

Résumé Déploiement & Tests Cloud Run – ÉMERGENCE

Version consolidée – Référence pour futurs déploiements

1. Contexte & Objectifs

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

Cible : Déploiement Google Cloud Run.

Ressources :

- Projet GCP : emergence-469005
 - Service : emergence-app
 - Région : europe-west1
 - Secrets API : OPENAI_API_KEY, GOOGLE_API_KEY, ANTHROPIC_API_KEY via Secret Manager
- Configuration Cloud Run : RAM=2 GiB, CPU=2, timeout=900s, CHROMA_TELEMETRY_ENABLED=FALSE
Région par défaut : gcloud config set run/region europe-west1

2. Déploiement & Configuration

IAM :

- roles/secretmanager.secretAccessor → SA Cloud Run
- roles/run.invoker → gonzalefernando@gmail.com
- 14/08 : ajout roles/artifactregistry.reader au SA d'exécution

Commandes :

- Tentative gcloud run deploy --source . → ContainerImageImportFailed
- Build Cloud Build OK → image dans cloud-run-source-deploy mais non importée

URL Service : <https://emergence-app-47nct44nma-ew.a.run.app>

3. Tests Fonctionnels API

/api/health : OK

Module Documents :

- Upload TXT : OK
- Upload PDF invalide → 500 (PyMuPDF FzErrorFormat code=7)
- Upload PDF valide : OK
- CRUD complet : OK
- Nettoyage pending après PDF invalides : OK

Routes disponibles : /api/health, /api/documents, /api/debate/, /api/dashboard/costs/summary

Absents : /api/chat, POST /api/debate/

4. Analyse Problèmes & Résolutions

- PDF invalide → 500 : nécessite validation backend
- OOM initial : corrigé (RAM=2 GiB)
- /api/debate/ 500 : bug DI corrigé localement, pas en prod
- ContainerImageImportFailed : lié IAM/dépôt image
- POST chat/débat manquant

5. Étapes Suivantes

Backend : Validation PDF stricte, corriger /api/debate/, ajouter POST /api/debate/

Déploiement : Vérifier URI image, droits artifactregistry.reader, build local + push

Tests : CRUD complet documents/débats, smoke tests /api/health et /api/debate/

Optimisations : Init lazy embeddings, --min-instances=1

6. Script PowerShell Utilitaire IAM et Logs

Pour automatiser certaines opérations récurrentes sur le service Cloud Run, un script PowerShell a été u

Variables globales :

```
$PROJECT = "emergence-469005"
```

```
$REGION = "europe-west1"
```

```
$SERVICE = "emergence-app"
```

A. Donner accès public authentifié :

```
gcloud run services add-iam-policy-binding $SERVICE `
```

```
--region $REGION `
```

```
--member="allAuthenticatedUsers" `
```

```
--role="roles/run.invoker"
```

Résultat : Ajout de allAuthenticatedUsers et de l'utilisateur gonzalefernando@gmail.com comme invokers

B. Lecture des logs du service :

```
gcloud run services logs read $SERVICE `
```

```
--region $REGION `
```

```
--limit=100
```

Exemple de logs significatifs :

- Initialisation SessionManager, CostTracker, SentenceTransformer
- Chargement des prompts des agents Anima, Neo, Nexus
- Erreurs ChromaDB telemetry (posthog)
- Succès upload TXT
- Erreur parsing PDF invalide (PyMuPDF FzErrorFormat code=7)
- Suppression document OK
- Erreur 500 sur PDF corrompu (backend)

C. Vérifier la révision active :

```
gcloud run services describe $SERVICE `
```

```
--region $REGION `
```

```
--format="value(status.latestReadyRevisionName)"
```

Retour : emergence-app-00009-22m