AWS Serverless URL Shortener – POC Report

1. Executive Summary:

This Proof of Concept (PoC) demonstrates the design and implementation of a **low-cost, scalable, and serverless URL shortener application** built entirely on **Amazon Web Services (AWS)**.

The solution provides:

- URL shortening service with unique IDs.

- Redirection to original URLs.

- Statistics tracking (click counts, creation date).

- A simple web-based dashboard hosted on S3.

This PoC highlights AWS's ability to deliver **production-grade applications** with **minimal cost**, using a **pay-per-use model** and fully managed services.

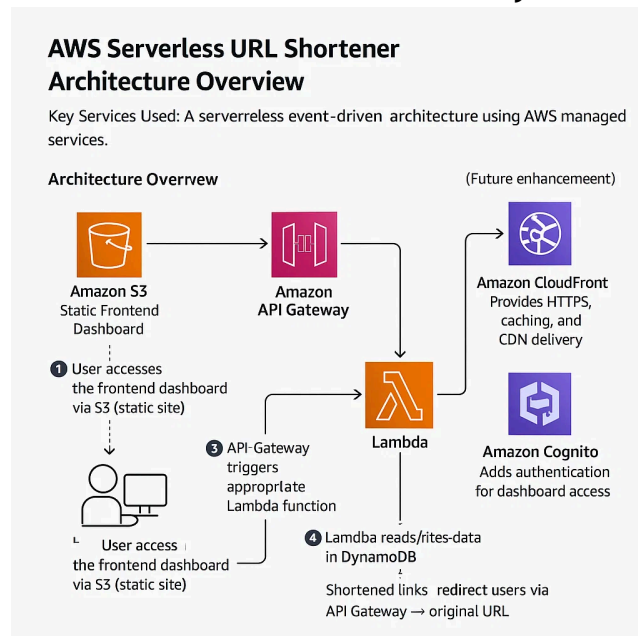# AWS Serverless URL Shortener – POC Report

2. Architecture Overview

The solution follows a **serverless event-driven architecture** using AWS managed services.

Key Services Used:

- **Amazon S3**: Hosts the static frontend dashboard.
- **Amazon API Gateway**: Serves as the API entry point.
- **AWS Lambda**: Executes backend logic for shortening URLs, redirecting, and fetching stats.
- **Amazon DynamoDB**: Stores URL mappings and click statistics.
- **Amazon CloudFront** *(future enhancement)*: Provides HTTPS, caching, and CDN delivery.
- **Amazon Cognito** *(future enhancement)*: Adds authentication for dashboard access.

Data Flow:

1. The user accesses the frontend dashboard via **S3 (static site)**.
2. Dashboard calls APIs exposed via **API Gateway**.
3. API Gateway triggers appropriate **Lambda function**.
4. Lambda reads/writes data in **DynamoDB**.
5. Shortened links redirect users via API Gateway → Lambda → original URL.

AWS Serverless URL Shortener Architecture Overview

Key Services Used: A serverreless event-driven architecture using AWS managed services.

Architecture Overrrvew   (Future enhancemeent)

Amazon S3
Static Frontend Dashboard

Amazon API Gateway

Amazon CloudFront
Provides HTTPS, caching, and CDN delivery

❶ User accesses the frontend dashboard via S3 (static site)

Lambda

Amazon Cognito
Adds authentication for dashboard access

❸ API-Gateway triggers approprlate Lambda function

User access the frontend dashboard via S3 (static site)

❹ Lamdba reads/rites-data in DynamoDB

Shortened links redirect users via API Gateway → original URL

AWS Serverless URL Shortener – POC Report

3. Implementation Details

3.1 Backend APIs

- **POST /shorten** → Creates a shortId for a given long URL and stores it in DynamoDB.

- **GET /{shortId}** → Redirects to the original long URL and increments the click counter.

- **GET /stats/{shortId}** → Returns statistics (long URL, click count, creation date).

3.2 Lambda Functions

- **shortenUrl** → Generates unique ID, stores mapping.

- **redirectUrl** → Handles HTTP 302 redirection.

- **getStats** → Fetches statistics from DynamoDB.

3.3 Frontend Dashboard

- Built using **HTML + JavaScript**, hosted on S3.

- Provides forms to shorten URLs and fetch statistics.

- Integrated with API Gateway via `fetch()` API calls.

- Implements CORS handling for API requests.

AWS Serverless URL Shortener – POC Report

4. Results & Testing

- **Short URL Creation**
  Input: `https://www.google.com`
  Output:
  `https://1mamyoez59.execute-api.ap-south-1.amazonaws.com/dev/shortid/`

- **Redirection**
  Visiting `https://.../shortid` → redirects to `https://www.openai.com`.

- **Stats API**
  Output:

```json
{

  "shortId": "abc123",

  "longUrl": "https://www.openai.com",

  "clickCount": 5,

  "createdAt": "2025-09-02T12:00:00Z"

}
```

Frontend Dashboard

- Successfully created short links.
- Redirected to original URLs.
- Displayed formatted JSON stats.

AWS Serverless URL Shortener – POC Report

AWS Serverless URL Shortener – POC Report
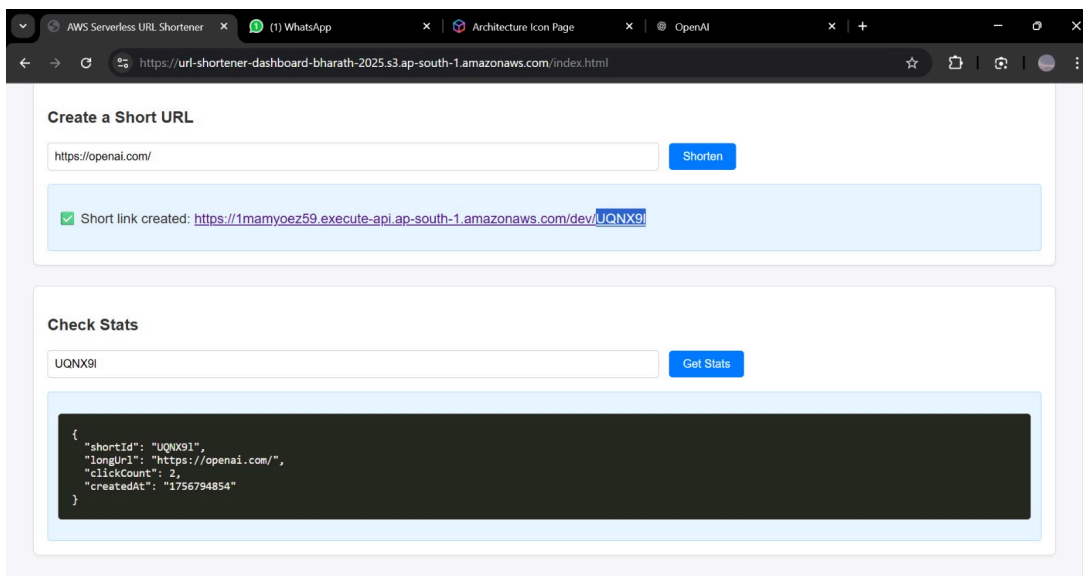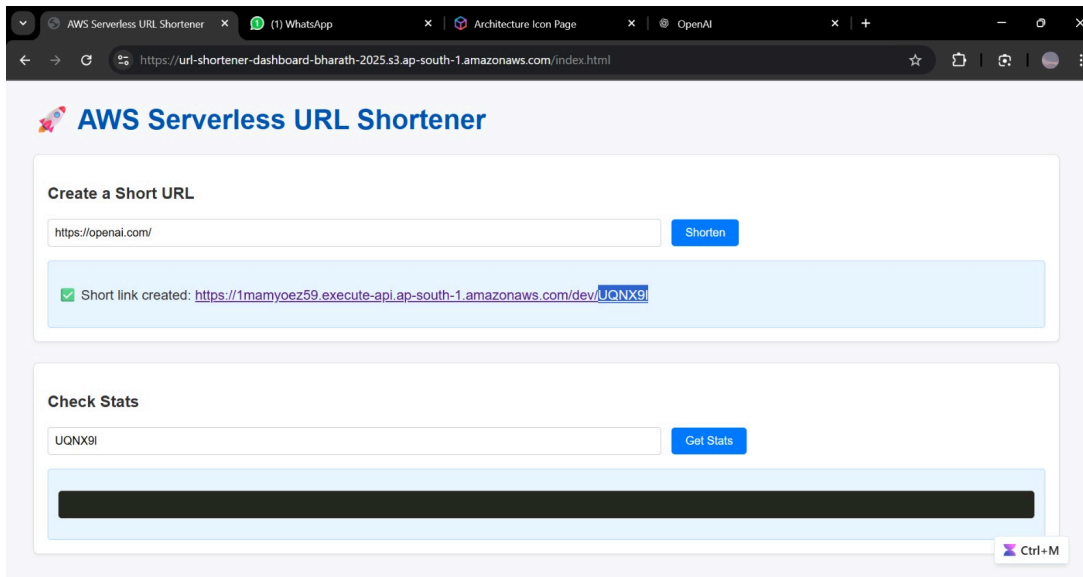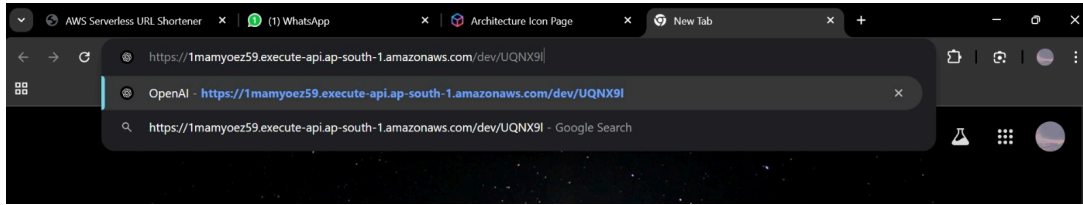
AWS Serverless URL Shortener – POC Report

5. Cost Analysis

This PoC incurs **near-zero cost** under the AWS Free Tier:

- **S3**: Hosting static files → few MBs storage, negligible.
- **API Gateway**: 1M free requests/month.
- **Lambda**: 1M free invocations + 400,000 GB-sec/month.
- **DynamoDB**: Free tier (25GB storage + 25 WCU/RCU).

Estimated cost after Free Tier: **<$1/month** for light usage.

6. Future Enhancements

- **Cognito Authentication**: Secure the dashboard with user login.
- **CloudFront Distribution**: Provide HTTPS and better performance.
- **Custom Domain**: Map short links to
  `https://short.mydomain.com/{id}`.
- **Analytics Dashboard**: Aggregate clicks, time-based charts.
- **QR Code Generation**: Auto-generate QR codes for each short URL.

AWS Serverless URL Shortener – POC Report

7. Conclusion

This PoC successfully demonstrates how AWS services can be combined to build a **scalable, secure, and cost-effective URL shortener** with analytics.

It showcases **serverless best practices**:

- Event-driven design
- Fully managed services
- Pay-per-use billing model
- Zero infrastructure management

This solution can be extended to production with minimal modifications, making it a strong candidate for real-world deployment.