

# CV

## Jérémy Goutin

**Architecte de solutions · 10+ ans d'expérience · 20+ projets réussis**

Expert freelance en logiciel, cloud et DevOps.

---

E-mail	<a href="mailto:contact@jgoutin.dev">contact@jgoutin.dev</a>
Site web	<a href="http://jgoutin.dev">jgoutin.dev</a>
LinkedIn	<a href="https://linkedin.com/in/jgoutin">linkedin.com/in/jgoutin</a>
GitHub	<a href="https://github.com/jgoutin">github.com/jgoutin</a>

---

## Compétences clés & expertise

### Architecture cloud

**Expert AWS** - Audit de sécurité externe réussi

#### Points Forts :

- **Serverless & Traditionnel** : Architecture complète pour les deux paradigmes
- **Infrastructure as Code** : Expert Terraform pour déploiements reproductibles
- **Sécurité d'Abord** : IAM au moindre privilège, pare-feu VPC, SecurityHub
- **Optimisation des Coûts** : Bonnes pratiques FinOps, architecture multi-compte

**Services AWS Maîtrisés** : VPC, EC2, ELB, RDS/Aurora, Lambda, DynamoDB, CloudFront, CloudWatch, SQS, SNS, SES, Cognito, Route53, S3, ECS, IAM

**Multi-Cloud** : Microsoft Azure (AAD, MS365, Azure DevOps), OpenStack, Alibaba Cloud

## Développement logiciel

**Expert Python** - Plus de 10 ans d'expérience

### Points Forts :

- **Architecture & Conception** : Architecte logiciel pour plusieurs projets d'entreprise
- **Développement Web Moderne** : FastAPI, SQLAlchemy, PostgreSQL, DynamoDB
- **Calcul Scientifique** : Numpy, Scipy, Pandas, Matplotlib, Cython
- **Focus Qualité** : Code propre, tests automatisés, maintenabilité long terme
- **Développement Augmenté par l'IA** : Exploitation d'outils IA pour productivité accrue

**Approche** : Construction de logiciels évolutifs et maintenables qui croissent avec les besoins business

## DevOps & CI/CD

### Points Forts :

- **Automatisation Complète** : Du push de code au déploiement en production
- **DevSecOps** : Analyse de sécurité intégrée dans les pipelines CI/CD
- **GitOps** : Gestion de l'infrastructure et déploiements via Git
- **Multi-Plateforme** : GitHub Actions, GitLab CI/CD, Azure Pipeline, AWS CodeBuild
- **Gestion de Configuration** : Ansible, Docker

**Résultat** : Déploiements sans intervention avec contrôles qualité et sécurité intégrés

## Intelligence artificielle & LLM

### Points Forts :

- **LLMOps** : Déploiement et gestion de modèles de langage en production
- **Plateformes IA d'Entreprise** : Architecture complète avec gouvernance des données
- **Systèmes RAG** : Retrieval-Augmented Generation pour intelligence documentaire
- **AWS Bedrock** : Accès et intégration multi-modèles
- **Solutions IA Personnalisées** : Chatbots, assistants de développement, workflows automatisés

**Focus :** Implémentations IA sécurisées et prêtes pour la production avec confidentialité des données

## **Administration système**

**Expertise Linux :** Fedora, CentOS, Debian, Ubuntu, Alpine Linux

**Compétences Clés :** Automatisation Ansible, durcissement & sécurité Linux, pare-feu PfSense, stockage TrueNAS

## **Compétences professionnelles**

**Techniques :** Architecture de solutions complètes, résolution de problèmes complexes, respect des bonnes pratiques, apprentissage rapide **Savoir-être :** Haute autonomie, esprit critique, force de proposition, rigueur méthodologique

---

## **Projets significatifs**

### **stdapi.ai : API Compatible OpenAI pour les Services IA d’AWS (Août 2025-Aujourd’hui)**

**Contexte :** Projet Freelance **Réalisations :**

- Création d’une API open source permettant la compatibilité des SDK OpenAI avec les services IA d’AWS et les modèles Amazon Bedrock
- Réalisation complète du projet couvrant la conception d’architecture, le développement full-stack, le déploiement d’infrastructure AWS, la rédaction de documentation technique et la stratégie de mise sur le marché
- Couche de compatibilité établissant un pont entre les SDK OpenAI et plus de 80 modèles Amazon Bedrock ainsi que plusieurs services IA d’AWS
- Implémentation technique intégrant **Amazon Bedrock** pour l’orchestration des LLM, **Amazon Polly** pour la synthèse vocale, **Amazon Transcribe** pour la transcription audio et **Amazon Translate**
- Capacités multi-modales couvrant les conversations textuelles, la génération d’images, le traitement audio et les embeddings vectoriels
- Infrastructure déployée sur AWS avec architecture multi-régions, observabilité **CloudWatch** et images de conteneurs durcies
- Modèle de double licence (AGPL-3.0 et licence commerciale AWS Marketplace) établi pour équilibrer contribution à la communauté open source et viabilité en entreprise

Site du projet : <https://stdapi.ai> | GitHub : <https://github.com/stdapi-ai/stdapi.ai>

**Technologies** : AWS (Bedrock, Lambda, ECS, CloudWatch), Python, FastAPI, API OpenAI, IA Multi-modale

## **Plateforme IA/LLM d'Entreprise (Juillet-Août 2025)**

**Contexte** : Projet Freelance

**Réalisations** :

- Création d'une solution centralisée pour exploiter des modèles LLM avec gouvernance des données
- Déploiement d'une plateforme LLM sur l'infrastructure AWS
- Développement d'un chatbot comme point d'accès unique et contrôlé aux capacités d'IA
- Intégration de **RAG** (Retrieval-Augmented Generation) pour enrichir les réponses avec des documents internes
- Intégration IDE pour permettre aux équipes de développement de générer et améliorer le code
- Utilisation d'**AWS Bedrock** pour l'accès à un portefeuille varié de modèles de langage
- Gouvernance stricte pour protéger les données de l'entreprise

**Technologies** : AWS Bedrock, Python, RAG, LLM, IA

## **Migration de Runners GitLab vers Architecture Serverless AWS (Mars-Juin 2025)**

**Contexte** : Projet Freelance

**Réalisations** :

- Migration complète de l'infrastructure de CI/CD vers une architecture serverless AWS
- Déploiement de runners GitLab basés sur **ECS Fargate ARM** pour les tâches génériques (Terraform, curl, linters)
- Implémentation de runners GitLab exploitant **AWS CodeBuild** pour les tâches de compilation
- Abandon des access keys IAM statiques au profit de **rôles IAM temporaires** à moindres privilèges
- Centralisation et sécurisation de la gestion des secrets (registres Docker, dépôts Maven et NPM)
- Support natif de l'architecture **ARM64** pour la construction d'images Docker
- Amélioration notable des performances et réduction des temps d'attente
- Scalabilité à la demande éliminant les congestions de jobs
- Optimisation significative des coûts opérationnels

**Technologies** : GitLab CI/CD, AWS (ECS Fargate, CodeBuild, IAM), Docker, ARM64

## **Standardisation d'Infrastructure AWS avec Terraform (Octobre 2023-Octobre 2024)**

**Contexte** : Projet Freelance

**Réalisations** :

- Développement d'architectures de référence via des modules Terraform réutilisables
- Migration d'architectures historiques EC2 vers des services managés AWS (ECS, Aurora, Lambda, SQS, SES)
- Création de modules avec interface épurée encapsulant une configuration avancée et complexe
- Intégration native des bonnes pratiques de sécurité (IAM moindres privilèges, security groups, chiffrement, monitoring)
- Réduction notable du temps de mise en place de nouvelles applications
- Amélioration de la fiabilité par la standardisation
- Simplification de la maintenance des infrastructures
- Adoption des processus DevOps par les équipes de développement

**Technologies** : Terraform, AWS (ECS, Aurora, Lambda, SQS, SES, IAM), Infrastructure as Code

## **Architecture Réseau AWS Multi-Compte avec VPC Partagé (Octobre 2023-Avril 2024)**

**Contexte** : Projet Freelance

**Réalisations** :

- Conception et implémentation d'un VPC commun multi-compte **Dualstack (IPv4/IPv6)**
- Intégration de **AWS Network Firewall, Route53 Resolver Firewall, AWS Site-to-Site VPN**
- Optimisation **FinOps** avec gestion simplifiée et centralisée
- Configuration de **VPC Endpoints** pour sécuriser les communications
- Architecture permettant une scalabilité et une sécurité optimales

**Technologies** : AWS VPC, Terraform, Network Firewall, Route53, VPN

## **Marketplace FPGA avec Architecture Cloud Native (Novembre 2021-Avril 2023)**

**Contexte** : Accelize - Architecture & Développement

**Réalisations** :

**Architecture Cloud** :

- Service web haute disponibilité avec backend EC2 multi-AZ
- Frontend **Angular** avec **S3** et **CloudFront**
- Microservices **serverless** basés sur **Lambda**
- Authentification utilisateurs avec **Cognito**
- Déploiement totalement automatisé via CI/CD

**Architecture Logicielle** :

- Conception complète des modèles de données SQL (**PostgreSQL**) et NoSQL (**DynamoDB**)
- Développement du backend principal en Python avec **FastAPI** et **SQLAlchemy Core**
- Conception et développement de microservices Python
- Définition de toutes les API internes et externes (REST)
- Stratégies de tests et monitoring
- Configuration Linux optimisée pour les serveurs EC2

**Technologies** : AWS (EC2, Lambda, S3, CloudFront, Cognito, DynamoDB, RDS), Python, FastAPI, SQLAlchemy, PostgreSQL, Terraform

## **Service d'Exécution d'Applications FPGA Conteneurisées (Juin 2021-Octobre 2021)**

**Contexte** : Accelize - Architecture Cloud

**Réalisations** :

- Architecture **serverless** pour l'exécution de démos FPGA publiques
- Provisionnement et résiliation automatique des instances FPGA sur **AWS** et **OpenStack**
- Sécurisation de l'exécution Docker (isolation, sandboxing)
- Utilisation de **Lambda**, **CloudFront**, **S3** pour l'infrastructure
- Gestion automatique du cycle de vie des ressources en fonction de la demande

**Technologies** : AWS (Lambda, CloudFront, S3, EC2), OpenStack, Docker, Python

## **Gestionnaire de Dépôts Linux Hautement Disponible (Octobre 2021-Mars 2022)**

**Contexte** : Accelize - Architecture Cloud **Réalisations** :

- Architecture serverless pour héberger des dépôts **Debian** et **Red Hat**
- Distribution via **S3** et **CloudFront** avec haute disponibilité
- Ajout de paquets automatisé via CI/CD (paquets internes)
- Service web pour les partenaires (upload de paquets externes)
- Mise à jour automatique des métadonnées et signatures GPG
- Utilisation de **Lambda** pour le traitement asynchrone

**Technologies** : AWS (S3, Lambda, CloudFront), Python, CI/CD, GPG

## **Migration et Administration d'Infrastructure Cloud Microsoft (Avril 2021-Juin 2021)**

**Contexte** : Accelize - Administration Système **Réalisations** :

- Architecture complète basée sur **Microsoft Azure, AAD, MS365**
- **Single Sign-On (SSO)** entre tous les services (AWS, GitHub, MS365)
- Migration de **Google Workspace** vers **MS365**
- Gestion automatisée des laptops Windows (provisionnement, configuration, sécurité)
- Administration **Azure DevOps** et gestion des repositories

**Technologies** : Microsoft Azure, AAD, MS365, AWS, GitHub, PowerShell

## **ACID : Agents Cloud Dynamiques pour Azure Pipelines (Juin 2021)**

**Contexte** : Accelize - DevOps **Réalisations** :

- Exécution de jobs Azure Pipelines sur des agents éphémères provisionnés à la demande sur AWS EC2 et Azure VM
- Développement d'un utilitaire s'appuyant sur **Terraform** pour le provisionnement et la suppression automatisés des ressources
- Configuration logicielle des agents assurée par des playbooks **Ansible** pour une personnalisation complète
- Utilisation d'instances spot intégrée pour minimiser les coûts

**Technologies** : Azure Pipelines, AWS EC2, Azure VM, Terraform, Ansible

## **Modernisation de l'Infrastructure et Sécurité Cloud AWS (Janvier 2019-Janvier 2020)**

**Contexte** : Accelize - Architecture Cloud **Réalisations** :

- Refonte complète de l'infrastructure via Infrastructure as Code avec **Terraform**
- Révision stricte des politiques AWS IAM, segmentation réseau via VPCs
- Intégration de **Security Hub** pour surveillance centralisée des menaces
- Infrastructure entièrement gérée et versionnée via Terraform
- **Passage réussi de l'audit de sécurité externe** avec d'excellents résultats

**Technologies** : AWS (VPC, EC2, RDS, S3, Lambda, Security Hub), Terraform

## **Environnement de Développement AWS Sécurisé (Juin 2019-Juillet 2019)**

**Contexte** : Accelize - Architecture Cloud **Réalisations** :

- Conception d'un environnement de développement interne multi-utilisateurs
- Système d'**ownership des ressources** pour la traçabilité
- Politiques IAM à **moindres privilèges** pour chaque développeur
- Gestion automatique des coûts avec résiliation des ressources orphelines via **Lambda** et **EventBridge**
- Sauvegardes automatiques des instances de développement avec **DLM**

**Technologies** : AWS (IAM, EC2, CloudWatch, Lambda, EventBridge, DLM), Terraform, Python

## **Accelpy : Déploiement d'Applications FPGA (Juillet 2019-Octobre 2019)**

**Contexte** : Accelize - Développement Logiciel **Réalisations** :

- Outil d'automatisation pour le provisionnement et le déploiement d'applications FPGA sur infrastructures cloud et on-premise
- Outil en ligne de commande orchestrant le déploiement de solutions matérielles accélérées par FPGA
- Interaction avec les API de la plateforme pour gérer le cycle de vie des designs FPGA
- Provisionnement automatisé des ressources sur instances FPGA cloud ou serveurs on-premise

- Téléchargement sécurisé des bitstreams, programmation des puces FPGA, et mise en place des environnements logiciels hôtes

**Technologies** : Python, AWS, OpenStack, FPGA, CLI

## **Apyfal : Déploiement Cloud d'Applications FPGA (Avril 2018-Avril 2019)**

**Contexte** : Accelize - Développement Logiciel **Réalisations** :

- Développement d'Apyfal, une solution logicielle facilitant l'accélération de calculs sur des FPGA disponibles dans le cloud
- Client Python capable de piloter à distance le cycle de vie complet d'une application
- API RESTful pour la communication entre le client et le serveur d'orchestration
- Gestion dynamique des ressources FPGA, incluant la programmation des bitstreams et l'allocation des instances

**Technologies** : Python, AWS, OpenStack, FPGA, REST API

## **Airfs (Pycosio) : Bibliothèque Python Unifiée pour le Stockage Cloud (Juillet 2018-Février 2021)**

**Contexte** : Open Source - Développement Logiciel **Réalisations** :

- Interface de programmation unifiée pour interagir avec divers systèmes de stockage distants et cloud
- Implémentation des classes abstraites "io.RawIOBase" et "io.BufferedIOBase" pour compatibilité native
- Fonctionnalités avancées : écriture asynchrone, prefetching, verrouillage basé sur la mémoire, connexions parallèles
- Support de multiples fournisseurs : **AWS S3, Azure Blob Storage, Azure Files, OpenStack Swift**, HTTP/HTTPS
- Initialement créé sous le nom "pycosio", repris sous forme de fork pour extension et maintenance

**Technologies** : Python, AWS S3, Azure Storage, OpenStack Swift

## **Ansible Home : Collection Ansible pour Services Auto-Hébergés (Octobre 2019-Octobre 2021)**

**Contexte** : Open Source - DevOps **Réalisations** :

- Développement d'une collection Ansible pour l'auto-hébergement de logiciels libres avec sécurité renforcée
- Rôles spécialisés pour installation automatisée : **Nextcloud, Squid, Kodi, MPD**

- Rôles de dépendances modulaires : **Nginx, PostgreSQL, PHP-FPM, Valkey**
- Rôle “common” centralisant l’initialisation système : pare-feu, durcissement **SELinux**, mises à jour automatiques, sécurisation SSH
- Workflow CI/CD avec **GitHub Actions** pour validation et déploiement

**Technologies** : Ansible, Fedora Linux, GitHub Actions, PostgreSQL, Nginx

## **Compilertools : Paquets Binaires Python Hautes Performances (Février 2017-Décembre 2017)**

**Contexte** : Open Source - Développement Logiciel **Réalisations** :

- Solution complète pour la compilation d’extensions C et C++ en paquets Python binaires
- Détection et utilisation des jeux d’instructions avancés des processeurs (**SIMD, AVX, SSE**) pour binaires optimisés
- Intégration avec les outils de packaging Python standards pour distribution sur PyPI
- Gains significatifs en vitesse d’exécution pour applications de calcul intensif

**Technologies** : Python, C/C++, SIMD, PyPI

## **Fazpy : Logiciel d’Analyse de Mesures Optiques (Octobre 2014-Septembre 2017)**

**Contexte** : Thales SESO - Développement Logiciel **Réalisations** :

- Application de bureau complète sous Windows avec Python
- Interface utilisateur développée avec le framework **Qt** pour ingénieurs en optique et mécanique
- Modules de calculs optiques avancés et de traitement d’image
- Optimisation des performances nécessitant optimisation d’algorithmes pour grands volumes de données
- Architecture modulaire et évolutive intégrant **70+ modules fonctionnels**
- Automatisation et fiabilisation de l’analyse des mesures, lien direct entre contrôle qualité et fabrication

**Technologies** : Python, Qt, Numpy, Scipy, Cython, Windows

# Logiciel de Banc de Test pour Cartes Électroniques (Octobre 2017-Avril 2018)

**Contexte** : SuperSonic Imagine - Développement Logiciel

**Réalisations** :

- Développement d'un logiciel de test électronique sous **Debian**
- Architecture **client/serveur** pour contrôle distant et collecte de données
- Instrumentation logicielle pour le pilotage et la communication avec équipements de mesure
- Optimisation de calculs avec **Numpy** pour grands jeux de données
- Implémentation de la méthode **SPC** (Statistical Process Control)
- Scénarios de test automatisés

**Technologies** : Python, Debian, Numpy, Communication série/  
TCP

---

## Contributions open source

Contributeur actif sur divers projets open source disponibles sur GitHub :

- Développement et maintenance de bibliothèques Python
- Contribution à des projets tiers
- Création d'outils pour la communauté

<https://github.com/jgoutin>

---

## Langues

- **Français** : Natif
  - **Anglais** : Professionnel à l'écrit, intermédiaire à l'oral
  - **Allemand** : Notions de base
- 

*JGoutin-dev SARL - 994495422 R.C.S. Aix-en-Provence, France*

© Tous droits réservés.