DEVX360

Sipho Sehlapelo
Kesley Hamann
Sibusiso Mngomezulu
Lwando Msindo
David Musa-Aisien

DevX360 API — Deployment Model (AWS Lambda + API Gateway)

Purpose: provide a clear deployment model for hosting the DevX360 Express API on AWS using Lambda + API Gateway (HTTP API), optimized for staying within the AWS Free Tier and for reproducible CI/CD deployments.

Goals & Constraints

- **Goals:** serverless hosting (minimal ops), predictable cost (free-tier conscious), secure secrets management, simple infra-as-code, GitHub-based CI/CD using OIDC.
- **Constraints:** prefer af-south-1 (Cape Town) region, keep memory and timeout conservative (MemorySize 256 MB, Timeout 10s), use MongoDB where possible to reduce RDS costs, and keep package size under Lambda limits.

High-level architecture

- The Express app runs inside a Lambda using @vendia/serverless-express adapter (lambda.js with exports.handler).
- API Gateway HTTP API is preferred (cheaper, free-tier friendly) and configured to route ANY / {proxy+} to the Lambda.

Components & responsibilities

- **API (Express)**
  - Exports an Express app (e.g., app.js exports app).
  - lambda.js (or index.js) wraps the Express app with @vendia/serverless-express and exports handler.
  - Environment variables read DB url and secrets from SSM resolver strings (no secrets in code).
- **AWS Lambda**
  - Runtime nodejs20.x.
  - Memory: 256 MB. Timeout: 10s (tune only if needed).
  - Tracing: Active (X-Ray optional).
- **API Gateway (HTTP API)**
  - Terminate HTTPS, provide route ANY /{proxy+}.
  - CORS enabled when frontend calls cross-origin.
- **Secrets & Config**
  - Use **SSM Parameter Store** for DB_URL and JWT_SECRET. Use SecureString for secrets.
  - Use resolver syntax in SAM template: "{{resolve:ssm:/devx360/api/DB_URL}}".

- **Database**
    - **Default:** DynamoDB for cost control and free-tier friendliness.
    - **Optional:** RDS (db.t3.micro) only if relational features are required; monitor free-tier usage carefully.
- **Monitoring & Logging**
    - CloudWatch Logs captures Lambda logs automatically.
    - CloudWatch Metrics: invocations, duration, errors, throttles.
- **IAM & Deployment**
    - Prefer GitHub OIDC for GitHub Actions to assume a deploy role in AWS (no long-lived credentials in GitHub).
    - Minimum Lambda deployment policies: lambda:* (or scoped), apigateway:* (or scoped), cloudformation:*, ssm:GetParameter, ssm:GetParametersByPath, ssm:PutParameter (if CI writes parameters), and cloudwatch:* for log/group creation if required.

Security best-practices

- Do **not** commit secrets to Git. Use SSM SecureString.
- Use least-privilege IAM policies for the OIDC role (scope resources to the stack or resource ARNs where possible).
- Enable HTTPS only (API Gateway handles TLS).
- Rotate secrets and update SSM values as needed.

Monitoring, tracing & alerts

- Use CloudWatch Alarms for elevated error rate and high durations.
- Export basic dashboards (invocations, errors, duration, throttle count).
- (Optional) enable X-Ray for deeper tracing—be mindful of extra costs.

Free-tier cost controls

- Keep Lambda memory low and short timeouts.
- Use HTTP API (cheaper than REST API Gateway).
- Prefer DynamoDB (on-demand or provisioned with autoscaling). Monitor RDS usage if used.
- Set AWS Budgets with alerts (as per your one-time setup).

Testing & validation

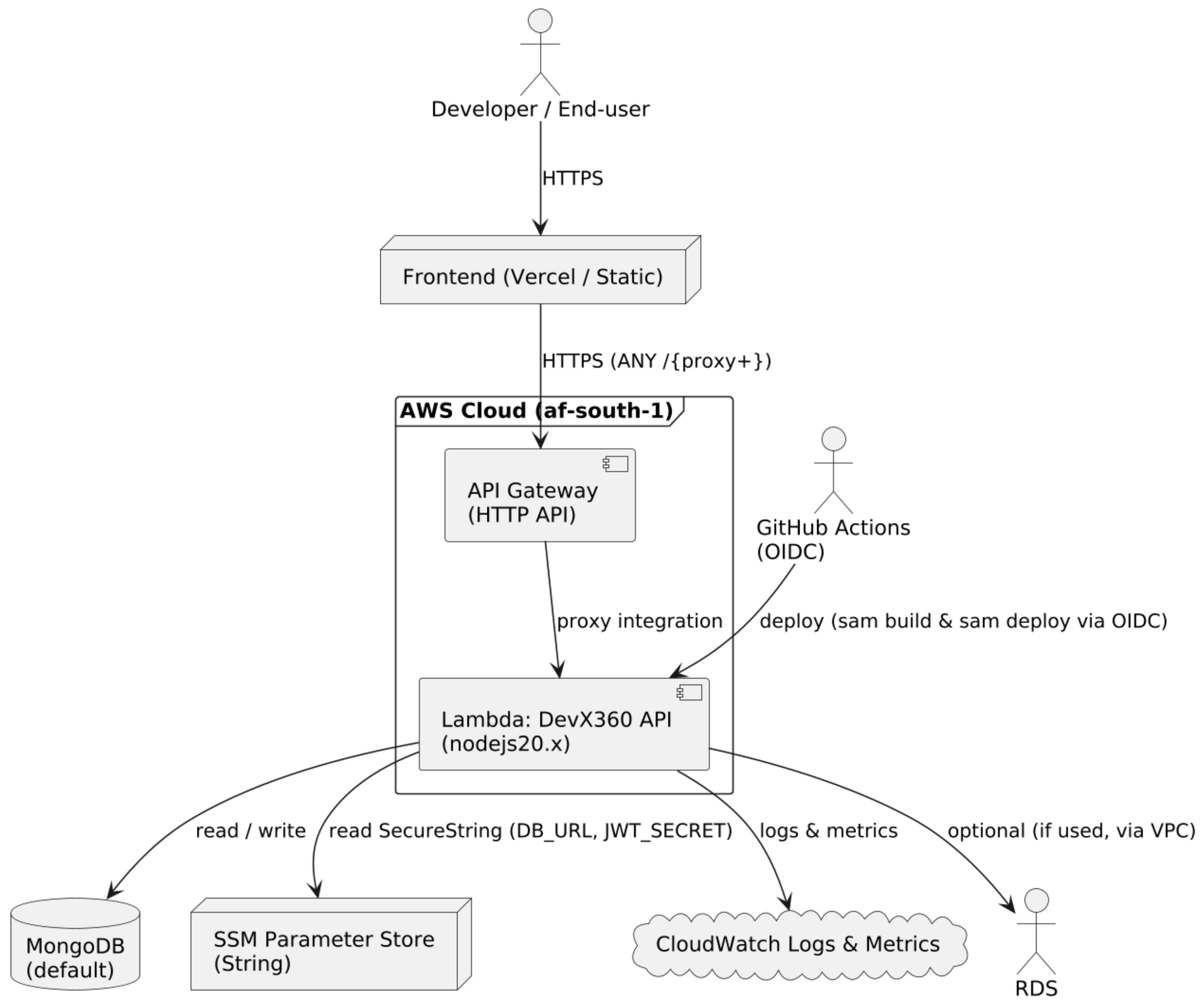
- After deployment, test GET {ApiUrl}/health and expected endpoints.
- Run simple load tests that stay within Free Tier limits.

- Validate environment variables: Lambda console -> Configuration -> Environment variables (or read via SSM resolver).

Rollback & versioning

- Use CloudFormation stack versions and sam deploy stack names per stage (e.g., DevX360-API-dev, DevX360-API-prod).
- For quick rollback, re-deploy a previous commit or use CloudFormation to roll back to the previous stack template.

Diagram


Developer / End-user
HTTPS
Frontend (Vercel / Static)
HTTPS (ANY /{proxy+})
AWS Cloud (af-south-1)
API Gateway (HTTP API)
proxy integration
GitHub Actions (OIDC)
deploy (sam build & sam deploy via OIDC)
Lambda: DevX360 API (nodejs20.x)
read / write
read SecureString (DB_URL, JWT_SECRET)
logs & metrics
optional (if used, via VPC)
MongoDB (default)
SSM Parameter Store (String)
CloudWatch Logs & Metrics
RDS