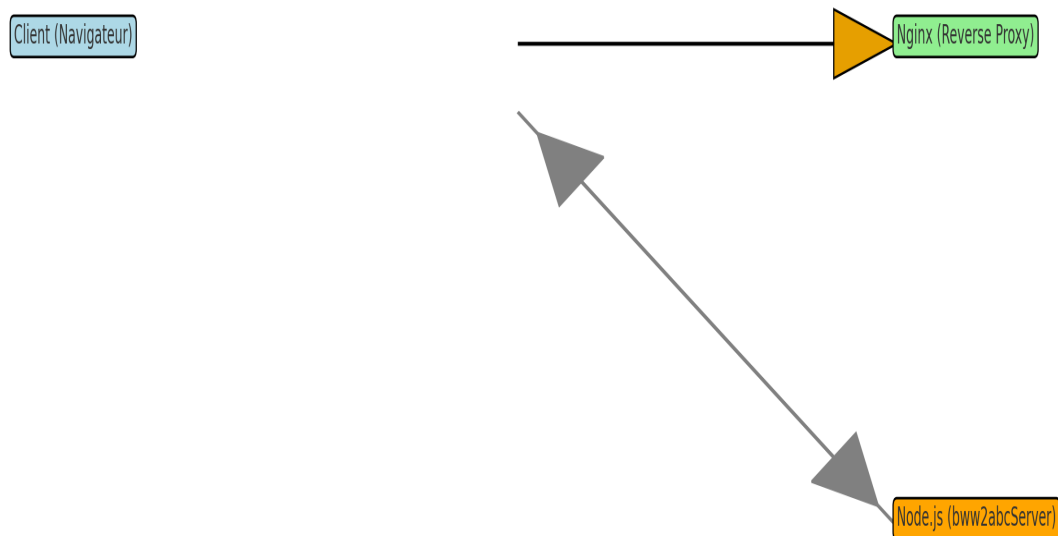


# Guide complet : Mise en ligne de bww2abcServer sur un VPS OVH

Ce document décrit de manière détaillée toutes les étapes nécessaires pour installer et mettre en ligne le serveur **bww2abcServer** (une application Node.js) sur un VPS OVH sous Ubuntu, avec nginx comme reverse proxy et Let's Encrypt pour le HTTPS. Chaque étape est expliquée afin de comprendre le pourquoi des actions.

## Diagramme d'architecture :



## 1. Pré-requis

- Un VPS OVH fonctionnant sous Ubuntu. - Un nom de domaine configuré (exemple : magavel.ovh). - Un sous-domaine pointant vers l'adresse IP publique du VPS (exemple : bww2abc.magavel.ovh).

```
dig A bww2abc.magavel.ovh +short
```

## 2. Installer Node.js avec nvm

Pourquoi nvm ? Les dépôts Ubuntu peuvent devenir obsolètes. nvm (Node Version Manager) permet d'installer et gérer facilement plusieurs versions de Node.js, indépendamment d'APT.

```
curl -fsSL https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh | bash
export NVM_DIR="$HOME/.nvm"
source "$NVM_DIR/nvm.sh"
nvm install --lts
node -v
npm -v
```

## 3. Déployer l'application

On place l'application dans /var/www/ pour respecter la hiérarchie classique des serveurs web.

```
sudo mkdir -p /var/www/bww2abcServer
sudo chown -R ubuntu:ubuntu /var/www/bww2abcServer
```

```
cd /var/www/bww2abcServer
git clone https://github.com/avelFeron/bww2abcServer.git .
npm install
echo 'PORT=3001' > .env
node server.js
curl -I http://127.0.0.1:3001/
```

## 4. Créer un service systemd

Cela permet de lancer automatiquement le serveur Node.js au démarrage du VPS et de le redémarrer en cas de crash.

```
sudo tee /etc/systemd/system/bww2abc.service >/dev/null <<'EOF'
[Unit]
Description=bww2abc Server (Node.js via nvm)
After=network.target

[Service]
WorkingDirectory=/var/www/bww2abcServer
Environment=PORT=3001
ExecStart=/usr/bin/env bash -lc 'source /home/ubuntu/.nvm/nvm.sh && node server.js'
Restart=always
RestartSec=3
User=ubuntu
Group=ubuntu

[Install]
WantedBy=multi-user.target
EOF

sudo systemctl daemon-reload
sudo systemctl enable --now bww2abc
sudo systemctl status bww2abc --no-pager
```

## 5. Configurer nginx comme reverse proxy

nginx sert de proxy entre le client et Node.js (port 3001). Cela permet de gérer TLS/SSL (HTTPS) et d'exposer le service sur le port 80/443 standard.

```
sudo tee /etc/nginx/sites-available/bww2abc.magavel.ovh >/dev/null <<'EOF'
server {
    listen 80;
    listen [::]:80;
    server_name bww2abc.magavel.ovh;

    location / {
        proxy_pass          http://127.0.0.1:3001;
        proxy_http_version  1.1;
        proxy_set_header    Host                $host;
        proxy_set_header    X-Real-IP           $remote_addr;
        proxy_set_header    X-Forwarded-For     $proxy_add_x_forwarded_for;
        proxy_set_header    X-Forwarded-Proto  $scheme;
        proxy_read_timeout  120s;
    }
}
EOF

sudo ln -s /etc/nginx/sites-available/bww2abc.magavel.ovh /etc/nginx/sites-enabled/
sudo nginx -t && sudo systemctl reload nginx
```

## 6. Activer HTTPS avec Let's Encrypt

On utilise certbot pour générer un certificat SSL gratuit et assurer son renouvellement automatique.

```
sudo apt-get install -y certbot python3-certbot-nginx  
sudo certbot --nginx -d bww2abc.magavel.ovh
```

## 7. Vérifications

- Vérifier l'accès : ``curl -I https://bww2abc.magavel.ovh`` - Logs nginx : ``/var/log/nginx/bww2abc.error.log`` - Logs application : ``journalctl -u bww2abc -n 200 -f`` - Vérifier le port local : ``ss -lntp | grep 3001``

## 8. Maintenance

- Mettre à jour le code : ``cd /var/www/bww2abcServer && git pull && npm install && sudo systemctl restart bww2abc`` - Tester le renouvellement SSL : ``sudo certbot renew --dry-run`` - Surveiller l'espace disque : ``df -hT``