Documentation Configuration d'Infrastructure

Aperçu

Cette documentation fournit un guide complet pour la mise en place d'une infrastructure robuste pour une application web.

- Configuration du frontend servi par Apache
- Configuration du backend servi par Nginx
- Configuration du serveur de base de données MySQL
- Gestion des applications Node.js avec PM2
- Configuration SSL
- Règles de pare-feu
- Gestion des accès

Prérequis

- Serveur Ubuntu
- Node.js et npm installés
- PM2 installé globalement (`npm install pm2 -g`)
- Apache et Nginx installés
- MySQL installé

Guide Pas à Pas

1. Configuration des Applications Node.js avec PM2

Démarrage et Gestion des Applications avec PM2

Assurez-vous que vos applications frontend et backend fonctionnent avec PM2 :

```
pm2 start /home/andre/Downloads/HotemManagementMvp/frontend1/
server.js --name frontend
pm2 start /home/andre/Downloads/HotemManagementMvp/backend/in
dex.js --name backend
pm2 save
pm2 startup
pm2 log
pm2 status
pm2 restart backend
pm2 restart frontend
```

2. Configuration d'Apache pour le Frontend

Activer les Modules Apache Nécessaires

```
sudo a2enmod proxy
sudo a2enmod proxy_http
sudo systemctl restart apache2
```

Créer et Modifier la Configuration de l'Hôte Virtuel Apache

Créez un nouveau fichier de configuration pour le frontend :

```
sudo nano /etc/apache2/sites-available/frontend.conf
```

Ajoutez la configuration suivante :

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /home/andre/Downloads/HotemManagementMvp/fro
ntend1/components
    <Directory /home/andre/Downloads/HotemManagementMvp/front
end1/components>
        Options Indexes FollowSymLinks
        AllowOverride All
```

```
        Require all granted
    </Directory>
    ProxyPass / <http://localhost:8080/>
    ProxyPassReverse / <http://localhost:8080/>
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Activer la Configuration du Site

```
sudo a2ensite frontend.conf
sudo systemctl reload apache2
```

3. Configuration de Nginx pour le Backend

Créer et Modifier la Configuration du Bloc Serveur Nginx

Créez un nouveau fichier de configuration pour le backend :

```
sudo nano /etc/nginx/sites-available/backend
```

Ajoutez la configuration suivante :

```
server {
    listen 3067;
    server_name localhost;

    location / {
        proxy_pass <http://localhost:3068>;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
```

```
    }
}
```

Activer la Configuration du Site

```
sudo ln -s /etc/nginx/sites-available/backend /etc/nginx/site
s-enabled/
```

Tester et Redémarrer Nginx

```
sudo nginx -t
sudo systemctl restart nginx
```

4. Configuration du Serveur de Base de Données MySQL

Installer le Serveur MySQL

```
sudo apt update
sudo apt install mysql-server
```

Sécuriser l'Installation MySQL

```
sudo mysql_secure_installation
```

Suivez les invites pour sécuriser votre installation MySQL.

Créer la Base de Données et l'Utilisateur

```
sudo mysql -u root -p
```

Dans le shell MySQL, exécutez les commandes suivantes :

```
CREATE DATABASE hotelmanagement;
CREATE USER 'root'@'localhost' IDENTIFIED BY 'yvan2021';
```

```
GRANT ALL PRIVILEGES ON taskmanage.* TO 'root'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

Importer Votre Schéma de Base de Données

```
cd ~/Desktop
mysql -u root -p taskmanage < hotelmanagement.sql
mysql -u root -p
pass:yvan2021
USE hotelmanagement;
mysql> SELECT * FROM user WHERE email = 'mi@gmail.com';
```

Remplacez `hotelmanagement.sql` par le nom de votre fichier SQL contenant le schéma de la base de données.

5. Configuration SSL pour les Serveurs Frontend et Backend

Générer un Certificat Autofirmé

Pour les serveurs frontend et backend, générez des certificats SSL autofirmés à l'aide de OpenSSL :

```
# Générer un certificat SSL et une clé pour le frontend
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -key
out /etc/ssl/private/frontend.key -out /etc/ssl/certs/fronten
d.crt

# Générer un certificat SSL et une clé pour le backend
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -key
out /etc/ssl/private/backend.key -out /etc/ssl/certs/backend.
crt
```

Configurer Apache pour le Frontend (SSL)

Mettez à jour la configuration de l'hôte virtuel Apache pour le frontend pour activer SSL :

```
<VirtualHost *:443>
    ServerAdmin webmaster@localhost
    DocumentRoot /path/to/frontend/public

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/frontend.crt
    SSLCertificateKeyFile /etc/ssl/private/frontend.key

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Configurer Nginx pour le Backend (SSL)

Mettez à jour la configuration du bloc serveur Nginx pour le backend pour activer SSL :

```
server {
    listen 443 ssl;
    server_name your_backend_domain;

    ssl_certificate /etc/ssl/certs/backend.crt;
    ssl_certificate_key /etc/ssl/private/backend.key;

    location / {
        # Configuration pour servir l'application backend
    }

    error_log /var/log/nginx/backend_error.log;
    access_log /var/log/nginx/backend_access.log;
}
```

Redémarrer les Serveurs Web

Après avoir apporté les modifications de configuration, redémarrez les serveurs Apache et Nginx :

```
# Redémarrer Apache
sudo systemctl restart apache2

# Redémarrer Nginx
sudo systemctl restart nginx
```

6. Configuration du Pare-feu et Gestion des Accès

Configuration du Pare-feu

- **Outil de Pare-feu**: `UFW` (Uncomplicated Firewall)
- **État Actuel du Pare-feu**: Actif
- **Politique par Défaut**: Refuser les connexions entrantes, Autoriser les connexions sortantes
- **Règles Autorisées**:
  - Ports TCP 80 et 443 pour le trafic HTTP et HTTPS
- Port TCP 3068 pour l'accès au service backend
- Autoriser les requêtes ICMP (ping)

```
sudo ufw allow 'Apache Full'
sudo ufw allow 'Nginx Full'
sudo ufw enable
```

Gestion des Accès

- **Ports en Écoute**:
  - Ports 80 et 443 : Trafic HTTP et HTTPS
  - Port 3068 : Service backend

- **Services en Écoute**:
  - Serveurs Apache et Nginx pour l'hébergement web
  - Service backend en écoute sur le port 3068

```
# Vérifier les ports TCP en écoute et les services associés
sudo ss -tuln
```

7. Test

Accéder au Frontend

Ouvrez votre navigateur et accédez à :

```
<http://172.16.147.128/login.html>
```

Vous devriez voir l'application frontend.

Accéder au Backend

Ouvrez votre navigateur et accédez à :

```
<http://172.16.147.128:3067>
```

Vous devriez voir l'application backend.

8. Dépannage

Vérifier les Journaux Apache

En cas de problèmes avec le frontend, vérifiez les journaux Apache :

```
sudo tail -f /var/log/apache2/error.log
```

Vérifier les Journaux Nginx

En cas de problèmes avec le backend, vérifiez les journaux Nginx :

```
sudo tail -f /var/log/nginx/error.log
```

Conclusion

En suivant ce guide, vous allez parcourir toutes les étapes que nous avons suivies pour réaliser notre infrastructure. Vous mettrez en place une infrastructure moderne pour votre application web avec un frontend servi par Apache, un backend servi par Nginx et une base de données MySQL. Les deux applications seront gérées par PM2, garantissant fiabilité et redémarrages automatiques.