Lomba Kompetensi Siswa Sekolah Menengah Kejuruan Tingkat Provinsi Jawa Barat Tahun 2022

Modul 2 - Static Website

July 21, 2022

Bidang Lomba Cloud Computing

1 Overview

You have been tasked to deploy a front-end web application on AWS. The application is built using Next.js without server-side rendering, thus can be be hosted on S3. The application also requires API call to previous task (modul 1). If you fail to deploy the previous task, please ask us for a working API endpoint.

2 General Rules

1. Failure to comply with the rules will result in immediate disqualification.
2. You have 2 hours to finish the tasks.
3. You may not open any website unless otherwise specified in section 6 and you may open the control panel of your domain provider to update the nameserver to Route 53.
4. You may use AWS Console and AWS CLI to deploy the solutions. You may not use SAM, CloudFormation or CDK.
5. Between and after the event, you may not access your account. Any activity on your AWS account during this period is not allowed.
6. During the event, multiple login is not permitted.
7. This module requires compilation, you can use an EC2 instance or your own computer to compile the project.
8. If you have any question, do not hesitate to ask.

3 Architecture

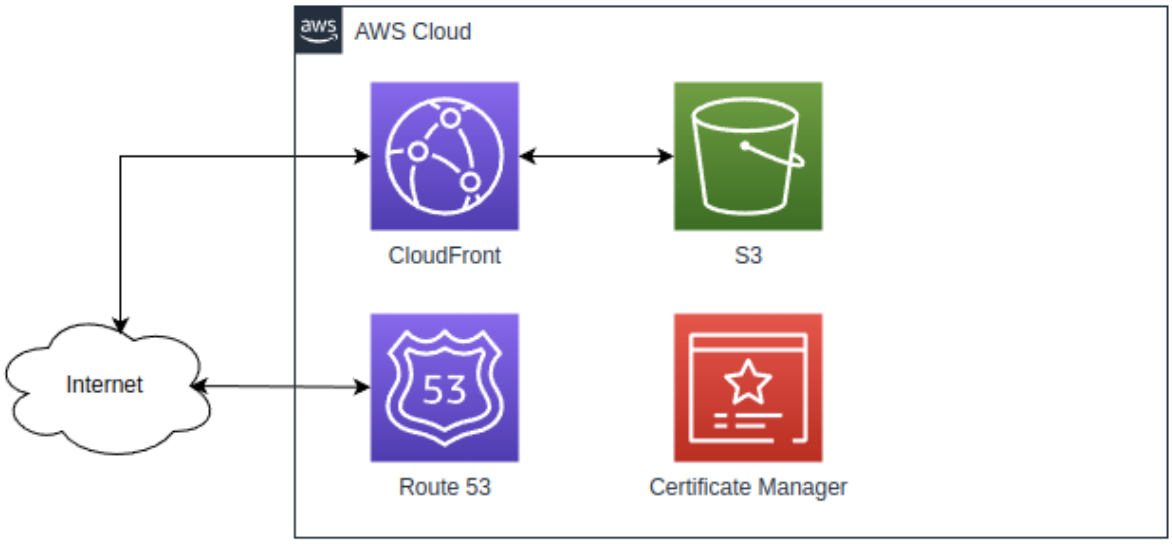

Figure 1: Architecture Diagram

4 Information

1. The repository URL for the project is https://github.com/kensasongko/lksccjabar2022modul2_aplikasi

2. The URL for the CloudFront Function code is
https://gist.github.com/kensasongko/610912a2b9a645b00743c773a00848ea

3. This solution must be deployed in **us-east-1 (N. Virginia) region**. Deploying in another region will result in a major point reduction.

5 Task

1. Create a standard S3 bucket with the following specifications:
   - Bucket name: modul2.[YOUR_DOMAIN]
     If the bucket name is taken, you may append a 6-character random string to the end of the bucket name.
   - Tags: Key=LKS-ID, Value=MODUL2
   - Block all public access: True
   - Bucket versioning: Disabled
   - Server-side encryption: Enabled
   - Encryption key type: Amazon S3-managed keys
2. Clone the repository, follow the instruction from README.md to generate the static files.
3. Upload the generated static files to the S3 bucket.
4. Create a CloudFront Distribution with the following specifications:
   - Origin domain: Your S3 bucket
   - Use origin access identity to access the s3 bucket
   - Enable Origin Shield: No
   - Viewer protocol policy: Redirect HTTP to HTTPS
   - Caching Policy: CachingOptimized
   - Tags: Key=LKS-ID, Value=MODUL2
5. Create a certificate from ACM using DNS validated
   - Domain Name: modul2.[YOUR_DOMAIN]
   - Validation Method: DNS validation
   - Tags: Key=LKS-ID, Value=MODUL2
6. Configure CloudFront distribution to use custom domain modul2.[YOUR_DOMAIN].
7. Create and publish a CloudFront Function using the code provided in number 2 from Section 4.
   Associate the function with your distribution.
   Make sure you don't have to append index.html to access the website.
8. Open pages/index.js with any text editor, replace the text "My Simple TODO Application" with "Peserta_[YOUR_PARTICIPATION_ID] Simple TODO Application"
9. Rebuild and re-export the application, re-upload the static page to the S3 bucket.
10. Invalidate CloudFront cache to make sure the cache is renewed.

6 References

- S3 documentation
- CloudFront documentation
- CloudFront Function documentation
- Route 53 documentation
- Certificate Manager documentation
- npm documentation
- yarn documentation
- Next.js documentation

Good luck!