

Design and implement a short-lived URL shortener solution with AWS

Scenario:

In Company M, employees often share internal documents via URLs. However, these URLs are often lengthy and non-descriptive, so there is a need for a URL shortening solution for convenience and clarity. As the documents are confidential, free online services like Bitly.com are prohibited. As the solution architect, you are tasked with designing and implementing a Proof of Concept (PoC) URL shortener using AWS. The url should be auto-generated, and expire after 10 minutes, as the documents are expected to be opened shortly. Operational efficiency is important and the company wants to manage as little infrastructure as possible.

Functional requirements:

- Provide a simple web interface for users to submit long URLs and receive shortened versions of it
- Generate short URLs in the format: `http://xxxxx.xxx/h2so4`
- The last segment of the short URL should be a random 5-character string comprising letters and digits
- Do not care about the domain for this PoC
- Each short URL should be valid for 10 minutes after creation

Infrastructure requirements:

- Implement Infrastructure as Code (IaC) to build the solution (preferably using AWS SAM, AWS CDK, or Terraform)
- Utilise Amazon API Gateway for REST API implementation
- Implement CRUD logic using AWS Lambda (in your preferred programming language)
- Leverage additional AWS services as needed based on your design

Deliverables:

- Architecture diagram of the solution
- IaC templates and associated code
- Detailed steps for building, deploying and running the application
- URL of the deployed web interface

Submission:

Create a private GitHub repository containing the diagram, steps, and URL in the README file, along with all scripts and code. Invite the following GitHub users as collaborator: `ricky-mzc`, `nicklee76`

Optional Enhancements (Bonus):

- Only expire the short url if it has not been accessed for 10 min, i.e. every time when users access it, start countdown of 10 min all over again
- Allow users to customise the short URL suffix, instead of random generation
- Calculate the cost of running this setup and write down the calculations (excluding free tier quotas), feel free to make assumptions
- Display a custom error page for non-existing short URLs, something like this:
`https://bit.ly/a-link-that-does-not-exist-30624700`
- Only allow users with IP range from `218.189.44.128 - 218.189.44.255` to submit/access short URLs
- Implement additional security measures, and document them in README
- Develop a strategy for handling collisions between newly generated and existing short URLs.
- Find a way to handle collisions between newly generated and existing short URLs
- Set up a CI/CD pipeline from GitHub to AWS
- Assess the limitations on concurrent users and propose scaling strategies for larger user bases
- Suggest potential improvements if additional time and resources are given

Note: Feel free to make reasonable assumptions where necessary