Resume

Jérémy Goutin

Solutions Architect · 10+ years of experience · 20+ successful projects

Freelance expert in software, cloud, and DevOps.

Email	contact@jgoutin.dev
Website	jgoutin.dev
LinkedIn	linkedin.com/in/jgoutin
GitHub	github.com/jgoutin

Key skills & expertise

Cloud architecture

AWS Expert - Passed external security audit

Key Strengths:

- **Serverless & Traditional**: Complete solution architecture for both paradigms
- **Infrastructure as Code**: Terraform expert for reproducible deployments
- **Security-First**: Least-privilege IAM, VPC firewall, SecurityHub
- **Cost Optimization**: FinOps best practices, multi-account architecture

Mastered AWS Services: 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

Software development

Python Expert - 10+ years experience

Key Strengths:

- **Architecture & Design**: Software architect for multiple enterprise projects
- **Modern Web Development**: FastAPI, SQLAlchemy, PostgreSQL, DynamoDB
- **Scientific Computing**: Numpy, Scipy, Pandas, Matplotlib, Cython
- **Quality Focus**: Clean code, automated testing, long-term maintainability
- **AI-Augmented Development**: Leveraging AI tools for enhanced productivity

Approach: Building scalable, maintainable software that grows with business needs

DevOps & CI/CD

Key Strengths:

- **Full Pipeline Automation**: From code push to production deployment
- **DevSecOps**: Security scanning integrated in CI/CD pipelines
- **GitOps**: Infrastructure and deployment management via Git
- **Multi-Platform**: GitHub Actions, GitLab CI/CD, Azure Pipeline, AWS CodeBuild
- **Configuration Management**: Ansible, Docker

Result: Zero-touch deployments with built-in security and quality gates

Artificial intelligence & LLM

Key Strengths:

- **LLMOps**: Production deployment and management of language models
- **Enterprise AI Platforms**: Complete architecture with data governance
- **RAG Systems**: Retrieval-Augmented Generation for document intelligence
- **AWS Bedrock**: Multi-model access and integration
- **Custom AI Solutions**: Chatbots, development assistants, automated workflows

Focus: Secure, production-ready AI implementations with data privacy

System administration

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

Key Skills: Ansible automation, Linux hardening & security, PfSense firewall, TrueNAS storage

Professional competencies

Technical: Complete solution architecture, complex problem solving, best practices compliance, fast learner **Soft Skills:** High autonomy, critical thinking, proactive approach, rigorous methodology

---

Significant projects

stdapi.ai: OpenAI-Compatible API for AWS AI Services (August 2025-Present)

Context: Freelance Project **Achievements**:

- Creation of an open-source API enabling OpenAI SDK compatibility with AWS AI services and Amazon Bedrock models
- End-to-end project realization covering architecture design, full-stack development, AWS infrastructure deployment, technical documentation writing, and go-to-market strategy
- Compatibility layer bridging OpenAI SDKs with 80+ Amazon Bedrock models and multiple AWS AI services
- Technical implementation integrating **Amazon Bedrock** for LLM orchestration, **Amazon Polly** for speech synthesis, **Amazon Transcribe** for speech-to-text, and **Amazon Translate**
- Multi-modal capabilities spanning text conversations, image generation, audio processing, and vector embeddings
- Infrastructure deployed on AWS with multi-region architecture, **CloudWatch** observability, and hardened container images
- Dual licensing model (AGPL-3.0 and AWS Marketplace commercial license) established to balance open-source community contribution and enterprise viability

Project website: https://stdapi.ai | GitHub: https://github.com/stdapi-ai/stdapi.ai

Technologies: AWS (Bedrock, Lambda, ECS, CloudWatch), Python, FastAPI, OpenAI API, Multi-modal AI

Enterprise AI/LLM Platform (July-August 2025)

Context: Freelance Project **Achievements**:

- Creation of a centralized solution to exploit LLM models with data governance
- Deployment of an LLM platform on AWS infrastructure
- Development of a chatbot as a single, controlled access point to AI capabilities
- Integration of **RAG** (Retrieval-Augmented Generation) to enrich responses with internal documents
- IDE integration to enable development teams to generate and improve code
- Use of **AWS Bedrock** for access to a varied portfolio of language models
- Strict governance to protect company data

Technologies: AWS Bedrock, Python, RAG, LLM, AI

GitLab Runners Migration to AWS Serverless Architecture (March-June 2025)

Context: Freelance Project
Achievements:

- Complete migration of CI/CD infrastructure to AWS serverless architecture
- Deployment of GitLab runners based on **ECS Fargate ARM** for generic tasks (Terraform, curl, linters)
- Implementation of GitLab runners leveraging **AWS CodeBuild** for compilation tasks
- Abandonment of static IAM access keys in favor of **temporary least privileged IAM roles**
- Centralization and securing of secrets management (Docker registries, Maven and NPM repositories)
- Native support for **ARM64** architecture for Docker image building
- Notable performance improvement and reduction of waiting times
- On-demand scalability eliminating job congestion
- Significant optimization of operational costs

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

AWS Infrastructure Standardization with Terraform (October 2023-October 2024)

Context: Freelance Project
Achievements:

- Development of reference architectures via reusable Terraform modules
- Migration from legacy EC2 architectures to AWS managed services (ECS, Aurora, Lambda, SQS, SES)
- Creation of modules with clean interface encapsulating advanced and complex configuration
- Native integration of security best practices (least privileged IAM, security groups, encryption, monitoring)
- Notable reduction in time to set up new applications
- Improved reliability through standardization
- Simplified infrastructure maintenance
- Adoption of DevOps processes by development teams

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

Multi-Account AWS Shared VPC Architecture (October 2023-April 2024)

Context: Freelance Project
Achievements:

- Design and implementation of a multi-account **Dualstack (IPv4/IPv6)** shared VPC
- Integration of **AWS Network Firewall**, **Route53 Resolver Firewall**, **AWS Site-to-Site VPN**
- **FinOps** optimization with simplified and centralized management
- Configuration of **VPC Endpoints** to secure communications
- Architecture enabling optimal scalability and security

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

FPGA Marketplace with Cloud-Native Architecture (November 2021-April 2023)

Context: Accelize - Architecture & Development **Achievements**:

Cloud Architecture:

- High availability web service with multi-AZ EC2 backend
- **Angular** frontend with **S3** and **CloudFront**
- **Serverless** microservices based on **Lambda**

- User authentication with **Cognito**
- Fully automated deployment via CI/CD

Software Architecture:

- Complete design of SQL (**PostgreSQL**) and NoSQL (**DynamoDB**) data models
- Main backend development in Python with **FastAPI** and **SQLAlchemy Core**
- Python microservices design and development
- Definition of all internal and external APIs (REST)
- Testing and monitoring strategies
- Optimized Linux configuration for EC2 servers

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

Containerized FPGA Application Execution Service (June 2021-October 2021)

Context: Accelize - Cloud Architecture **Achievements**:

- **Serverless** architecture for executing public FPGA demos
- Automatic provisioning and termination of FPGA instances on **AWS** and **OpenStack**
- Securing Docker execution (isolation, sandboxing)
- Use of **Lambda**, **CloudFront**, **S3** for infrastructure
- Automatic resource lifecycle management based on demand

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

Highly Available Linux Repository Manager (October 2021-March 2022)

Context: Accelize - Cloud Architecture **Achievements**:

- Serverless architecture to host **Debian** and **Red Hat** repositories
- Distribution via **S3** and **CloudFront** with high availability
- Automated package addition via CI/CD (internal packages)
- Web service for partners (external package upload)
- Automatic metadata and GPG signature updates
- Use of **Lambda** for asynchronous processing

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

Microsoft Cloud Infrastructure Migration (April 2021-June 2021)

Context: Accelize - System Administration **Achievements**:

- Complete architecture based on **Microsoft Azure**, **AAD**, **MS365**
- **Single Sign-On (SSO)** between all services (AWS, GitHub, MS365)
- Migration from **Google Workspace** to **MS365**
- Automated Windows laptop management (provisioning, configuration, security)
- **Azure DevOps** administration and repository management

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

ACID: Dynamic Cloud Agents for Azure Pipelines (June 2021)

Context: Accelize - DevOps **Achievements**:

- Execution of Azure Pipelines jobs on ephemeral agents provisioned on-demand on AWS EC2 and Azure VM
- Development of a utility leveraging **Terraform** for automated resource provisioning and deletion
- Agent software configuration handled by **Ansible** playbooks for complete execution environment customization
- Spot instance usage integrated to minimize costs

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

AWS Infrastructure Security Modernization (January 2019-January 2020)

Context: Accelize - Cloud Architecture **Achievements**:

- Complete infrastructure overhaul via Infrastructure as Code with **Terraform**
- Strict review of AWS IAM policies, network segmentation via VPCs
- Integration of **Security Hub** for centralized threat monitoring
- Entire infrastructure managed and versioned via Terraform
- **Successfully passed external security audit** with excellent results

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

Secure AWS Development Environment (June 2019-July 2019)

Context: Accelize - Cloud Architecture **Achievements**:

- Design of a multi-user internal development environment
- **Resource ownership** system for traceability
- **Least privileged** IAM policies for each developer
- Automatic cost management with orphaned resource termination via **Lambda** and **EventBridge**
- Automatic backups of development instances with **DLM**

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

Accelpy: FPGA Application Deployment (July 2019-October 2019)

Context: Accelize - Software Development **Achievements**:

- Automation tool for provisioning and deploying FPGA applications on cloud and on-premise infrastructures
- Command-line tool orchestrating deployment of FPGA-accelerated hardware solutions
- Interaction with platform APIs to manage FPGA design lifecycle
- Automated resource provisioning on FPGA instances in the cloud or on-premise servers
- Secure bitstream download, FPGA chip programming, and host software environment setup

Technologies: Python, AWS, OpenStack, FPGA, CLI

Apyfal: Cloud FPGA Application Deployment (April 2018-April 2019)

Context: Accelize - Software Development **Achievements**:

- Development of Apyfal, a software solution facilitating computation acceleration on cloud-available FPGAs
- Python client capable of remotely controlling complete application lifecycle
- RESTful API for communication between client and orchestration server
- Dynamic FPGA resource management, including bitstream programming and instance allocation

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

Airfs (Pycosio): Unified Python Library for Cloud Storage (July 2018-February 2021)

Context: Open Source - Software Development **Achievements**:

- Unified programming interface to interact with various remote and cloud storage systems
- Implementation of “io.RawIOBase” and “io.BufferedIOBase” abstract classes for native compatibility
- Advanced features: asynchronous writing, prefetching, memory-based locking, parallel connections
- Support for multiple providers: **AWS S3**, **Azure Blob Storage**, **Azure Files**, **OpenStack Swift**, HTTP/HTTPS
- Initially created as “pycosio”, taken over as a fork for extension and maintenance

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

Ansible Home: Ansible Collection for Self-Hosted Software (October 2019-October 2021)

Context: Open Source - DevOps **Achievements**:

- Development of an Ansible collection for self-hosting free software with enhanced security
- Specialized roles for automated installation: **Nextcloud**, **Squid**, **Kodi**, **MPD**
- Modular dependency roles: **Nginx**, **PostgreSQL**, **PHP-FPM**, **Valkey**
- “Common” role centralizing system initialization: firewall, **SELinux** hardening, automatic updates, SSH hardening
- CI/CD workflow with **GitHub Actions** for validation and deployment

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

Compilertools: High-Performance Python Binary Packages (February 2017-December 2017)

Context: Open Source - Software Development **Achievements**:

- Complete solution for compiling C and C++ extensions into Python binary packages
- Detection and use of advanced processor instruction sets (**SIMD**, **AVX**, **SSE**) for optimized binaries
- Integration with standard Python packaging tools for PyPI distribution
- Significant execution speed gains for compute-intensive applications

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

Fazpy: Optical Measurement Analysis Software (October 2014-September 2017)

Context: Thales SESO - Software Development **Achievements**:

- Complete desktop application under Windows using Python
- User interface developed with **Qt** framework for optical and mechanical engineers
- Advanced optical calculations and image processing modules
- Performance optimization requiring algorithm optimization for large data volumes
- Modular and scalable architecture integrating **70+ functional modules**
- Automation and reliability of measurement analysis, direct link between quality control and manufacturing

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

Electronic Card Test Bench Software (October 2017-April 2018)

Context: SuperSonic Imagine - Software Development **Achievements**:

- Development of electronic test software on **Debian**
- **Client/server** architecture for remote control and data collection
- Software instrumentation for control and communication with measurement equipment
- Calculation optimization with **Numpy** for large datasets
- Implementation of **SPC** (Statistical Process Control) analysis method
- New test scenarios to extend electronic card validation coverage

Technologies: Python, Debian, Numpy, Serial/TCP Communication

---

Open source contributions

Active contributor on various open source projects available on GitHub:

- Development and maintenance of Python libraries
- Contribution to third-party projects
- Creation of tools for the community

https://github.com/jgoutin

Languages

- **French**: Native
- **English**: Professional in writing, intermediate in speaking
- **German**: Basic notions
