VPS OVH — CyrilleCode

Procédure — Mise en place du serveur d'hébergement personnel

Développeur	COUR Cyrille
Organisation	Projet personnel
Formation	BTS SIO SLAM — Alternance 2025/2027
Stack	Debian 13 / Docker / Caddy / GitHub Actions
Date	24/08/2026
Version	1.0

Sommaire

1. Contexte

Cette procédure documente la configuration complète du serveur d'hébergement personnel de Cyrille COUR, depuis le VPS vierge jusqu'au premier déploiement automatisé du portfolio (cyrillecode.fr). Elle couvre l'achat des services OVH, la sécurisation initiale du serveur, l'installation de Docker et du reverse proxy Caddy, ainsi que la mise en place d'un déploiement continu via GitHub Actions.

Note — Ce serveur est un projet personnel hébergeant le portfolio et de futurs projets (Gourmandex, Wanderbook…), hors périmètre applicatif de l'UPV. La procédure suit toutefois la charte graphique standard afin de conserver une documentation homogène.

1.1 Infrastructure

Élément	Valeur
VPS	OVH VPS-1 — 2 vCores, 4 Go RAM, 40 Go SSD NVMe
Système d'exploitation	Debian 13 (Trixie)
Domaine	cyrillecode.fr (OVHcloud)
Utilisateur système	debian
Reverse proxy	Caddy (HTTPS automatique via Let's Encrypt)
Orchestration	Docker / Docker Compose
Déploiement continu	GitHub Actions (push sur main)

2. Environnement

2.1 Prérequis

- Compte client OVHcloud
- Accès PowerShell (Windows) ou terminal Unix pour la génération des clés SSH
- Un dépôt GitHub contenant le code à déployer (ex. portfolio)
- Accès à l'espace client OVHcloud pour la zone DNS du domaine

2.2 Achat des services OVH

Le nom de domaine cyrillecode.fr est acheté en premier chez OVHcloud. Le VPS est commandé séparément dans le même espace client, via Bare Metal Cloud > Serveurs Privés Virtuels. L'offre retenue est VPS-1 (2 vCores / 4 Go RAM / 40 Go SSD), largement suffisante pour plusieurs projets légers en Docker et évolutive à chaud si besoin. Le système installé est Debian 13.

Attention — Ne pas confondre avec les offres d'hébergement web mutualisé (gammes Eco/Business/Agencies) proposées après l'achat du domaine — ce sont des offres différentes, sans accès root ni Docker.

3. Procédure

3.1 Étape 1 — Première connexion SSH et sécurisation de base

Détail technique

L'adresse IP du VPS et le mot de passe root initial sont envoyés par email par OVH, et également visibles dans l'espace client (Bare Metal Cloud > VPS > Accueil).

```
ssh root@IP_DU_VPS
# ou selon l'image OVH, l'utilisateur par défaut peut être "debian" :
ssh debian@IP_DU_VPS
```

Sécurisation via clé SSH

Sur le PC (Windows / PowerShell), génération de la paire de clés :

```
ssh-keygen -t ed25519 -C "email@exemple.fr"
```

Copie de la clé publique sur le VPS (équivalent PowerShell de ssh-copy-id) :

```
type $env:USERPROFILE\.ssh\id_ed25519.pub | ssh debian@IP_DU_VPS "mkdir -p ~/.ssh
&& chmod 700 ~/.ssh && cat >> ~/.ssh/authorized_keys && chmod 600
~/.ssh/authorized_keys"
```

Vérification de la connexion par clé (sans mot de passe) :

```
ssh debian@IP_DU_VPS
```

Désactivation du mot de passe et du login root

Sur le VPS, éditer la configuration SSH :

```
sudo nano /etc/ssh/sshd_config
```

Modifier les directives suivantes :

```
PasswordAuthentication no
PermitRootLogin no
```

Puis redémarrer le service :

```
sudo systemctl restart sshd
```

Attention — Toujours tester la connexion dans un NOUVEAU terminal avant de fermer la session active, pour ne pas se retrouver bloqué hors du serveur.

3.2 Étape 2 — Mise à jour du système

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

Vérification des droits sudo :

```
groups
sudo whoami   # doit retourner "root"
```

3.3 Étape 3 — Installation de Docker

Installation via le dépôt officiel Docker (plus fiable que le paquet Debian par défaut) :

```
# Prérequis
sudo apt update && sudo apt install -y ca-certificates curl

# Clé GPG officielle
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Dépôt Docker
echo "deb [arch=$(dpkg --print-architecture) signed-
by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/debian $(.
/etc/os-release && echo "$VERSION_CODENAME") stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null

# Installation
sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin

# Vérification
sudo docker run hello-world
```

Pour utiliser Docker sans sudo :

```
sudo usermod -aG docker debian
# puis se déconnecter/reconnecter
```

3.4 Étape 4 — Configuration du firewall (UFW)

```
sudo apt install -y ufw

# IMPORTANT : autoriser SSH AVANT d'activer le firewall
sudo ufw allow OpenSSH
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp

# Vérifier les règles avant activation
sudo ufw show added

# Activer
sudo ufw enable

# Vérifier
sudo ufw status verbose
```

Attention — Si SSH n'est pas autorisé avant ufw enable, on se coupe soi-même l'accès au serveur.

3.5 Étape 5 — Configuration DNS (zone DNS OVHcloud)

Dans l'espace client OVHcloud > Web Cloud > Noms de domaine > cyrillecode.fr > Zone DNS, créer les enregistrements suivants :

Sous-domaine	Type	Cible
@	A	IP du VPS
www	A	IP du VPS

Attention — OVH crée par défaut des enregistrements A pointant vers une page de parking (213.186.33.5), en plus ou à la place de ceux vers le VPS. Vérifier qu'il n'y a qu'un seul A record par sous-domaine, pointant bien vers l'IP du VPS. Supprimer aussi les éventuels TXT de type "1|www.domaine.fr" liés à l'ancienne redirection parking.

Vérification de la propagation :

```
ping cyrillecode.fr
```

3.6 Étape 6 — Installation de Caddy (reverse proxy + HTTPS automatique)

docker-compose.yml (première version, site de test) :

```
services:
  caddy:
    image: caddy:latest
    restart: unless-stopped
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - ./Caddyfile:/etc/caddy/Caddyfile
      - caddy_data:/data
      - caddy_config:/config

volumes:
  caddy_data:
  caddy_config:
```

Caddyfile (test initial) :

```
cyrillecode.fr {
    respond "Test"
}
```

Lancement :

```
docker compose up -d
docker ps   # vérifier le statut "Up"
```

Caddy obtient automatiquement un certificat HTTPS via Let's Encrypt au premier accès au domaine — aucune configuration manuelle n'est nécessaire, à condition que le DNS soit correctement propagé et que le port 80 soit accessible.

3.7 Étape 7 — Déploiement du portfolio (PHP + fichiers statiques)

Clone depuis GitHub

```
cd /home/debian/apps
git clone https://github.com/USERNAME/REPO.git portfolio
```

Pour un dépôt privé, GitHub exige un Personal Access Token (Settings > Developer settings > Personal access tokens > Tokens classic) à la place du mot de passe lors du clone.

```
git config --global credential.helper store   # pour ne pas re-taper le token à chaque fois
```

Ajout de PHP-FPM

Le portfolio contenant de vraies pages PHP (formulaire de contact send_mail.php, includes/), un conteneur PHP-FPM est nécessaire, pas seulement un file_server.

docker-compose.yml (version finale) :

```
services:
  caddy:
    image: caddy:latest
    restart: unless-stopped
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - ./Caddyfile:/etc/caddy/Caddyfile
      - ./portfolio:/var/www/portfolio
      - caddy_data:/data
      - caddy_config:/config
    depends_on:
      - portfolio-php

  portfolio-php:
    image: php:8.3-fpm
    restart: unless-stopped
    volumes:
      - ./portfolio:/var/www/portfolio

volumes:
  caddy_data:
  caddy_config:
```

Caddyfile (version finale, avec compression et cache) :

```
cyrillecode.fr {
    root * /var/www/portfolio
    encode gzip

    @static {
        path *.css *.js *.png *.jpg *.jpeg *.gif *.svg *.woff *.woff2
    }
    header @static Cache-Control "public, max-age=31536000, immutable"

    php_fastcgi portfolio-php:9000
    file_server
}
```

Attention — Linux est sensible à la casse. Le fichier Index.php (majuscule) n'était pas reconnu comme page d'accueil par défaut — renommé en index.php.

Résolution de conflits de conteneurs

Suite au renommage du dossier caddy/ en apps/, un ancien conteneur (caddy-caddy-1) est resté actif et bloquait le port 80 pour le nouveau (apps-caddy-1) :

```
docker stop caddy-caddy-1
docker rm caddy-caddy-1
docker compose down
docker compose up -d
```

4. Structure / Données

Le serveur héberge tous les projets sous une arborescence unique, orchestrée par un seul fichier docker-compose.yml.

```
/home/debian/apps/
├── docker-compose.yml      ← fichier unique orchestrant TOUS les services
├── Caddyfile                ← routage pour tous les sous-domaines
├── portfolio/                 ← code source du portfolio
├── prism/                     ← (futur) code source PRISM
├── wanderbook/                 ← (futur) code source Wanderbook
└── ...                        ← un sous-dossier par projet
```

Élément	Description	Remarque
docker-compose.yml	Orchestration de tous les services Docker	Un service par projet, tous sur le même réseau Docker interne
Caddyfile	Routage HTTP(S) vers chaque service	Permet à Caddy de router par nom de service
portfolio/	Code source du site cyrillecode.fr	Servi via Caddy + PHP-FPM

5. Déploiement / CI-CD

Le déploiement du portfolio est automatisé via GitHub Actions : à chaque push sur la branche main, le workflow se connecte en SSH au VPS, récupère les changements et redémarre les conteneurs concernés.

5.1 Clé SSH dédiée au déploiement

```
ssh-keygen -t ed25519 -f deploy_key -C "github-actions-deploy" -N '""'
```

Ajout de la clé publique au VPS :

```
echo "CONTENU_DE_deploy_key.pub" >> ~/.ssh/authorized_keys
```

5.2 Secrets GitHub

Repo > Settings > Secrets and variables > Actions :

Secret	Valeur
VPS_HOST	IP du VPS
VPS_USER	debian
VPS_SSH_KEY	Contenu complet de la clé privée (deploy_key)

5.3 Fichier .github/workflows/deploy.yml

```yaml
name: Deploy to VPS

on:
  push:
    branches:
      - main

jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - name: Deploy via SSH
        uses: appleboy/ssh-action@v1.0.3
        with:
          host: ${{ secrets.VPS_HOST }}
          username: ${{ secrets.VPS_USER }}
          key: ${{ secrets.VPS_SSH_KEY }}
          script: |
            cd /home/debian/apps/portfolio
            git pull origin main
            docker compose restart portfolio-php
            docker compose restart caddy
```

Attention — Ne pas modifier les fichiers directement sur le VPS (via WinSCP par exemple) sans les committer sur GitHub, sous peine de conflit lors du prochain git pull automatique.

6. Points de vigilance

Attention — Ne pas confondre l'offre VPS (Bare Metal Cloud > Serveurs Privés Virtuels) avec les offres d'hébergement mutualisé proposées après l'achat du domaine — ces dernières n'ont ni accès root ni Docker.

Attention — "Permission denied" en SSH malgré un mot de passe correct peut venir d'un bannissement fail2ban (sudo fail2ban-client set sshd unbanip TON_IP) ou de l'authentification par mot de passe désactivée par défaut sur l'image OVH. Diagnostic possible via la console de secours OVH (espace client > VPS > Console/KVM), qui contourne SSH.

Attention — Toujours tester la connexion SSH dans un nouveau terminal avant de fermer la session active lors de la désactivation du mot de passe / login root, pour éviter de se retrouver bloqué hors du serveur.

Attention — Toujours autoriser SSH (sudo ufw allow OpenSSH) avant d'activer UFW (sudo ufw enable), sous peine de se couper soi-même l'accès au serveur.

Attention — OVH crée par défaut des enregistrements DNS A vers une page de parking. Vérifier qu'il n'existe qu'un seul A record par sous-domaine, pointant vers l'IP du VPS.

Attention — Linux étant sensible à la casse, un fichier Index.php en majuscule n'est pas reconnu comme page d'accueil par défaut — utiliser index.php.

Attention — Ne jamais modifier les fichiers directement sur le VPS sans les committer sur GitHub : le prochain git pull automatique du workflow de déploiement écraserait ou entrerait en conflit avec ces modifications.

Commandes utiles à retenir

Action	Commande
Voir les conteneurs actifs	docker ps
Voir les logs d'un service	docker compose logs NOM_SERVICE
Redémarrer un service	docker compose restart NOM_SERVICE
Arrêter/relancer tout proprement	docker compose down puis docker compose up -d
Statut du firewall	sudo ufw status verbose
Mise à jour système	sudo apt update && sudo apt upgrade -y