JEAN-MARIE EGA FOSSO

Architecte cloud | Architecte Cloud & DevOps | Ingénieur de plateforme | Ingénieur IA | Responsable technique

Québec, Canada | +1 514 246 1417 | egafosso@gmail.com | linkedin.com/in/ega-fosso | github.com/egafossojm

Profil professionnel

Professionnel senior Cloud, DevOps, plateforme et IA avec environ 10 ans d'expérience TI et plus de 5 ans consacrés à l'architecture infonuagique, AWS, l'automatisation d'infrastructure, DevOps, l'ingénierie de plateforme, les architectures serverless, l'orchestration de conteneurs (Docker, ECS, Kubernetes) et les charges conteneurisées.

Expérience pratique de conception, mise en œuvre et exploitation de plateformes infonuagiques sécurisées, hautement disponibles et évolutives en développement, staging et production. Solide bagage en architecture AWS, Infrastructure as Code, CI/CD, conteneurisation, réseautage, sécurité, observabilité, reprise après sinistre et optimisation des coûts.

Historique éprouvé de pilotage d'initiatives infonuagiques greenfield, de la découverte technique et l'architecture jusqu'à l'IaC, le déploiement applicatif, le durcissement sécurité, l'observabilité, la mise en production et l'amélioration continue.

Expérience pratique des architectures d'IA générative, Amazon Bedrock, Amazon Bedrock AgentCore, pipelines RAG, modèles de fondation, ingénierie de prompts, évaluation de modèles, garde-fous IA, flux d'IA agentique et applications propulsées par l'IA.

Solide expérience de leadership technique : revues d'architecture, décisions techniques, standards d'ingénierie, revues de code et collaboration interfonctionnelle

Proposition de valeur

- Architecture cloud AWS
- Architecture cloud-native
- Architecture de solutions
- Migration infonuagique d'entreprise
- Modernisation d'applications legacy
- Architecture serverless
- Applications conteneurisées
- Amazon ECS / AWS Fargate
- Ingénierie de plateforme Kubernetes
- Amazon EKS
- Sécurité des conteneurs et GitOps
- Infrastructure as Code
- Terraform
- AWS CDK
- AWS CloudFormation
- DevOps
- DevSecOps
- CI/CD
- Sécurité infonuagique
- Architecture identité et sécurité
- Zero Trust

- Garde-fous de type CSPM
- AWS Networking
- Infonuagique hybride et connectivité
- Landing zones multi-comptes
- Haute disponibilité
- Reprise après sinistre
- Continuité des activités
- Observabilité
- Alertes orientées SLO
- Analyse de cause racine (RCA)
- Optimisation des coûts infonuagiques / FinOps
- IA générative
- Amazon Bedrock
- Amazon Bedrock AgentCore
- RAG
- Embeddings
- AI Agents
- IA agentique
- Modèles de fondation
- Ingénierie de prompts
- Garde-fous IA
- Leadership technique
- Gouvernance d'architecture
- Standards d'ingénierie
- Automatisation
- Python
- TypeScript
- Bash
- SQL

Compétences techniques

Plateformes infonuagiques

Amazon Web Services

- AWS Organizations
- AWS Control Tower
- AWS IAM
- AWS IAM Identity Center
- AWS VPC
- AWS Transit Gateway
- AWS PrivateLink
- AWS VPN
- AWS Route 53
- AWS CloudFront
- AWS WAF
- AWS ALB
- AWS NLB
- AWS API Gateway
- AWS Lambda
- AWS Step Functions

- AWS EventBridge
- Amazon SQS
- Amazon SNS
- Amazon Kinesis
- Amazon ECS
- AWS Fargate
- Amazon EKS
- Amazon EC2
- Auto Scaling
- Amazon ECR
- Amazon RDS
- Amazon Aurora
- Amazon DynamoDB
- Amazon ElastiCache
- Amazon S3
- Amazon EFS
- AWS Backup
- AWS Elastic Reprise après sinistre
- AWS Systems Manager
- AWS CloudWatch
- AWS CloudTrail
- AWS Config
- AWS Security Hub
- Amazon GuardDuty
- Amazon Inspector
- AWS KMS
- AWS Secrets Manager
- AWS Certificate Manager
- AWS Key Management Service
- AWS Cost Explorer
- AWS Budgets
- AWS Savings Plans
- AWS Trusted Advisor
- Amazon Bedrock
- Amazon Bedrock AgentCore
- Amazon SageMaker / SageMaker AI
- AWS Glue
- Amazon Athena
- Amazon Kinesis Data Firehose

Architecture infonuagique

- AWS Well-Architected Framework
- Cloud-native architecture
- Serverless architecture
- Event-driven architecture
- Microservices architecture
- Distributed systems
- Multi-tier architectures
- Multi-account AWS architecture
- Multi-environment architecture
- Multi-AZ architecture

- High-availability architecture
- Fault-tolerant architecture
- Resilient architecture
- Scalable architecture
- Stateless application architecture
- API-driven architecture
- Asynchronous architectures
- Decoupled architectures
- SaaS architectures
- Multi-tenant SaaS architecture
- Secure-by-design architecture
- Zero-trust principles
- Disaster recovery architecture
- Business continuity architecture
- Hybrid cloud architecture
- Cloud migration architecture
- Application modernization
- Legacy modernization
- Cloud-native transformation
- Landing zones
- Cloud governance
- Architecture standards
- Architecture decision records
- Technical reference architectures
- Plateformes applicatives privées
- Garde-fous de type CSPM
- Patrons d'ingénierie de plateforme
- Landing zones multi-comptes

Réseautage AWS

- VPC design
- Public/private subnet architecture
- Route tables
- Internet Gateway
- NAT Gateway
- Network ACLs
- Security Groups
- VPC endpoints
- Interface endpoints
- Gateway endpoints
- Transit Gateway
- VPC peering
- DNS architecture
- Route 53
- Private hosted zones
- Public hosted zones
- Route 53 Resolver
- Application Load Balancer
- Network Load Balancer
- CloudFront

- AWS WAF
- TLS/SSL
- ACM
- Network segmentation
- Private service communication
- Cross-account networking
- Hybrid connectivity
- VPN
- Site-to-Site VPN
- Network troubleshooting
- DNS troubleshooting
- Traffic routing
- High-availability network design

Calcul et plateformes applicatives

- Amazon EC2
- Auto Scaling Groups
- Elastic Load Balancing
- ECS
- ECS Fargate
- Docker
- Amazon ECR
- Lambda
- API Gateway
- EventBridge
- Step Functions
- SQS
- SNS
- Microservices
- REST APIs
- Event-driven applications
- Container orchestration
- Ingénierie de plateforme Kubernetes
- Amazon EKS
- Application deployment automation
- Blue/green deployments
- Rolling deployments
- Canary deployments
- Autoscaling
- Application health checks
- Limitation de débit API
- Patrons d'authentification/autorisation

Serveurs Web et proxy inverse

- Nginx
- Apache HTTP Server
- Traefik
- Proxy inverse
- Routage upstream
- Terminaison TLS
- Hôtes virtuels / configuration de services

- Routage HTTP (entrypoints, routers, services)
- Routage de type Ingress
- Diagnostics et dépannage 502 / upstream

Bases de données et stockage

- Amazon RDS
- Amazon Aurora
- MySQL
- PostgreSQL
- DynamoDB
- Redis / ElastiCache
- S3
- EFS
- Database backup
- Database restore
- Database migration
- Database high availability
- Read replicas
- Multi-AZ databases
- Database performance optimization
- Database connection management
- Database encryption
- Data lifecycle management
- S3 lifecycle policies
- Object versioning
- Cross-region replication
- Backup and retention strategies

Infrastructure as Code

- Terraform
- Terraform modules
- Terraform state management
- Terraform remote state
- Terraform workspaces
- Terraform variables
- Terraform outputs
- Terraform providers
- Infrastructure modules
- Reusable infrastructure components
- AWS CDK
- AWS CDK with Python
- AWS CDK with TypeScript
- CloudFormation
- CloudFormation stacks
- StackSets
- Infrastructure drift detection
- Infrastructure testing
- Automatisé provisioning
- Environment standardization
- Infrastructure versioning
- Infrastructure code reviews

DevOps et CI/CD

- GitHub Actions
- AWS CodePipeline
- AWS CodeBuild
- AWS CodeDeploy
- CI/CD pipeline design
- Automatisé builds
- Automatisé testing
- Automatisé deployment
- Artifact management
- Container image pipelines
- Infrastructure deployment pipelines
- Environment promotion
- Release automation
- Deployment approvals
- Rollback strategies
- Blue/green deployment
- Canary deployment
- Git-based workflows
- GitOps concepts
- GitOps
- DevSecOps
- Pipeline security
- SAST
- Secrets management
- Dependency scanning
- Analyse de secrets
- Container scanning
- Infrastructure security scanning
- Analyse conteneurs/IaC
- Vulnerability management
- Portes de vulnérabilité

Conteneurisation

- Docker
- Dockerfiles
- Docker Compose
- Multi-stage Docker builds
- Container image optimization
- Container security
- Analyse de sécurité des conteneurs
- Amazon ECR
- ECS
- ECS Fargate
- Définitions de tâches ECS
- Profils Fargate
- Découverte de services
- Secrets
- Injection de secrets
- Health checks

- Resource limits
- Limites et requêtes de ressources
- Autoscaling
- Container observability
- Container networking
- Container troubleshooting
- Déploiements blue/green et canary

Kubernetes et orchestration de conteneurs

- Architecture de cluster (plan de contrôle, nœuds, ordonnancement)
- Pods
- Deployments
- StatefulSets
- DaemonSets
- Jobs / CronJobs
- ReplicaSets
- Services (ClusterIP, NodePort, LoadBalancer)
- Ingress
- Contrôleurs Ingress
- Namespaces
- ConfigMaps
- Secrets
- PersistentVolumes
- PersistentVolumeClaims
- StorageClasses
- RBAC
- NetworkPolicy
- Limites et requêtes de ressources
- Probes liveness / readiness / startup
- Mises à jour rolling et rollbacks
- Autoscaling horizontal de pods (HPA)
- Dépannage kubectl (get, describe, logs, events)
- Helm
- GitOps (concepts Argo CD)
- Correspondance de charges ECS → Kubernetes
- Container networking
- Patrons de clusters multi-environnements
- Amazon EKS
- Patrons d'ingénierie de plateforme pour Kubernetes

Automatisation DevOps

- Bash
- Python
- Ansible
- Shell scripting
- Infrastructure automation
- Configuration management
- Environment automation
- Deployment automation
- Operational automation
- Automatisé remediation

- Scheduled automation
- AWS Systems Manager automation
- Lambda automation
- Event-driven automation

Sécurité

- AWS IAM
- Least-privilege access
- IAM policies
- IAM roles
- Service roles
- Cross-account IAM
- IAM Identity Center
- AWS Organizations
- Security groups
- Network segmentation
- AWS WAF
- AWS Shield
- AWS KMS
- Secrets Manager
- Parameter Store
- CloudTrail
- AWS Config
- Security Hub
- GuardDuty
- Inspector
- Vulnerability management
- Security monitoring
- Encryption at rest
- Encryption in transit
- TLS
- Certificate management
- Secret rotation
- Security baselines
- Security guardrails
- DevSecOps
- Infrastructure security
- Container security
- Application security
- Secure CI/CD
- Zero-trust principles
- Garde-fous de type CSPM
- Security architecture reviews
- Standards d'étiquetage
- Automatisé remediation

Observabilité et opérations

- Amazon CloudWatch
- CloudWatch Logs
- CloudWatch Metrics
- CloudWatch Alarms

- CloudWatch Dashboards
- CloudWatch Events / EventBridge
- AWS CloudTrail
- AWS Config
- Distributed application monitoring
- Application performance monitoring
- Concepts APM
- Infrastructure monitoring
- Log aggregation
- Centralisé logging
- Alerting
- Alertes orientées SLO
- Incident response
- Root-cause analysis
- Performance troubleshooting
- Capacity planning
- Availability monitoring
- SLI/SLO concepts
- Operational dashboards
- Automatisé remediation
- Revues de préparation à la production

Reprise après sinistre

- Backup strategies
- AWS Backup
- Database backups
- Point-in-time recovery
- Cross-region backups
- Disaster recovery planning
- RPO/RTO definition
- Recovery procedures
- Disaster recovery testing
- Failover strategies
- Business continuity
- High availability
- Multi-AZ architecture
- Multi-region architecture
- Recovery automation

FinOps / Optimisation des coûts infonuagiques

- AWS Cost Explorer
- AWS Budgets
- AWS Trusted Advisor
- Rightsizing
- Resource utilization analysis
- Instance optimization
- Savings Plans
- Reserved capacity analysis
- Storage optimization
- S3 lifecycle policies
- Log retention optimization

- Idle resource detection
- Cost allocation
- Tagging strategies
- Environment cost management
- Architecture-level cost optimization
- Serverless cost analysis
- Container cost optimization

Programmation et scripts

- Python
- TypeScript
- Bash
- Shell scripting
- SQL
- YAML
- JSON
- Infrastructure configuration
- REST API development
- Automatisation scripts
- AWS SDK / Boto3
- Lambda development
- CDK development

IA générative et ingénierie IA

- IA générative
- Large Language Models
- Modèles de fondation
- Amazon Bedrock
- Amazon Bedrock AgentCore
- Amazon SageMaker AI
- RAG
- Retrieval-Augmented Generation
- Ingénierie de prompts
- Prompt Templates
- Model Evaluation
- Garde-fous IA
- LLM application architecture
- AI-powered APIs
- AI agents
- IA agentique
- Appel d'outils
- Conversational AI
- AI assistants
- Multimodal AI
- Embeddings
- Vector search
- Knowledge bases
- Document ingestion
- Document processing
- Ingestion et découpage de documents
- Pipelines de récupération

- Prompting contextuel
- Grounding
- Semantic search
- Context retrieval
- LLM orchestration
- AI workflow automation
- Model selection
- Model routing
- Token/cost optimization
- AI observability
- AI security
- Responsible AI
- AI application deployment
- AI integration with serverless architectures
- Intégration Bedrock + Lambda/API Gateway
- AgentCore Runtime / patrons de déploiement d'agents
- Intégration et gouvernance des outils d'agents
- Récupération IA tenant-aware

Architecture IA / RAG

Domaines d'expérience validés potentiels :

- Conçu RAG architectures using managed AWS services
- Construit document ingestion pipelines
- Implémenté document chunking strategies
- Implémenté embeddings generation
- Intégré vector search
- Conçu retrieval pipelines
- Implémenté contextual prompting
- Implémenté grounding mechanisms
- Conçu evaluation workflows
- Implémenté hallucination mitigation strategies
- Implémenté access controls for enterprise knowledge
- Conçu tenant-aware AI retrieval
- Intégré LLM applications with existing APIs
- Construit AI-powered backend workflows
- Implémenté AI guardrails
- Monitored model usage and costs
- Optimisé prompt and retrieval performance

Leadership technique

- Technical leadership
- Architecture ownership
- Technical discovery
- Requirements analysis
- Architecture workshops
- Technical design
- Architecture reviews
- Design reviews
- Code reviews
- Engineering standards

- Coding standards
- Cloud governance
- Technical documentation
- Architecture Decision Records
- Technical mentoring
- Engineer onboarding
- Knowledge sharing
- Technical presentations
- Cross-functional collaboration
- Stakeholder communication
- Technical risk management
- Technical debt management
- Engineering best practices
- Revues de préparation à la production

Expérience professionnelle

Architecte senior Cloud & DevOps

Toumoro Group Inc. — Québec, Canada

août 2023 – juin 2026

Architecture de base et leadership technique

- Piloté de bout en bout architecture for greenfield cloud-native products from technical discovery through production deployment.
- Conçu scalable AWS architectures aligned with availability, security, performance, maintainability and cost requirements.
- Défini technical architecture standards and reusable engineering patterns across cloud projects.
- Traduit business and technical requirements into cloud architecture and implementation plans.
- Animé architecture reviews and evaluated technical trade-offs across competing implementation approaches.
- Établi engineering standards covering infrastructure, application deployment, security and operational practices.
- Fourni hands-on technical leadership to engineers throughout implementation and production rollout.
- Effectué code and infrastructure reviews to enforce quality, security and maintainability standards.
- Mentoré engineers on AWS architecture, Infrastructure as Code, DevOps, cloud security and AI engineering.

Architecture AWS

- Architecturé highly available multi-AZ AWS environments supporting mission-critical workloads.
- Conçu serverless architectures using Lambda, API Gateway, EventBridge, DynamoDB and S3.
- Conçu containerized workloads using Docker, ECS and Fargate.
- Conçu event-driven application architectures to decouple application components and improve scalability.
- Implémenté scalable backend architectures capable of handling variable workloads without manual infrastructure provisioning.
- Conçu secure application architectures using private networking, IAM, encryption, WAF and centralized security controls.
- Conçu application architectures around managed AWS services to reduce infrastructure operational overhead.
- Évalué architecture trade-offs between serverless, containers and traditional compute.
- Conçu resilient architectures with appropriate failure isolation and recovery mechanisms.

Infrastructure as Code

- Implémenté infrastructure using AWS CDK with Python and TypeScript.
- Implémenté infrastructure using Terraform.
- Développé reusable Infrastructure as Code components to standardize AWS deployments.
- Automatisé environment provisioning and configuration.
- Établi version-controlled infrastructure workflows.
- Intégré infrastructure deployment into CI/CD pipelines.
- Réduit le délai de provisionnement d'environnements d'environ cinq jours à environ une heure grâce à l'automatisation.
- Implémenté infrastructure review practices to detect configuration and security issues before deployment.

DevOps et CI/CD

- Conçu and implemented CI/CD pipelines using GitHub Actions and Ansible.
- Automatisé application build, validation and deployment workflows.
- Automatisé infrastructure deployment through CI/CD.
- Implémenté environment promotion workflows across development, staging and production.
- Amélioré deployment consistency and reduced deployment errors.
- Établi repeatable release processes for cloud-native applications.
- Intégré automated checks into delivery pipelines.
- Amélioré engineering feedback loops through automated deployment workflows.

Sécurité

- Implémenté AWS security guardrails across IAM, WAF, CloudTrail and Security Hub.
- Applied least-privilege access principles to cloud workloads.
- Conçu secure IAM roles and policies for applications and infrastructure.
- Implémenté encryption and secret-management mechanisms.
- Intégré security controls into cloud infrastructure and deployment workflows.
- Amélioré security visibility through centralized AWS security services.
- Appuyé security reviews of cloud architectures.
- Intégré security considerations into Infrastructure as Code.

Observabilité et opérations

- Conçu CloudWatch monitoring and alerting for production workloads.
- Implémenté operational dashboards and application/infrastructure monitoring.
- Centralisé operational logging.
- Défini monitoring requirements for critical cloud services.
- Appuyé production troubleshooting and root-cause analysis.
- Implémenté automated operational responses where appropriate.
- Établi production-readiness practices covering monitoring, logging, alerting and recovery.

Performance et fiabilité

- Architecturé des environnements visant une disponibilité supérieure à 99,95 %.
- Conçu architectures using multi-AZ deployment patterns.
- Implémenté resilient application components and failure-isolation strategies.
- Évalué performance bottlenecks across application and infrastructure layers.
- Implémenté scalable architectures capable of responding to workload changes.
- Amélioré reliability through automation and standardized infrastructure.

Réalisations supplémentaires

- Conçu multi-account AWS environments.
- Implémenté centralized AWS logging and security monitoring.
- Implémenté cross-account IAM access patterns.

- Conçu private AWS service connectivity using VPC endpoints.
- Implémenté automated compliance checks.
- Implémenté infrastructure security scanning.
- Conçu disaster recovery procedures.
- Automatisé backup and recovery workflows.
- Implémenté cloud cost dashboards.
- Implémenté automated detection of unused AWS resources.
- Établi tagging and cost-allocation standards.
- Conçu reusable internal AWS architecture patterns.
- Établi standardized project bootstrapping templates.
- Créé reusable CDK constructs and Terraform modules.
- Construit internal developer platform capabilities.
- Implémenté automated environment teardown for temporary environments.
- Conçu ephemeral development environments.
- Implémenté automated infrastructure testing.

Ingénieur senior Cloud & DevOps — Contractuel

ITS Group — Lyon, France

janvier 2023 – avril 2023

Architecture infonuagique et modernisation

- Dirigé les activités techniques architecture activities for AWS cloud modernization initiatives.
- Évalué existing application architectures and identified modernization opportunities.
- Conçu des architectures pour applications à grande échelle (200 000+ utilisateurs actifs mensuels).
- Évalué workload requirements across availability, scalability, performance, security and operational complexity.
- Développé modernization approaches for legacy workloads.
- Appuyé migration of application components to AWS managed services.

Conteneurisation

- Conteneurisé legacy application workloads using Docker.
- Conçu Docker-based deployment patterns for consistent application execution.
- Déployé containerized workloads using Amazon ECS.
- Standardisé application environments across development, staging and production.
- Réduit environment drift through containerized deployment.
- Optimisé container configurations for application reliability and operational consistency.

CI/CD

Responsabilités validées potentielles :

- Conçu CI/CD workflows for containerized workloads.
- Automatisé Docker image builds.
- Automatisé image publishing to Amazon ECR.
- Automatisé ECS deployments.
- Implémenté environment-specific configuration management.
- Automatisé release and rollback procedures.
- Intégré quality and security checks into deployment pipelines.

Observabilité

- Conçu centralized monitoring using Amazon CloudWatch.
- Implémenté application and infrastructure monitoring.
- Créé operational dashboards.
- Configuré alarms for critical services.
- Centralisé application logs.

- Appuyé production incident investigation.
- Effectué root-cause analysis of infrastructure and application failures.

Sauvegarde et reprise après sinistre

- Conçu automated backup strategies.
- Implémenté recovery procedures for critical workloads.
- Conçu disaster recovery architecture around defined RPO/RTO requirements.
- Atteint des objectifs de reprise : RPO inférieur à 15 minutes et RTO inférieur à une heure.
- Évalué recovery dependencies across application, compute, storage and database layers.
- Appuyé disaster recovery testing and validation.

Réalisations supplémentaires

- Implémenté ECS service autoscaling.
- Implémenté Application Load Balancer integration.
- Intégré ECS services with CloudWatch.
- Implémenté centralized container logging.
- Implémenté ECR image lifecycle policies.
- Implémenté vulnerability scanning for container images.
- Implémenté blue/green ECS deployments.
- Conçu zero-downtime deployment strategies.
- Implémenté AWS Backup policies.
- Conçu cross-region disaster recovery.
- Automatisé infrastructure provisioning using Terraform.
- Automatisé application deployments using GitHub Actions.
- Implémenté IAM least-privilege roles for ECS workloads.

Architecte cloud & responsable technique

RenovaCloud — Ho Chi Minh City, Vietnam

octobre 2020 – septembre 2022

Architecture infonuagique

- Architectured design for AWS-based multi-tenant SaaS platforms.
- Conçu secure, scalable and highly available AWS environments.
- Conçu network architectures using VPC, subnets, routing, security groups and load balancing.
- Conçu application architectures using ALB, RDS Aurora, CloudFront and AWS WAF.
- Conçu architectures supporting multiple application tenants.
- Évalué scalability and availability requirements for SaaS workloads.
- Défini technical architecture patterns for cloud-native application delivery.

Architecture serverless

- Conçu high-performance REST APIs using AWS Lambda and API Gateway.
- Implémenté serverless application components capable of scaling automatically with demand.
- Conçu API authentication and authorization mechanisms.
- Implémenté event-driven backend workflows.
- Intégré serverless applications with AWS storage and database services.
- Optimisé serverless workloads for performance and cost.

Réseautage

- Conçu AWS VPC architectures.
- Conçu public and private subnet segmentation.
- Configuré routing and security controls.
- Intégré Application Load Balancers with application workloads.
- Configuré CloudFront distributions for application delivery.

- Conçu DNS and traffic-routing architectures.
- Implémenté WAF protection for internet-facing applications.

DevSecOps

- Piloté DevSecOps implementation across application delivery workflows.
- Intégré automated vulnerability scanning into CI/CD pipelines.
- Intégré security validation into software delivery processes.
- Réduit exposure to critical application security risks.
- Implémenté secure deployment practices.
- Promu security ownership across development and operations workflows.

Optimisation des coûts

- Piloté AWS cost optimization initiatives.
- Effectué resource rightsizing and utilization analysis.
- Évalué Savings Plans and other AWS pricing mechanisms.
- Réduit les dépenses infonuagiques annuelles d'environ 25 %.
- Identifié underutilized resources.
- Optimisé cloud resource allocation based on workload requirements.
- Amélioré cost visibility across cloud environments.

Bases de données

Responsabilités validées potentielles :

- Conçu Amazon RDS and Aurora architectures.
- Conçu database high-availability configurations.
- Implémenté database backup and recovery strategies.
- Optimisé database resource sizing.
- Analysé database performance.
- Implémenté database monitoring.
- Conçu secure database connectivity from application workloads.
- Implémenté encryption for database workloads.

CDN et périphérie

- Conçu CloudFront architectures for application delivery.
- Configuré origin integrations.
- Implémenté caching strategies.
- Intégré AWS WAF with CloudFront.
- Amélioré application delivery performance through edge caching.
- Protégé internet-facing workloads using edge security controls.

Réalisations supplémentaires

- Conçu multi-account AWS environments.
- Implémenté Terraform-based infrastructure provisioning.
- Construit reusable Terraform modules.
- Implémenté automated environment provisioning.
- Implémenté GitHub Actions or AWS CodePipeline deployments.
- Implémenté ECS/Fargate workloads.
- Implémenté Docker-based application delivery.
- Conçu centralized logging.
- Implémenté CloudWatch monitoring.
- Implémenté AWS CloudTrail.
- Implémenté AWS Config.
- Implémenté GuardDuty.
- Implémenté Security Hub.

- Conçu disaster recovery architectures.
- Implémenté automated backups.
- Implémenté cross-region replication.
- Implémenté blue/green deployments.
- Implémenté canary releases.
- Conçu zero-downtime deployment architectures.

Expérience TI antérieure

Le CV actuel indique environ 10 ans d'expérience TI mais ne détaille que trois postes professionnels. Ne pas insérer ce qui suit dans l'historique d'emploi tant que les employeurs, postes et dates correspondants n'ont pas été retrouvés dans votre parcours réel.

Domaines de compétence de postes antérieurs à investiguer :

- Systems administration
- Linux administration
- Windows Server administration
- Network administration
- Cisco networking
- Routing and switching
- Firewall administration
- DNS
- DHCP
- Active Directory
- Virtualization
- VMware
- Hyper-V
- Red Hat Enterprise Linux
- Ansible automation
- Bash automation
- Enterprise infrastructure
- Monitoring
- Backup administration
- Security administration
- Infrastructure troubleshooting
- Production support
- Technical documentation

Réalisations professionnelles sélectionnables

Utiliser uniquement les réalisations que vous pouvez étayer.

Architecture

- Conçu cloud architectures supporting highly available production workloads.
- Réduit infrastructure provisioning time from days to hours through Infrastructure as Code.
- Conçu reusable cloud architecture patterns adopted across multiple projects.
- Modernized legacy workloads into containerized cloud architectures.
- Conçu event-driven architectures that reduced coupling between application components.
- Conçu serverless architectures that eliminated infrastructure management overhead.
- Conçu multi-tenant SaaS architectures.
- Conçu resilient multi-AZ architectures.
- Conçu secure cloud architectures using defense-in-depth security controls.

Automatisation

- Automatisé infrastructure provisioning using Terraform and AWS CDK.
- Automatisé application deployment through CI/CD.
- Automatisé environment configuration using Ansible.
- Automatisé operational workflows using Python and Bash.
- Réduit manual infrastructure operations through reusable automation.

DevOps

- Construit CI/CD pipelines covering source control, build, testing, security validation and deployment.
- Standardisé application deployment across development, staging and production.
- Réduit deployment errors through automated delivery workflows.
- Introduit infrastructure automation into engineering workflows.
- Amélioré deployment repeatability through Infrastructure as Code.

Sécurité

- Implémenté least-privilege IAM architectures.
- Intégré security controls into CI/CD pipelines.
- Implémenté WAF protection for internet-facing applications.
- Centralisé AWS audit logging using CloudTrail.
- Implémenté cloud security monitoring.
- Intégré vulnerability scanning into software delivery.

Fiabilité

- Conçu multi-AZ production architectures.
- Implémenté automated backup and recovery.
- Conçu disaster recovery solutions targeting defined RPO/RTO objectives.
- Amélioré application availability through resilient architecture patterns.
- Implémenté monitoring and alerting for production systems.

Coûts

- Réduit les dépenses AWS annuelles d'environ 25 %.
- Implémenté resource rightsizing.
- Évalué Savings Plans.
- Optimisé serverless and container workloads.
- Amélioré cloud cost visibility.

Leadership

- Établi engineering standards for cloud development.
- Mentoré engineers on AWS and DevOps.
- Animé architecture reviews.
- Animé code reviews.
- Dirigé les activités techniques discovery sessions.
- Traduit business requirements into technical architecture.
- Guidé engineering teams through cloud modernization.
- Agi comme hands-on technical lead from architecture through production.

Banque de projets maîtres

1. Landing zone AWS d'entreprise

Design and implement a multi-account AWS foundation using AWS Organizations, centralized security, IAM, networking, logging, monitoring, governance and account provisioning.

Technologies potentielles :

AWS Organizations, Control Tower, IAM Identity Center, CloudTrail, Config, Security Hub, GuardDuty, Terraform, AWS CDK, VPC, Transit Gateway.

2. Plateforme de sécurité AWS multi-comptes

Build centralized security controls across multiple AWS accounts including logging, threat detection, vulnerability management, security findings aggregation and automated remediation.

3. Plateforme SaaS cloud-native

Design and implement a multi-tenant SaaS platform using:

- API Gateway
- Lambda
- DynamoDB
- S3
- CloudFront
- Route 53
- WAF
- IAM
- CloudWatch
- EventBridge

Accent sur l'isolation des locataires, l'évolutivité, l'observabilité et la sécurité.

4. Plateforme API REST serverless

Build a production-grade serverless API using API Gateway, Lambda, DynamoDB/RDS, IAM, CloudWatch and CI/CD.

Inclure :

- Authentication
- Authorization
- Limitation de débit API
- Monitoring
- Logging
- Error handling
- Automatisé deployment
- Infrastructure as Code

5. Plateforme de microservices événementielle

Design an event-driven architecture using:

- EventBridge
- SQS
- SNS
- Lambda
- ECS
- DynamoDB
- S3

Implement asynchronous processing, retries, dead-letter queues and observability.

6. Plateforme microservices ECS Fargate

Containerize multiple services using Docker and deploy them to ECS Fargate behind an Application Load Balancer.

Inclure :

- ECR
- IAM
- Secrets Manager
- CloudWatch

- Autoscaling
- Health checks
- CI/CD
- Blue/green deployments
- Politiques de cycle de vie des images
- Analyse de vulnérabilités
- Observabilité CloudWatch

7. Ingénierie de plateforme Kubernetes (architecture prête pour EKS)

Construit des patrons de plateforme Kubernetes réutilisables : topologie de cluster, standards de charge (Deployments/Services/Ingress), ConfigMaps/Secrets, stockage, RBAC, probes, autoscaling, livraison GitOps et runbooks opérationnels — alignés avec des modules Terraform pour VPC, IAM et provisionnement EKS.

Inclure :

- Architecture de cluster (plan de contrôle, nœuds, ordonnancement)
- Deployments, Services, Ingress
- ConfigMaps / Secrets
- PersistentVolumes / PersistentVolumeClaims / StorageClasses
- RBAC and NetworkPolicy
- Probes liveness / readiness / startup
- Horizontal Pod Autoscaling
- Helm
- GitOps (concepts Argo CD)
- Terraform modules for VPC, IAM, EKS
- Patrons de clusters multi-environnements
- kubectl operational troubleshooting

9. Plateforme Infrastructure-as-Code

Create a reusable Terraform framework capable of provisioning standardized AWS environments.

Modules potentiels :

- VPC
- IAM
- ECS
- EKS
- RDS
- S3
- CloudFront
- Route 53
- WAF
- CloudWatch
- Security Hub

10. Cadre d'architecture AWS CDK

Develop reusable AWS CDK constructs using Python or TypeScript for standardized application infrastructure.

Composants potentiels :

- VPC
- Lambda
- API Gateway
- DynamoDB
- S3
- EventBridge
- ECS

- CloudFront
- WAF
- IAM

11. Plateforme CI/CD automatisée

Build a reusable CI/CD platform using GitHub Actions.

Étapes du pipeline :

1. Source

2. Dependency installation

3. Unit testing

4. Static analysis

5. Security scanning

6. Docker build

7. Image scanning

8. ECR publishing

9. Infrastructure deployment

10. Déploiement applicatif 11. Tests de fumée 12. Rollback

12. Plateforme DevSecOps

Integrate security into the entire software delivery lifecycle.

Capacités potentielles :

- SAST
- dependency scanning
- secret scanning
- container scanning
- IaC scanning
- vulnerability management
- security gates
- artifact validation
- AWS security controls

13. Plateforme de reprise après sinistre infonuagique

Design a production DR architecture with explicit RPO/RTO objectives.

Services potentiels :

- AWS Backup
- RDS/Aurora
- S3
- Cross-region replication
- Route 53
- CloudFormation/Terraform
- Lambda
- EventBridge

14. Architecture AWS multi-régions

Design an architecture capable of surviving a regional outage.

Composants potentiels :

- Route 53

- CloudFront
- S3
- Aurora
- DynamoDB
- Lambda
- ECS
- Terraform/CDK

15. Plateforme d'optimisation des coûts AWS

Build an automated system that detects:

- idle EC2 instances
- unattached EBS volumes
- unused Elastic IPs
- oversized databases
- inefficient storage
- excessive log retention
- underutilized resources

Generate recommendations and cost reports.

16. Plateforme d'observabilité infonuagique

Build centralized observability around:

- CloudWatch
- CloudTrail
- application logs
- infrastructure metrics
- alarms
- dashboards
- operational events
- automated incident notifications

17. Assistant de connaissances RAG propulsé par l'IA (Amazon Bedrock)

Construire un assistant de connaissances d'entreprise capable de répondre à des questions sur des documents internes. Livrer une architecture RAG d'entreprise avec ingestion, découpage, embeddings, récupération vectorielle, prompting contextuel, grounding, garde-fous et intégration API sécurisée via Lambda/API Gateway avec surveillance et contrôles d'accès.

L'architecture peut inclure :

- Amazon Bedrock
- Amazon Bedrock AgentCore
- S3
- embeddings
- vector database/search
- document ingestion
- chunking
- retrieval
- prompt orchestration
- guardrails
- API Gateway
- Lambda
- authentication
- monitoring

18. Assistant de support client IA

Build a conversational AI assistant capable of answering customer questions using enterprise knowledge.

Capacités potentielles :

- conversational interface
- RAG
- Bedrock
- knowledge base
- guardrails
- conversation history
- escalation
- API integration
- monitoring
- analytics

19. Pipeline de traitement de documents IA

Build an automated document-processing platform.

Flux de travail potentiel :

Document upload → extraction → classification → enrichment → embedding → indexing → RAG → AI analysis → structured output.

Services AWS potentiels :

S3, Lambda, EventBridge, Step Functions, Bedrock, DynamoDB.

20. Plateforme d'agents IA / IA agentique et appels d'outils (Amazon Bedrock AgentCore)

Construire et exploiter des agents IA capables d'interagir avec des API et services infonuagiques via des outils contrôlés. Concevoir des systèmes d'IA agentique avec Amazon Bedrock et Amazon Bedrock AgentCore : appels d'outils, récupération, mémoire, autorisation, intégrations API, garde-fous, journalisation d'audit et observabilité pour une automatisation contrôlée.

Capacités potentielles :

- Amazon Bedrock AgentCore
- AgentCore Runtime / patrons de déploiement d'agents
- appel d'outils (tool calling)
- planification
- récupération
- mémoire
- garde-fous
- autorisation
- intégrations API
- observabilité
- journalisation d'audit
- flux d'IA agentique
- automatisation contrôlée sur systèmes infonuagiques et applicatifs

21. Assistant IA de dépannage infrastructure infonuagique et CI/CD

Assistant — concevoir un assistant IA capable d'analyser :

- CloudWatch logs and metrics
- CI/CD deployment pipeline failures (GitHub Actions / CodePipeline)
- Terraform and AWS CDK execution errors
- Application stack traces and serverless runtime exceptions (Lambda / ECS Fargate)
- Security Hub findings and IAM misconfigurations
- AWS resource state drifts and generating automated remediation scripts/recommendations.

22. Assistant IA d'optimisation des coûts infonuagiques

Build an AI system that analyzes AWS cost data and generates:

- anomaly explanations
- optimization recommendations
- rightsizing suggestions
- resource waste detection
- monthly cost summaries

23. Remédiation sécurité AWS automatisée

Build an event-driven security remediation system.

Exemple :

Security finding → EventBridge → Lambda → remediation → CloudTrail → notification.

Potential remediations:

- revoke exposed credentials
- isolate compromised resources
- enforce security group rules
- disable public S3 access
- quarantine workloads

24. Plateforme développeur en libre-service

Build an internal platform allowing developers to provision standardized application environments without manually configuring AWS infrastructure.

Pile technologique potentielle :

Terraform/CDK + GitHub Actions + AWS + reusable templates.

25. Plateforme d'environnements de développement éphémères

Create automated environments that developers can create and destroy on demand.

Capacités potentielles :

- one-click environment creation
- Infrastructure as Code
- automatic DNS
- temporary credentials
- automated teardown
- cost controls
- environment expiration

26. Plateforme de déploiement sans interruption

Implement deployment strategies including:

- blue/green
- rolling deployment
- canary
- automated health checks
- rollback
- traffic shifting

27. Architecture SaaS multi-locataires sécurisée

Design tenant isolation across:

- identity
- API authorization
- database

- storage
- logging
- encryption
- networking

Implement tenant-aware access controls and monitoring.

28. Usine de migration infonuagique

Develop a repeatable framework for migrating applications from traditional infrastructure to AWS.

Phases :

Assessment → dependency mapping → target architecture → migration → validation → cutover → optimization.

29. Modernisation d'applications legacy

Transform a monolithic application into:

- containers
- microservices
- serverless components
- managed databases
- event-driven services

while maintaining business continuity.

30. Plateforme applicative AWS privée

Design a fully private application architecture with no direct public access to application workloads.

Composants potentiels :

- private subnets
- NAT
- VPC endpoints
- internal ALB
- private DNS
- Systems Manager
- IAM
- CloudFront/WAF at the edge

31. Pipeline d'analytique des journaux d'accès (Kinesis → S3 → Athena)

Centralisé application log ingestion and queryable analytics on AWS.

- Déployé and configured Amazon Kinesis Agent on application servers to collect access logs and stream them to Amazon Kinesis Data Firehose.
- Configuré Firehose with an Amazon S3 destination for durable, scalable storage of application access logs.
- Implémenté an AWS Glue Crawler over the S3 log prefix to automatically discover schema and create/update a Glue Data Catalog table.
- Activé interactive querying of application access logs with Amazon Athena for operational analysis and troubleshooting.

Certifications

Certifications déjà inscrites au CV actuel

- AWS Certified Cloud Practitioner

Badges de formation AWS supplémentaires

- AWS Simulearn - Cloud Practitioner - Training Badge
- AWS Simulearn - AI Practitioner - Training Badge

- AWS Simulearn - Solution Architect - Training Badge
- AWS Simulearn - Networking - Training Badge
- AWS Simulearn - Security - Training Badge

Formation

Master’s Degree — Networks, Systems and Security

Claude Bernard University Lyon 1 — France

Master’s Degree — Networks and Communication Systems

Vietnam National University — Vietnam

Langues

- Français — langue maternelle
- Anglais — maîtrise professionnelle