

Informe de Implementación del Optimizador de Cobranza en el Servidor VPS

Introducción

El presente informe documenta el proceso de implementación del proyecto "Optimizador de Cobranza" en el servidor proporcionado. Este proyecto se desarrolló con el objetivo de automatizar y optimizar la gestión de cobranza mediante técnicas avanzadas de machine learning. La implementación incluye la configuración de un servidor Linux, integración de componentes de software, y despliegue de modelos predictivos y servicios web.

Detalles del Servidor

- **Host:** vps-4535915-x.dattaweb.com
- **IP:** 149.50.142.4
- **Usuario:** root
- **Puerto SSH:** 5870
- **Sistema Operativo:** Ubuntu 22.04 (64 bits)
- **Aplicación Principal:** Node.js v18.4

Pasos de Implementación

1. Acceso al Servidor

Se estableció conexión al servidor utilizando SSH con las credenciales proporcionadas:

```
ssh -p5870 root@149.50.142.4
```

2. Configuración del Entorno

1. **Actualización del Sistema:** Se actualizó el sistema operativo y los paquetes base:
`sudo apt update && sudo apt upgrade -y`
2. **Instalación de Dependencias:** Se instalaron las herramientas necesarias como Python, pip, Node.js, y otros paquetes requeridos:
`sudo apt install python3 python3-pip nodejs npm git -y`

Configuración del Firewall: Se aseguró que los puertos necesarios estuvieran abiertos:

```
sudo ufw allow 5173/tcp
sudo ufw allow 5870/tcp
sudo ufw allow 8000/tcp
sudo ufw allow 8443/tcp
```

3. `sudo ufw enable`

3. Configuración de Keycloak

Keycloak fue implementado como sistema de autenticación. Se siguieron estos pasos:

Descarga e Instalación:

```
wget
https://downloads.jboss.org/keycloak/26.0.7/keycloak-26.0.7.zip
unzip keycloak-26.0.7.zip
```

1. `cd keycloak-26.0.7/bin`
2. **Inicialización del Servidor Keycloak:**
`/opt/keycloak/bin/kc.sh start --optimized`
3. **Configuración Inicial:** Se creó un usuario administrador y se configuró el cliente necesario para la autenticación del Optimizador de Cobranza.

4. Configuración de la Base de Datos

Se utilizó MongoDB Atlas como base de datos del sistema.

1. **Creación del Cluster en MongoDB Atlas:** Se configuró un clúster gratuito en la plataforma MongoDB Atlas y se obtuvo la URI de conexión.
2. **Configuración del Acceso desde el Servidor:** Se habilitó la IP del servidor (149.50.142.4) en las reglas de acceso de MongoDB Atlas.
3. **Integración con el Backend:** En el archivo de configuración del backend, se incluyó la URI de conexión a MongoDB Atlas.

```
# Conexión a MongoDB
```

```
MONGO_URI =  
"mongodb+srv://martsoto:<password>t@alloxentric.vhl900g.mongo  
db.net/alloxentric?retryWrites=true&w=majority"
```

```
client = MongoClient(MONGO_URI, tls=True,  
tlsAllowInvalidCertificates=False)
```

5. Implementación de los Modelos Predictivos

1. **Despliegue de Modelos:** Los modelos de machine learning desarrollados (K-Means y LSTM) se empaquetaron y desplegaron en el servidor.
2. **Configuración de Unicorn:** Se utilizó Unicorn para correr la aplicación FastAPI:

```
nohup  uvicorn data_base:app --host 0.0.0.0 --port 8000  
--ssl-keyfile  
/etc/letsencrypt/live/vps-4535915-x.dattaweb.com/privkey.pem  
--ssl-certfile  
/etc/letsencrypt/live/vps-4535915-x.dattaweb.com/fullchain.pe  
m
```

6. Pruebas y Monitoreo

1. **Pruebas Funcionales:** Se probaron los endpoints de la API para asegurar que los modelos de machine learning y las funciones de cobranza funcionaran correctamente.
2. **Monitoreo:** Se implementó monitoreo de logs para verificar el funcionamiento continuo: `tail -f uvicorn.log`

Resultados de la Implementación

- El Optimizador de Cobranza está operando en el puerto 8000, accesible mediante la IP 149.50.142.4.
- Se garantiza la seguridad de la autenticación mediante Keycloak.
- La base de datos MongoDB Atlas está correctamente integrada y operativa.

- El sistema permite cargar archivos de cobranza, clasificar acciones, y visualizar reportes de manera eficiente.

Conclusiones y Recomendaciones

1. Conclusiones:

- La implementación fue exitosa, cumpliendo con los requisitos del cliente.
- Los ajustes realizados durante el proceso permitieron una integración eficiente entre todos los componentes.

2. Recomendaciones:

- Implementar una gestión de usuarios más avanzada en futuras versiones.
- Realizar revisiones periódicas del rendimiento del servidor y los modelos para evitar problemas de sobrecarga o desactualización.
- Continuar monitoreando los logs para detectar posibles errores.