

Driving Cost-Efficient and Secure Cloud Transformation in HR with Terraform and CloudWatch

Re-engineering AI-driven recruitment infrastructure with Infrastructure as Code and native AWS observability to ensure high security and operational transparency.

Overview

We automated the provisioning of an AI-driven recruitment platform using Terraform and Jenkins, cutting production setup time to just two weeks.

- Developed a custom, native AWS observability framework using CloudWatch and Lambda to deliver real-time incident alerts directly into Google Chat.
- Eliminated third-party licensing fees and security risks by enforcing role-based communication and VPC endpoints within an AWS-native ecosystem.



Client Profile

This global HR powerhouse connects talent with fitting roles across hundreds of industries. They focus on delivering high-speed hiring solutions through technology, managing a massive portfolio of candidates and business partners worldwide.

Challenges: Security Constraints and Monitoring Gaps

The client's transition to an AI-led platform was hindered by restrictive legacy processes.

- **Manual Bottlenecks:** A limited-permission model required all infrastructure to be set up manually via the AWS console, making environment replication nearly impossible.
- **Observability Deficit:** Minimal system monitoring existed, with no structured dashboards or alerting mechanisms to track application health.
- **Dependency Concerns:** A strict requirement to use AWS-native solutions to avoid third-party tool sprawl and rising licensing costs.
- **Scripting Latency:** The manual effort required for complex DevOps scripting slowed down the release cycle for new recruitment features.

QBurst Solution: Automated IaC and Native Observability

We introduced Terraform to transform the client's infrastructure into a scalable, version-controlled environment. This was integrated with their Jenkins pipeline to allow controlled, automated deployments that adhered to strict IAM permissions. To address monitoring, we engineered a "fit-for-purpose" observability suite using only native AWS services.

Key Technical Implementations:

- **Infrastructure as Code (IaC):** Defined all networking and compute resources through Terraform modules to eliminate configuration drift and manual errors.
- **Real-Time Alerting Pipeline:** Configured CloudWatch alarms routed through SNS and Lambda to push critical notifications into Google Chat, ensuring immediate visibility into system health.
- **Security Hardening:** Implemented well-defined security group rules and role-based access controls (RBAC) to replace the previously manual and insecure credentialing.
- **GenAI-Assisted DevOps:** Leveraged tools like GitHub Copilot and Gemini to accelerate the creation of Terraform scripts and automation workflows.

Technical Highlights

- **Consistent Replication:** Terraform modules enable the instant setup of identical staging and production environments.
- **Native AWS Approach:** Zero reliance on external monitoring tools, reducing the client's annual licensing overhead.
- **Hardened Infrastructure:** Secured communication via VPC endpoints and documented, reviewed security group rules.
- **Unified Dashboard:** A custom CloudWatch interface providing a single pane of glass for performance metrics and incident tracking.

Impact

- **75% Faster Provisioning:** Reduced the timeline for production environment setup from several months to just two weeks.
- **Stronger Governance:** Every infrastructure change is now trackable and aligned with the client's global security policies.
- **Rapid Incident Response:** Real-time Google Chat alerts have significantly lowered the Mean Time to Repair (MTTR) for critical application issues.
- **Cost Optimization:** By using open-source Terraform and native AWS tools, the client avoided significant third-party software expenses.
- **Scalable Foundation:** The reusable modules and monitoring templates now serve as the blueprint for all future HR-tech projects.