(.venv)`

4.4 Installation des bibliothèques Python

PREMIÈRE FOIS UNIQUEMENT

Installez toutes les dépendances nécessaires au projet :

```
python -m pip install --upgrade pip
pip install fastapi uvicorn[standard]
pip install python-multipart
pip install Pillow
pip install torch torchvision --index-url https://download.pytorch.org/whl/
    cpu
pip install segmentation-models-pytorch
pip install opencv-python-headless
pip install scipy numpy
```

Information

Nous installons la version CPU de PyTorch car le VPS n'a pas de GPU. L'option `-index-url` permet de télécharger une version optimisée pour CPU, beaucoup plus légère.

Vérifiez les installations :

```
python -c "import torch; print(torch.__version__)"
python -c "import cv2; print(cv2.__version__)"
python -c "import fastapi; print(fastapi.__version__)"
```

5 Transfert des fichiers vers le serveur

Vous devez transférer 5 fichiers essentiels vers le serveur :

- `analyse_fusee.py` : Code d'analyse d'images
- `main_api.py` : API FastAPI
- `modele_fusee_final_V7.pth` : Modèle de segmentation de la fusée
- `modele_mires_final_V7.pth` : Modèle de détection des mires
- `noseConeShape.py` : Détection du type d'ogive

5.1 Méthode 1 : Via ligne de commande (SCP)

Depuis votre ordinateur Windows (PowerShell), dans le dossier contenant les fichiers :

```
scp aileronsDim.py ubuntu@xxx.xxx.xxx.xxx:~/ml_server/
scp analyse_fusee.py ubuntu@xxx.xxx.xxx.xxx:~/ml_server/
scp main_api.py ubuntu@xxx.xxx.xxx.xxx:~/ml_server/
scp modele_fusee_final_V7.pth ubuntu@xxx.xxx.xxx.xxx:~/ml_server/
scp modele_mires_final_V7.pth ubuntu@xxx.xxx.xxx.xxx:~/ml_server/
scp noseConeShape.py ubuntu@xxx.xxx.xxx.xxx:~/ml_server/
```

Note

Remplacez `ubuntu` par votre nom d'utilisateur et `xxx.xxx.xxx.xxx` par l'adresse IP de votre VPS.

5.2 Méthode 2 : Via FileZilla (Interface graphique)

RECOMMANDÉ pour les débutants

1. Téléchargez et installez FileZilla Client : `https://filezilla-project.org/`

2. Ouvrez FileZilla et configurez une nouvelle connexion :
   - **Hôte :** `sftp://xxx.xxx.xxx.xxx`
   - **Identifiant :** `ubuntu`
   - **Mot de passe :** Votre mot de passe VPS
   - **Port :** 22

   Note

   Remplacez `ubuntu` par votre nom d'utilisateur et `xxx.xxx.xxx.xxx` par l'adresse IP de votre VPS. Le port par défaut est 22.

3. Cliquez sur "Connexion rapide"
4. Dans le panneau de droite, naviguez vers `/home/ubuntu/ml_server`
5. Glissez-déposez les 5 fichiers depuis votre ordinateur (panneau gauche) vers le serveur (panneau droite)

Information

FileZilla offre une interface graphique intuitive, permet de reprendre les transferts interrompus, et affiche la progression des téléchargements. C'est la méthode recommandée pour transférer les gros fichiers de modèles (.pth).

5.3 Vérification des fichiers transférés

Sur le serveur, vérifiez que tous les fichiers sont présents :

```
cd ~/ml_server
ls -lh
```

Vous devriez voir :

```
-rw-rw-r-- 1 ubuntu ubuntu  16K Jan 11 12:00 aileronsDim.py
-rw-rw-r-- 1 ubuntu ubuntu  43K Jan 11 12:00 analyse_fusee.py
-rw-rw-r-- 1 ubuntu ubuntu 4.5K Jan 11 12:00 main_api.py
-rw-rw-r-- 1 ubuntu ubuntu  94M Jan 11 12:00 modele_fusee_final_V7.pth
-rw-rw-r-- 1 ubuntu ubuntu  94M Jan 11 12:00 modele_mires_final_V7.pth
-rw-rw-r-- 1 ubuntu ubuntu 5.2K Jan 11 12:00 noseConeShape.py
drwxrwxr-x 6 ubuntu ubuntu 4.0K Jan 11 12:00 venv
```

6 Test manuel de l'API

Avant de configurer le service permanent, testez que l'API fonctionne correctement.

6.1 Lancement manuel

Depuis le répertoire du projet, avec l'environnement virtuel activé :

```
cd ~/ml_server
source .venv/bin/activate
uvicorn main_api:app --host 0.0.0.0 --port 8000
```

Note

Remplacez `ubuntu` par votre nom d'utilisateur.

Vous devriez voir :

```
INFO:     Started server process [12345]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:8000
```

Information

L'option `-host 0.0.0.0` permet d'accepter les connexions depuis Internet, pas seulement depuis le serveur lui-même.

6.2 Test depuis votre ordinateur

Depuis un navigateur web, accédez à :

`http://xxx.xxx.xxx.xxx:8000/docs`

Vous devriez voir l'interface Swagger de FastAPI permettant de tester l'endpoint `/predict`. Pour arrêter le serveur manuel, faites `Ctrl+C` dans le terminal.

Note

Remplacez `xxx.xxx.xxx.xxx` par l'adresse IP de votre VPS.

7 Configuration du service permanent

Pour que l'API tourne en permanence, même après déconnexion SSH ou redémarrage du serveur, nous allons créer un service systemd.

7.1 Création du fichier de service

Créez le fichier de configuration du service :

```
sudo nano /etc/systemd/system/mlserver.service
```

Copiez le contenu suivant dans le fichier :

```
[Unit]
Description=ML Server API FastAPI
After=network.target

[Service]
Type=simple
User=ubuntu
WorkingDirectory=/home/ubuntu/ml_server
Environment="PATH=/home/ubuntu/ml_server/.venv/bin"
ExecStart=/home/ubuntu/ml_server/.venv/bin/uvicorn main_api:app --host
    0.0.0.0 --port 8000
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target
```

Sauvegardez avec `Ctrl+O`, puis quittez avec `Ctrl+X`.

Note

Remplacez `ubuntu` par votre nom d'utilisateur.

7.2 Activation du service

Rechargez la configuration systemd :

```
sudo systemctl daemon-reload
```

Activez le service pour qu'il démarre automatiquement au boot :

```
sudo systemctl enable mlserver.service
```

Démarrez le service :

```
sudo systemctl start mlserver.service
```

Vérifiez que le service fonctionne :

```
sudo systemctl status mlserver.service
```

Vous devriez voir :

```
Active: active (running)
```

> **Information**
>
> Votre serveur est maintenant opérationnel 24h/24 ! Il redémarrera automatiquement en cas de crash ou de redémarrage du VPS.

8 Commandes utiles

Voici les commandes essentielles pour gérer le service :

8.1 Gestion du service

```
# Demarrer le service
sudo systemctl start mlserver.service

# Arreter le service
sudo systemctl stop mlserver.service

# Redemarrer le service
sudo systemctl restart mlserver.service

# Verifier l'etat du service
sudo systemctl status mlserver.service

# Desactiver le demarrage automatique
sudo systemctl disable mlserver.service

# Reactiver le demarrage automatique
sudo systemctl enable mlserver.service
```

8.2 Consultation des logs

Pour surveiller les logs en temps réel :

```
# Afficher les logs en temps reel (suivre)
journalctl -u mlserver.service -f

# Afficher les 100 dernieres lignes
journalctl -u mlserver.service -n 100

```

```
# Afficher les logs d'aujourd'hui
journalctl -u mlserver.service --since today

# Afficher les logs avec horodatage complet
journalctl -u mlserver.service -o verbose
```

Information

Pour quitter la visualisation des logs en temps réel (`-f`), appuyez sur `Ctrl+C`.

8.3 Diagnostic en cas de problème

Si le service ne démarre pas correctement :

```
# Verifier l'etat detaille
sudo systemctl status mlserver.service -l

# Verifier les erreurs dans les logs
journalctl -u mlserver.service --since "10 minutes ago"

# Tester manuellement la commande
cd ~/ml_server
source .venv/bin/activate
uvicorn main_api:app --host 0.0.0.0 --port 8000
```

Note

Remplacez `ubuntu` par votre nom d'utilisateur.

9 Mise à jour du code

Lorsque vous modifiez le code Python sur votre ordinateur, vous devez le transférer à nouveau sur le serveur et redémarrer le service.

9.1 Procédure de mise à jour

1. Transférez les fichiers modifiés via FileZilla ou SCP
2. Connectez-vous au serveur via SSH
3. Redémarrez le service :

```
sudo systemctl restart mlserver.service

```

4. Vérifiez que le service a redémarré correctement :

```
sudo systemctl status mlserver.service
journalctl -u mlserver.service -n 50

```

Attention

N'oubliez pas de redémarrer le service après chaque modification du code. Les changements ne seront pas pris en compte tant que le service n'est pas redémarré.

10 Sauvegarde et maintenance

10.1 Sauvegarde des modèles et du code

Il est recommandé de sauvegarder régulièrement les fichiers importants :

```
# Creer une archive de sauvegarde
cd ~
tar -czf ml_server_backup_$(date +%Y%m%d).tar.gz ml_server/

# Telecharger la sauvegarde sur votre ordinateur (depuis Windows)
scp ubuntu@xxx.xxx.xxx.xxx:~/ml_server_backup_*.tar.gz ./
```

> **Note**
>
> Remplacez `ubuntu` par votre nom d'utilisateur et `xxx.xxx.xxx.xxx` par l'adresse IP de votre VPS.

10.2 Surveillance de l'espace disque

Vérifiez régulièrement l'espace disque disponible :

```
df -h
```

Les images reçues sont sauvegardées dans `/home/ubuntu/ml_server/received_images`. Pensez à nettoyer ce dossier périodiquement :

```
# Supprimer les images de plus de 30 jours
find ~/ml_server/received_images -type f -mtime +30 -delete

# Ou supprimer toutes les images
rm -rf ~/ml_server/received_images/*
```

> **Note**
>
> Remplacez `ubuntu` par votre nom d'utilisateur.