JEAN-MARIE EGA FOSSO

Cloud Architect | Cloud & DevOps Architect | Platform Engineer | AI Engineer | Technical Lead

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

Professional Profile

Senior Cloud, DevOps, Platform, and AI professional with approximately 10 years of IT experience and more than 5 years focused on cloud architecture, AWS, infrastructure automation, DevOps, platform engineering, serverless architectures, container orchestration (Docker, ECS, Kubernetes), and containerized workloads.

Hands-on experience designing, implementing and operating secure, highly available and scalable cloud platforms across development, staging and production environments. Strong background in AWS architecture, Infrastructure as Code, CI/CD, containerization, networking, security, observability, disaster recovery and cloud cost optimization.

Proven track record driving greenfield cloud initiatives from technical discovery and architecture through Infrastructure as Code, application deployment, security hardening, observability, production launch, and continuous improvement.

Hands-on experience with Generative AI architectures, Amazon Bedrock, Amazon Bedrock AgentCore, RAG pipelines, foundation models, prompt engineering, model evaluation, AI guardrails, agentic AI workflows, and AI-enabled application workflows.

Strong technical leadership background including architecture reviews, technical decision-making, engineering standards, code reviews, and cross-functional collaboration

Core Value Proposition

- AWS Cloud Architecture
- Cloud-Native Architecture
- Solutions Architecture
- Enterprise Cloud Migration
- Modernization of Legacy Applications
- Serverless Architecture
- Containerized Applications
- Amazon ECS / AWS Fargate
- Kubernetes platform engineering
- Amazon EKS
- Container security & GitOps
- Infrastructure as Code
- Terraform
- AWS CDK
- AWS CloudFormation
- DevOps
- DevSecOps
- CI/CD
- Cloud Security
- Identity & security architecture
- Zero Trust
- CSPM-style guardrails
- AWS Networking

- Hybrid cloud & connectivity
- Multi-account landing zones
- High Availability
- Disaster Recovery
- Business Continuity
- Observability
- SLO-minded alerting
- Incident RCA
- Cloud Cost Optimization / FinOps
- Generative AI
- Amazon Bedrock
- Amazon Bedrock AgentCore
- RAG
- Embeddings
- AI Agents
- Agentic AI
- Foundation Models
- Prompt Engineering
- AI Guardrails
- Technical Leadership
- Architecture Governance
- Engineering Standards
- Automation
- Python
- TypeScript
- Bash
- SQL

Technical Skills

Cloud Platforms

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 Disaster Recovery
- 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

Cloud Architecture

- 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
- Private application platforms
- CSPM-style guardrails
- Platform engineering patterns
- Multi-account landing zones

AWS Networking

- 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

Compute & Application Platforms

- 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
- Kubernetes platform engineering
- Amazon EKS
- Application deployment automation
- Blue/green deployments
- Rolling deployments
- Canary deployments
- Autoscaling
- Application health checks
- API throttling
- Authentication/authorization patterns

Web Servers & Reverse Proxy

- Nginx
- Apache HTTP Server
- Traefik
- Reverse proxy
- Upstream routing
- TLS termination
- Virtual hosts / service configuration
- HTTP routing (entrypoints, routers, services)
- Ingress-style routing

- 502 / upstream diagnostics and troubleshooting

Databases & Storage

- 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
- Automated provisioning
- Environment standardization
- Infrastructure versioning
- Infrastructure code reviews

DevOps & CI/CD

- GitHub Actions
- AWS CodePipeline
- AWS CodeBuild
- AWS CodeDeploy
- CI/CD pipeline design
- Automated builds
- Automated testing
- Automated 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
- Secret scanning
- Container scanning
- Infrastructure security scanning
- Container/IaC scanning
- Vulnerability management
- Vulnerability gates

Containerization

- Docker
- Dockerfiles
- Docker Compose
- Multi-stage Docker builds
- Container image optimization
- Container security
- Container security scanning
- Amazon ECR
- ECS
- ECS Fargate
- ECS task definitions
- Fargate profiles
- Service discovery
- Secrets
- Secrets injection
- Health checks
- Resource limits
- Resource limits & requests

- Autoscaling
- Container observability
- Container networking
- Container troubleshooting
- Blue/green & canary deployments

Kubernetes & Container Orchestration

- Cluster architecture (control plane, nodes, scheduling)
- Pods
- Deployments
- StatefulSets
- DaemonSets
- Jobs / CronJobs
- ReplicaSets
- Services (ClusterIP, NodePort, LoadBalancer)
- Ingress
- Ingress controllers
- Namespaces
- ConfigMaps
- Secrets
- PersistentVolumes
- PersistentVolumeClaims
- StorageClasses
- RBAC
- NetworkPolicy
- Resource limits & requests
- Liveness / readiness / startup probes
- Rolling updates & rollbacks
- Horizontal Pod Autoscaling (HPA)
- kubectl troubleshooting (get, describe, logs, events)
- Helm
- GitOps (Argo CD concepts)
- ECS-to-Kubernetes workload mapping
- Container networking
- Multi-environment cluster patterns
- Amazon EKS
- Platform engineering patterns for Kubernetes

DevOps Automation

- Bash
- Python
- Ansible
- Shell scripting
- Infrastructure automation
- Configuration management
- Environment automation
- Deployment automation
- Operational automation
- Automated remediation
- Scheduled automation
- AWS Systems Manager automation

- Lambda automation
- Event-driven automation

Security

- 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
- CSPM-style guardrails
- Security architecture reviews
- Tagging standards
- Automated remediation

Observability & Operations

- Amazon CloudWatch
- CloudWatch Logs
- CloudWatch Metrics
- CloudWatch Alarms
- CloudWatch Dashboards
- CloudWatch Events / EventBridge

- AWS CloudTrail
- AWS Config
- Distributed application monitoring
- Application performance monitoring
- APM concepts
- Infrastructure monitoring
- Log aggregation
- Centralized logging
- Alerting
- SLO-minded alerting
- Incident response
- Root-cause analysis
- Performance troubleshooting
- Capacity planning
- Availability monitoring
- SLI/SLO concepts
- Operational dashboards
- Automated remediation
- Production readiness reviews

Disaster Recovery

- 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 / Cloud Cost Optimization

- 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

Programming & Scripting

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

Generative AI & AI Engineering

- Generative AI
- Large Language Models
- Foundation Models
- Amazon Bedrock
- Amazon Bedrock AgentCore
- Amazon SageMaker AI
- RAG
- Retrieval-Augmented Generation
- Prompt Engineering
- Prompt Templates
- Model Evaluation
- AI Guardrails
- LLM application architecture
- AI-powered APIs
- AI agents
- Agentic AI
- Tool calling
- Conversational AI
- AI assistants
- Multimodal AI
- Embeddings
- Vector search
- Knowledge bases
- Document ingestion
- Document processing
- Document ingestion & chunking
- Retrieval pipelines
- Contextual prompting
- 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
- Bedrock + Lambda/API Gateway integration
- AgentCore Runtime / agent deployment patterns
- Agent tool integration and governance
- Tenant-aware AI retrieval

AI / RAG Architecture

Potential validated experience areas:

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

Technical Leadership

- 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
- Production readiness reviews

Professional Experience

Senior Cloud & DevOps Architect

Toumoro Group Inc. — Quebec, Canada

August 2023 – June 2026

Core Architecture & Technical Leadership

- Led end-to-end architecture for greenfield cloud-native products from technical discovery through production deployment.
- Designed scalable AWS architectures aligned with availability, security, performance, maintainability and cost requirements.
- Defined technical architecture standards and reusable engineering patterns across cloud projects.
- Translated business and technical requirements into cloud architecture and implementation plans.
- Conducted architecture reviews and evaluated technical trade-offs across competing implementation approaches.
- Established engineering standards covering infrastructure, application deployment, security and operational practices.
- Provided hands-on technical leadership to engineers throughout implementation and production rollout.
- Performed code and infrastructure reviews to enforce quality, security and maintainability standards.
- Mentored engineers on AWS architecture, Infrastructure as Code, DevOps, cloud security and AI engineering.

AWS Architecture

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

Infrastructure as Code

- Implemented infrastructure using AWS CDK with Python and TypeScript.

- Implemented infrastructure using Terraform.
- Developed reusable Infrastructure as Code components to standardize AWS deployments.
- Automated environment provisioning and configuration.
- Established version-controlled infrastructure workflows.
- Integrated infrastructure deployment into CI/CD pipelines.
- Reduced environment provisioning time from approximately five days to approximately one hour through infrastructure automation.
- Implemented infrastructure review practices to detect configuration and security issues before deployment.

DevOps & CI/CD

- Designed and implemented CI/CD pipelines using GitHub Actions and Ansible.
- Automated application build, validation and deployment workflows.
- Automated infrastructure deployment through CI/CD.
- Implemented environment promotion workflows across development, staging and production.
- Improved deployment consistency and reduced deployment errors.
- Established repeatable release processes for cloud-native applications.
- Integrated automated checks into delivery pipelines.
- Improved engineering feedback loops through automated deployment workflows.

Security

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

Observability & Operations

- Designed CloudWatch monitoring and alerting for production workloads.
- Implemented operational dashboards and application/infrastructure monitoring.
- Centralized operational logging.
- Defined monitoring requirements for critical cloud services.
- Supported production troubleshooting and root-cause analysis.
- Implemented automated operational responses where appropriate.
- Established production-readiness practices covering monitoring, logging, alerting and recovery.

Performance & Reliability

- Architected environments targeting greater than 99.95% availability.
- Designed architectures using multi-AZ deployment patterns.
- Implemented resilient application components and failure-isolation strategies.
- Evaluated performance bottlenecks across application and infrastructure layers.
- Implemented scalable architectures capable of responding to workload changes.
- Improved reliability through automation and standardized infrastructure.

Additional Achievements

- Designed multi-account AWS environments.
- Implemented centralized AWS logging and security monitoring.
- Implemented cross-account IAM access patterns.
- Designed private AWS service connectivity using VPC endpoints.
- Implemented automated compliance checks.

- Implemented infrastructure security scanning.
- Designed disaster recovery procedures.
- Automated backup and recovery workflows.
- Implemented cloud cost dashboards.
- Implemented automated detection of unused AWS resources.
- Established tagging and cost-allocation standards.
- Designed reusable internal AWS architecture patterns.
- Established standardized project bootstrapping templates.
- Created reusable CDK constructs and Terraform modules.
- Built internal developer platform capabilities.
- Implemented automated environment teardown for temporary environments.
- Designed ephemeral development environments.
- Implemented automated infrastructure testing.

Senior Cloud & DevOps Engineer — Contract

ITS Group — Lyon, France

January 2023 – April 2023

Cloud Architecture & Modernization

- Led technical architecture activities for AWS cloud modernization initiatives.
- Assessed existing application architectures and identified modernization opportunities.
- Designed cloud architectures supporting high-scale applications with more than 200,000 monthly active users.
- Evaluated workload requirements across availability, scalability, performance, security and operational complexity.
- Developed modernization approaches for legacy workloads.
- Supported migration of application components to AWS managed services.

Containerization

- Containerized legacy application workloads using Docker.
- Designed Docker-based deployment patterns for consistent application execution.
- Deployed containerized workloads using Amazon ECS.
- Standardized application environments across development, staging and production.
- Reduced environment drift through containerized deployment.
- Optimized container configurations for application reliability and operational consistency.

CI/CD

Potential validated responsibilities:

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

Observability

- Designed centralized monitoring using Amazon CloudWatch.
- Implemented application and infrastructure monitoring.
- Created operational dashboards.
- Configured alarms for critical services.
- Centralized application logs.
- Supported production incident investigation.

- Performed root-cause analysis of infrastructure and application failures.

Backup & Disaster Recovery

- Designed automated backup strategies.
- Implemented recovery procedures for critical workloads.
- Designed disaster recovery architecture around defined RPO/RTO requirements.
- Achieved recovery objectives targeting RPO below 15 minutes and RTO below one hour.
- Evaluated recovery dependencies across application, compute, storage and database layers.
- Supported disaster recovery testing and validation.

Additional Achievements

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

Cloud Architect & Technical Lead

RenovaCloud — Ho Chi Minh City, Vietnam

October 2020 – September 2022

Cloud Architecture

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

Serverless Architecture

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

Networking

- Designed AWS VPC architectures.
- Designed public and private subnet segmentation.
- Configured routing and security controls.
- Integrated Application Load Balancers with application workloads.
- Configured CloudFront distributions for application delivery.
- Designed DNS and traffic-routing architectures.

- Implemented WAF protection for internet-facing applications.

DevSecOps

- Led DevSecOps implementation across application delivery workflows.
- Embedded automated vulnerability scanning into CI/CD pipelines.
- Integrated security validation into software delivery processes.
- Reduced exposure to critical application security risks.
- Implemented secure deployment practices.
- Promoted security ownership across development and operations workflows.

Cost Optimization

- Led AWS cost optimization initiatives.
- Performed resource rightsizing and utilization analysis.
- Evaluated Savings Plans and other AWS pricing mechanisms.
- Reduced annual cloud expenditure by approximately 25%.
- Identified underutilized resources.
- Optimized cloud resource allocation based on workload requirements.
- Improved cost visibility across cloud environments.

Databases

Potential validated responsibilities:

- Designed Amazon RDS and Aurora architectures.
- Designed database high-availability configurations.
- Implemented database backup and recovery strategies.
- Optimized database resource sizing.
- Analyzed database performance.
- Implemented database monitoring.
- Designed secure database connectivity from application workloads.
- Implemented encryption for database workloads.

CDN & Edge

- Designed CloudFront architectures for application delivery.
- Configured origin integrations.
- Implemented caching strategies.
- Integrated AWS WAF with CloudFront.
- Improved application delivery performance through edge caching.
- Protected internet-facing workloads using edge security controls.

Additional Achievements

- Designed multi-account AWS environments.
- Implemented Terraform-based infrastructure provisioning.
- Built reusable Terraform modules.
- Implemented automated environment provisioning.
- Implemented GitHub Actions or AWS CodePipeline deployments.
- Implemented ECS/Fargate workloads.
- Implemented Docker-based application delivery.
- Designed centralized logging.
- Implemented CloudWatch monitoring.
- Implemented AWS CloudTrail.
- Implemented AWS Config.
- Implemented GuardDuty.
- Implemented Security Hub.
- Designed disaster recovery architectures.

- Implemented automated backups.
- Implemented cross-region replication.
- Implemented blue/green deployments.
- Implemented canary releases.
- Designed zero-downtime deployment architectures.

Earlier IT Experience

The current résumé states approximately 10 years of IT experience but only provides three detailed professional positions. The following should not be inserted as employment history until the corresponding employers, positions and dates are recovered from your actual career history.

Potential earlier-role competency areas to investigate:

- 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

Selectable Professional Achievements

Use only achievements you can substantiate.

Architecture

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

Automation

- Automated infrastructure provisioning using Terraform and AWS CDK.

- Automated application deployment through CI/CD.
- Automated environment configuration using Ansible.
- Automated operational workflows using Python and Bash.
- Reduced manual infrastructure operations through reusable automation.

DevOps

- Built CI/CD pipelines covering source control, build, testing, security validation and deployment.
- Standardized application deployment across development, staging and production.
- Reduced deployment errors through automated delivery workflows.
- Introduced infrastructure automation into engineering workflows.
- Improved deployment repeatability through Infrastructure as Code.

Security

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

Reliability

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

Cost

- Reduced annual AWS expenditure by approximately 25%.
- Implemented resource rightsizing.
- Evaluated Savings Plans.
- Optimized serverless and container workloads.
- Improved cloud cost visibility.

Leadership

- Established engineering standards for cloud development.
- Mentored engineers on AWS and DevOps.
- Conducted architecture reviews.
- Conducted code reviews.
- Led technical discovery sessions.
- Translated business requirements into technical architecture.
- Guided engineering teams through cloud modernization.
- Served as hands-on technical lead from architecture through production.

Master Project Bank

1. Enterprise AWS Landing Zone

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

Potential technologies:

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

2. Multi-Account AWS Security Platform

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

3. Cloud-Native SaaS Platform

Design and implement a multi-tenant SaaS platform using:

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

Focus on tenant isolation, scalability, observability and security.

4. Serverless REST API Platform

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

Include:

- Authentication
- Authorization
- API throttling
- Monitoring
- Logging
- Error handling
- Automated deployment
- Infrastructure as Code

5. Event-Driven Microservices Platform

Design an event-driven architecture using:

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

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

6. ECS Fargate Microservices Platform

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

Include:

- ECR
- IAM
- Secrets Manager
- CloudWatch
- Autoscaling

- Health checks
- CI/CD
- Blue/green deployments
- Image lifecycle policies
- Vulnerability scanning
- CloudWatch observability

7. Kubernetes Platform Engineering (EKS-Ready Architecture)

Built reusable Kubernetes platform patterns: cluster topology, workload standards (Deployments/Services/Ingress), ConfigMaps/Secrets, storage, RBAC, probes, autoscaling, GitOps-ready delivery, and operational runbooks—aligned with Terraform modules for VPC, IAM, and EKS provisioning.

Include:

- Cluster architecture (control plane, nodes, scheduling)
- Deployments, Services, Ingress
- ConfigMaps / Secrets
- PersistentVolumes / PersistentVolumeClaims / StorageClasses
- RBAC and NetworkPolicy
- Liveness / readiness / startup probes
- Horizontal Pod Autoscaling
- Helm
- GitOps (Argo CD concepts)
- Terraform modules for VPC, IAM, EKS
- Multi-environment cluster patterns
- kubectl operational troubleshooting

9. Infrastructure-as-Code Platform

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

Potential modules:

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

10. AWS CDK Architecture Framework

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

Potential components:

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

- WAF
- IAM

11. Automated CI/CD Platform

Build a reusable CI/CD platform using GitHub Actions.

Pipeline stages:

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.Application deployment 11.Smoke tests 12.Rollback

12. DevSecOps Platform

Integrate security into the entire software delivery lifecycle.

Potential capabilities:

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

13. Cloud Disaster Recovery Platform

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

Potential services:

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

14. Multi-Region AWS Architecture

Design an architecture capable of surviving a regional outage.

Potential components:

- Route 53
- CloudFront

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

15. AWS Cost Optimization Platform

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. Cloud Observability Platform

Build centralized observability around:

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

17. AI-Powered RAG Knowledge Assistant (Amazon Bedrock)

Build an enterprise knowledge assistant capable of answering questions over internal documents. Deliver enterprise RAG architecture with document ingestion, chunking, embeddings, vector retrieval, contextual prompting, grounding, guardrails, and secure API integration via Lambda/API Gateway with monitoring and access controls.

Architecture may include:

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

18. AI Customer Support Assistant

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

Potential capabilities:

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

19. AI Document Processing Pipeline

Build an automated document-processing platform.

Potential workflow:

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

Potential AWS services:

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

20. AI Agent Platform / Agentic AI & Tool-Calling Workflows (Amazon Bedrock AgentCore)

Build and operate AI agents capable of interacting with APIs and cloud services through controlled tools. Design agentic AI systems using Amazon Bedrock and Amazon Bedrock AgentCore with tool calling, retrieval, memory, authorization, API integrations, guardrails, audit logging, and observability for controlled automation across cloud and application workflows.

Potential capabilities:

- Amazon Bedrock AgentCore
- agent runtime / deployment patterns
- tool calling
- planning
- retrieval
- memory
- guardrails
- authorization
- API integrations
- observability
- audit logging
- agentic AI workflows
- controlled automation across cloud and application systems

21. Cloud & CI/CD Infrastructure AI Troubleshooting

Assistant Build an AI assistant capable of analyzing:

- 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. AI Cloud Cost Assistant

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

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

23. Automated AWS Security Remediation

Build an event-driven security remediation system.

Example:

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. Self-Service Developer Platform

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

Potential stack:

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

25. Ephemeral Development Environment Platform

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

Potential capabilities:

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

26. Zero-Downtime Deployment Platform

Implement deployment strategies including:

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

27. Secure Multi-Tenant SaaS Architecture

Design tenant isolation across:

- identity
- API authorization
- database
- storage

- logging
- encryption
- networking

Implement tenant-aware access controls and monitoring.

28. Cloud Migration Factory

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

Phases:

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

29. Legacy Application Modernization

Transform a monolithic application into:

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

while maintaining business continuity.

30. Private AWS Application Platform

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

Potential components:

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

31. Application Access Log Analytics Pipeline (Kinesis → S3 → Athena)

Centralized application log ingestion and queryable analytics on AWS.

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

Certifications

Certifications Already Listed on the Current Résumé

- AWS Certified Cloud Practitioner

Additional Badges AWS

- 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

Education

Master’s Degree — Networks, Systems and Security

Claude Bernard University Lyon 1 — France

Master’s Degree — Networks and Communication Systems

Vietnam National University — Vietnam

Languages

- French — Native
- English — Professional proficiency