

Résumé Déploiement Web – ÉMERGENCE (Instance 16/08/2025)

1. Contexte & Objectifs

Projet : ÉMERGENCE – Backend FastAPI multi-agents IA (*Anima*, *Neo*, *Nexus*) avec gestion de documents, débats et dashboard.

Infrastructure cible : Google Cloud Run + HTTPS Load Balancer (domaine emergence-app.ch).

Projet GCP : emergence-469005

Service : emergence-app

Région : europe-west1

Secrets API : OPENAI_API_KEY, GOOGLE_API_KEY, ANTHROPIC_API_KEY, GOOGLE_OAUTH_CLIENT_SECRET (Secret Manager, version sans CR/LF).

Configuration Cloud Run : 2 GiB RAM, 2 CPU, timeout 900s, min-instances=1, ingress = internal-and-cloud-load-balancing.

IAM : rôles attribués correctement (service account + utilisateur développeur).

2. Problèmes identifiés & Corrections appliquées

- Les routes testées sur l'URL Cloud Run `.run.app` renvoyaient 404 car l'ingress est restreint (seul le LB/IAP peut accéder).
- Erreurs Pydantic sur ``openai_api_key``, ``google_api_key``, ``anthropic_api_key`` → secrets existaient mais non mappés. → Correction : ajout version et mapping secrets.
- Erreur ``Empty Google Account OAuth client ID(s)/secret(s)`` côté domaine → l'app attendait plusieurs alias (``GOOGLE_CLIENT_SECRET``, ``OAUTH_GOOGLE_CLIENT_SECRET``). → Correction : mapping supplémentaire de ``GOOGLE_OAUTH_CLIENT_SECRET`` sur ces alias.
- Tentatives d'activation IAP bloquées car projet sans organisation → pas de brand OAuth IAP possible. → Actuellement, domaine accessible sans IAP, avec auth gérée par l'app.

3. Fichiers livrés (inchangés lors de cette instance)

- `src/backend/main.py` – déjà expose ``/api/health``, routers documents, débat, dashboard, websocket.

- Aucun nouveau fichier créé ni déplacé → ARBO inchangée (cf. `arborescence_synchronisée_20250815.txt`).

4. Commandes exécutées

- Ajout de versions aux secrets : OPENAI_API_KEY, GOOGLE_API_KEY, ANTHROPIC_API_KEY.

- Mapping secrets :

`GOOGLE_OAUTH_CLIENT_SECRET=GOOGLE_OAUTH_CLIENT_SECRET:latest`

`GOOGLE_CLIENT_SECRET=GOOGLE_OAUTH_CLIENT_SECRET:latest`

`OAUTH_GOOGLE_CLIENT_SECRET=GOOGLE_OAUTH_CLIENT_SECRET:latest`

- Déploiement révision 00025-vs7 (100% trafic).

- Tests Invoke-WebRequest sur ``https://emergence-app.ch/api/health`` et

``https://www.emergence-app.ch/api/health`` → toujours erreur liée à OAuth interne.

5. État actuel

- L'app démarre correctement (``Application startup complete``).

- Endpoints exposés : ``/api/health``, ``/api/documents``, ``/api/debate``, ``/api/dashboard``, websocket ``chat``.

- Secrets API désormais correctement injectés.

- Problème restant : logique OAuth interne toujours active → l'app attend un flux OAuth complet même quand on tape ``/api/health``.

- IAP n'est pas actif (pas de `client_id` configuré dans le BackendService).

6. Étapes suivantes

- Décider : garder auth interne (OAuth Google) ou basculer en mode IAP-only (si rattachement projet à une organisation).

- Si maintien auth interne → vérifier que `/api/health` soit exempté du flow OAuth (public ou semi-public).
- Côté infra : envisager Cloud Armor (IP allowlist) ou Domain Mapping Cloud Run si on veut se passer du LB/IAP.
- Frontend : corriger référence `debate.css`.