

Automated Data Management System on AWS

Michael Ohene Aboagye

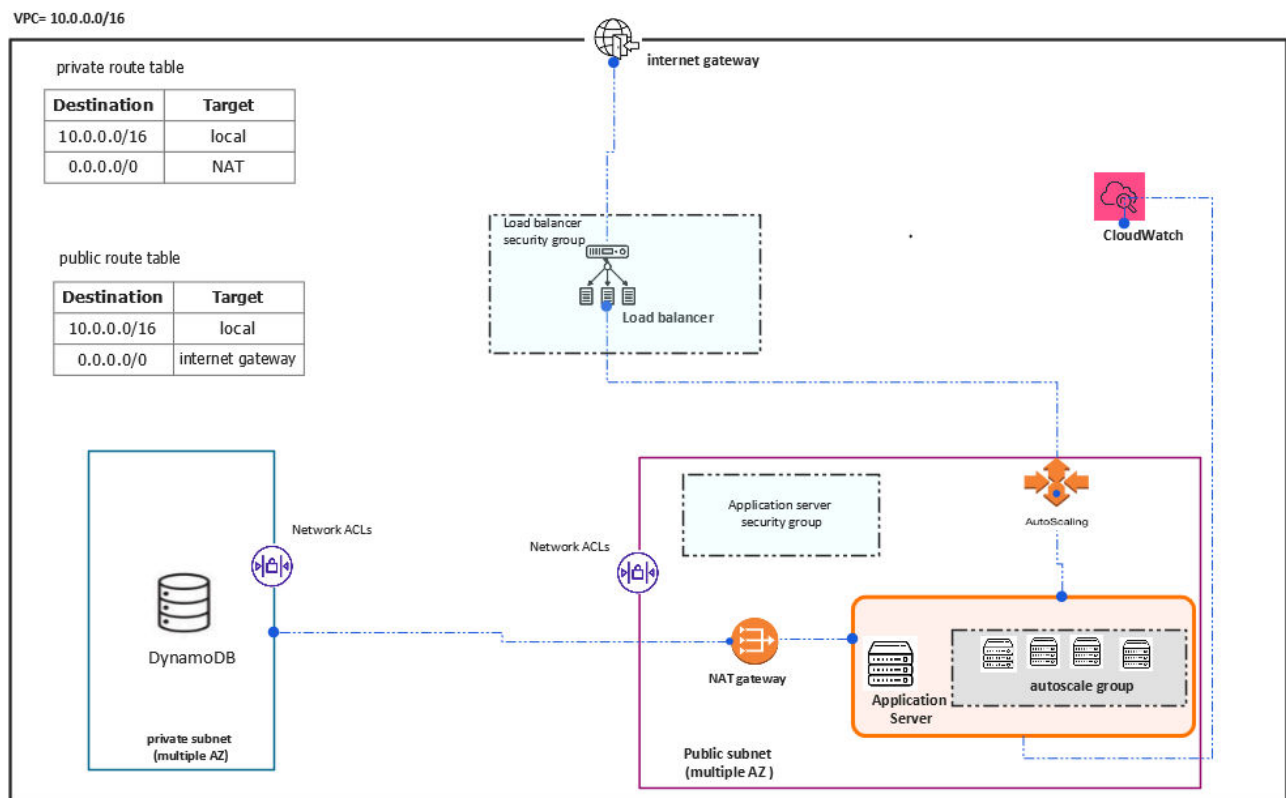
Cloud Engineer

7/7/2025

SUMMARY

This project establishes a logger database using **DynamoDB** and implements security and scalability features using **EC2**. Key components include a custom **VPC**, load balancing, and auto-scaling. Python 3 and Boto3 were used for data logging.

Automated Data Management Architecture Diagram



1. SETTING UP THE NETWORK ARCHITECTURE

1.1 Create a VPC

- VPC with CIDR block: 10.0.0.0/16.

1.2 Setup Subnets

- **Public Subnets:**
 - CIDRs: 10.0.1.0/24, 10.0.3.0/24.
- **Private Subnets:**
 - CIDRs: 10.0.2.0/24, 10.0.4.0/24.

1.3 Configure Route Tables and NAT Gateways

- **Public Route Table:** Routes to Internet Gateway.
- **Private Route Table:** Routes to NAT Gateway.

1.4 Setup Security Groups

- **Load Balancer Security Group:** Allows HTTP, HTTPS, SSH.
- **EC2 Instance Security Group:** Allows traffic from the load balancer only.

2. PREPARE THE EC2 INSTANCE

2.1 Create EC2 Instance

- Configure instance and test connectivity (e.g., run Nginx).

2.2 Create Launch Template

- Used by auto-scaling to create preconfigured instances.

2.3 Setup IAM Role

- Created role (AmazonDynamoDBFullAccess) and attached to the instance.

3. SETUP LOAD BALANCER AND AUTOSCALING GROUP

3.1 Configure Load Balancer

- Set listeners and attach the security group.

3.2 Configure Auto-Scaling Group

- Desired capacity: 2, Minimum: 1, Maximum: 4.

3.3 Test Connectivity

- Validate EC2 instance connectivity via load balancer DNS.

3.4 Test Auto-Scaling Response

- Applied stress test; auto-scaling responds to CPU thresholds.

4. SETUP DYNAMODB AND PYTHON CONFIGURATION

4.1 Install Boto3

```
pip install boto3
```

bash

4.2 Create Python Script

- **Staff_logger.py**: Automates logging of staff information to DynamoDB.

4.3 Run the Script

- Verify data insertion into DynamoDB.

CONCLUSION

The project demonstrates AWS capabilities in creating a scalable, secure architecture. By leveraging Auto Scaling, ELB, and DynamoDB, we ensured efficient workload management and low-latency data access, adhering to best practices for cost-efficiency and resilience.

Detailed steps on how to create an automated data management system using AWS can be found in the directory **/docs/Automated-Data Management-System-on-AWS.pdf** in the GitHub repository.