



Guide de Déploiement - Application CUS

Ce guide détaille les étapes pour déployer l'application CUS sur différentes plateformes.



Prérequis

1. Configuration Google Cloud

Avant le déploiement, vous devez configurer les APIs Google:

1. Créer un projet Google Cloud

- Aller sur console.cloud.google.com (<https://console.cloud.google.com>)
- Créer un nouveau projet "CUS-App"

2. Activer les APIs nécessaires

- Google Calendar API
- Google Sheets API
- Gmail API

3. Créer des credentials OAuth 2.0

- Dans "APIs & Services" > "Credentials"
- Créer "OAuth 2.0 Client ID"
- Type: Application Web
- Authorized redirect URIs: Ajouter les URLs de votre application

4. Télécharger le fichier credentials.json

- Garder ce fichier en sécurité
- Ne JAMAIS le committer dans git

2. Configuration des Scopes OAuth

Les scopes nécessaires:

```
https://www.googleapis.com/auth/calendar
https://www.googleapis.com/auth/calendar.events
https://www.googleapis.com/auth/spreadsheets
https://www.googleapis.com/auth/gmail.send
```



Option 1: Déploiement sur Streamlit Cloud

Avantages

- Gratuit pour les applications publiques
- Hébergement automatique
- SSL inclus
- Déploiement continu depuis GitHub

Étapes

1. Préparer le repository GitHub

```
# Initialiser git (si nécessaire)
cd /home/ubuntu/code_artifacts/cus-streamlit-app
git init

# Ajouter tous les fichiers
git add .
git commit -m "Initial commit - Application CUS"

# Créer un repository sur GitHub et pusher
git remote add origin https://github.com/VOTRE-USERNAME/cus-streamlit-app.git
git branch -M main
git push -u origin main
```

1. Déployer sur Streamlit Cloud


- Aller sur share.streamlit.io (<https://share.streamlit.io>)
- Se connecter avec GitHub
- Cliquer sur “New app”
- Sélectionner votre repository
- Configuration:
- Repository: `VOTRE-USERNAME/cus-streamlit-app`
- Branch: `main`
- Main file path: `app.py`
- Python version: `3.9`

1. Configurer les Secrets

Dans les paramètres de l'app, section “Secrets”:

```
NOTIFICATION_EMAIL = "seyidebnou@gmail.com"
```

1. Limitation importante

 **Authentification OAuth:** Streamlit Cloud ne supporte pas directement l'authentification OAuth avec les services Google pour les utilisateurs finaux.

Solutions possibles:

a) **Service Backend séparé** (Recommandé pour production):

- Créer une API backend (Flask/FastAPI) hébergée séparément
- Le backend gère l'authentification OAuth
- L'application Streamlit appelle cette API

b) **Service Account** (Pour tests/démo):

- Utiliser un Service Account Google
- Limitations: pas d'interaction utilisateur pour l'auth

c) **Mode démo:**

- L'application actuelle fonctionne en mode simulation
- Parfait pour présenter l'interface
- Remplacer les fonctions de simulation par de vraies APIs pour production

Option 2: Déploiement avec Docker

Avantages

- Environnement isolé
- Portable
- Contrôle total

Étapes

1. Créer un Dockerfile

```
FROM python:3.9-slim

WORKDIR /app

COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt

COPY . .

EXPOSE 8501

CMD ["streamlit", "run", "app.py", "--server.address=0.0.0.0"]
```

1. Construire l'image

```
docker build -t cus-streamlit-app .
```

1. Lancer le conteneur

```
docker run -p 8501:8501 \
-e NOTIFICATION_EMAIL=seyidebnou@gmail.com \
cus-streamlit-app
```

1. Utiliser Docker Compose (recommandé)

Créer `docker-compose.yml` :

```
version: '3.8'

services:
  streamlit-app:
    build: .
    ports:
      - "8501:8501"
    environment:
      - NOTIFICATION_EMAIL=seyidebnou@gmail.com
    volumes:
      - ./data:/app/data
      - ~/.config/abacusai_auth_secrets.json:/root/.config/abacus-
ai_auth_secrets.json:ro
    restart: unless-stopped
```

Lancer:

```
docker-compose up -d
```

Option 3: Déploiement sur VPS (Heroku, DigitalOcean, AWS)

Heroku

1. Créer un fichier Procfile

```
web: streamlit run app.py --server.port=$PORT --server.address=0.0.0.0
```

1. Créer setup.sh

```
mkdir -p ~/.streamlit/
echo "[server]
headless = true
port = $PORT
enableCORS = false
" > ~/.streamlit/config.toml
```

1. Déployer

```
heroku login
heroku create cus-streamlit-app
git push heroku main
```

DigitalOcean / AWS EC2

1. Provisionner un serveur (Ubuntu 20.04+)

2. Installer les dépendances

```
sudo apt update
sudo apt install python3-pip python3-venv nginx -y
```

1. Cloner et configurer l'application

```
git clone https://github.com/VOTRE-USERNAME/cus-streamlit-app.git
cd cus-streamlit-app
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

1. Créer un service systemd

Créer `/etc/systemd/system/cus-app.service` :

```
[Unit]
Description=CUS Streamlit Application
After=network.target

[Service]
User=ubuntu
WorkingDirectory=/home/ubuntu/cus-streamlit-app
Environment="PATH=/home/ubuntu/cus-streamlit-app/venv/bin"
ExecStart=/home/ubuntu/cus-streamlit-app/venv/bin/streamlit run app.py

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

1. Configurer Nginx comme reverse proxy

```
server {
    listen 80;
    server_name votre-domaine.com;

    location / {
        proxy_pass http://localhost:8501;
        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;
    }
}
```

1. Démarrer le service

```
sudo systemctl start cus-app
sudo systemctl enable cus-app
sudo systemctl restart nginx
```

Configuration de la Sécurité

HTTPS avec Let's Encrypt (si déployé sur VPS)

```
sudo apt install certbot python3-certbot-nginx
sudo certbot --nginx -d votre-domaine.com
```

Variables d'environnement sécurisées

Ne jamais mettre de secrets dans le code. Utiliser:

- Streamlit Cloud: Section "Secrets"
- Docker: Variables d'environnement ou Docker secrets
- VPS: Fichier `.env` avec permissions restrictives

```
chmod 600 .streamlit/secrets.toml
```

Monitoring et Logs

Logs Streamlit Cloud

- Disponibles dans le dashboard de l'app

Logs Docker

```
docker logs -f cus-streamlit-app
```

Logs VPS

```
sudo journalctl -u cus-app -f
```

Mises à jour

Streamlit Cloud

- Automatique lors du push sur GitHub

Docker

```
docker-compose down
docker-compose build --no-cache
docker-compose up -d
```

VPS

```
cd /home/ubuntu/cus-streamlit-app
git pull
sudo systemctl restart cus-app
```

Tests avant déploiement

1. Tester localement

```
streamlit run app.py
```

1. Vérifier les dépendances

```
pip check
```

1. Tester les fonctionnalités

- Chat intelligent
- Formulaire de rendez-vous
- Formulaire de propositions
- Notifications

Support

Pour toute question sur le déploiement:

- Email: seyidebnou@gmail.com
 - Documentation Streamlit: docs.streamlit.io (<https://docs.streamlit.io>)
-

Bonne chance pour votre déploiement! 🚀