Location of the Code :

https://github.com/Jagath-Wicks/Cloudfront_apigw_lambda

Technical Challenge

Requirement : Using Infrastructure as Code to create a page that serves a html page.

The content of the page must be ..

<h1>The saved string is dynamic string</h1>

This solution can be deployed in several always

1. EC2
2. AWS Apprunner
3. AWS APIGW + Lambda function -> Parameter Store
4. WAF -> CloudFront -> WAF -> APIGW - Lambda - Parameter Store

My architecture design based on following AWS well Architected pillars :

- Security
- Cost
- Reliability
- Operational excellence

I have implemented Option 4 and 3.

<u>Option 4 implementation details (my recommended way of implementation)</u>

User request comes to the CloudFront, CloudFront has a WAF implemented to protect the endpoint with DDOS attacks. Cloudfront sends the request to the APIGW.

When the Cloudfront sends the request to the APIGW, CloudFront sends a secret in the custom headers (name = "x-cloudfront-secret") to make sure that APIGW endpoint not directly accessible by public or any other service.

APIGW has a WAF implemented and the WAF checks for the correct header values pass from the CloudFront.Then the APIGW sends the request to the Lambda and Lambda displays the Html content.

Traffic from client to Cloudfront is https as CloudFront has TLS certificate.

APIGW endpoint is not exposed to the public and APIGW endpoint can only be accessible via the Cloud Frontend point.

This implementation meets the following AWS well Architected pillar recommendations

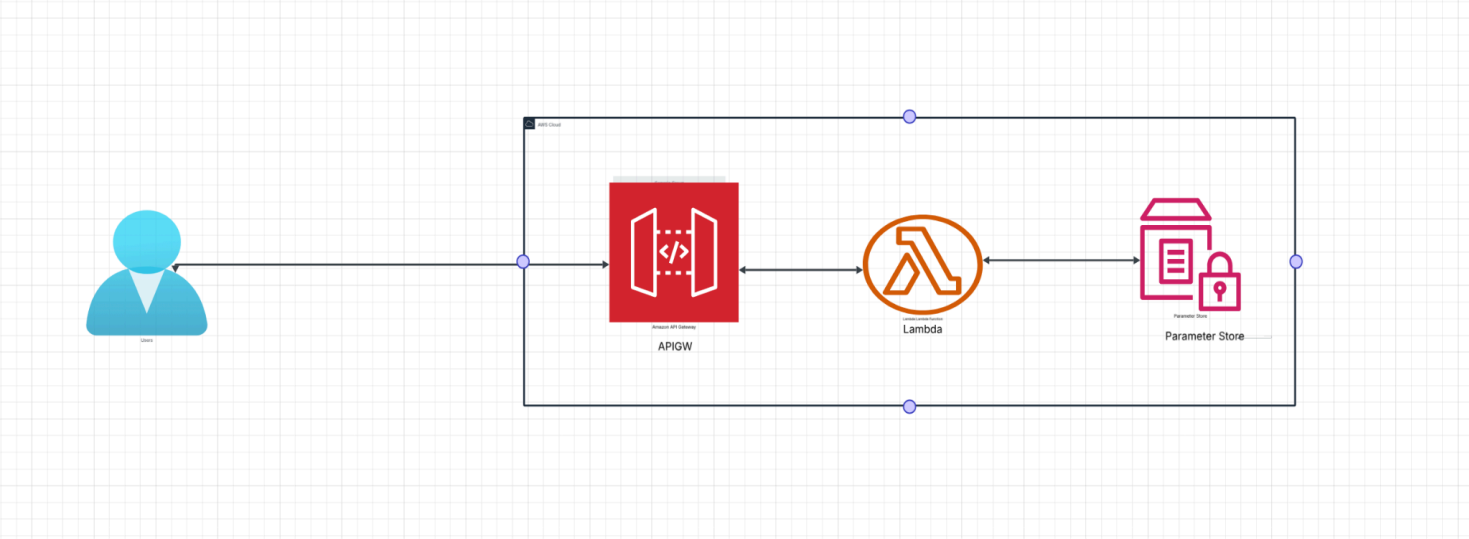
- Security - Client to the Cloudfront is https encrypted, WAF included for the ddos protection and CloudFront drops lots of unwanted traffic.Other WAF can be implemented.
- Cost - Lambda cost - only when invoke the url
- Reliability - All services are AWS managed
- Operational excellence - Cloudfront and has good tools to monitor traffic

Improvements:

Terraform code can be refactored, to create modules for each service.

Option - 3 - Architecture diagram

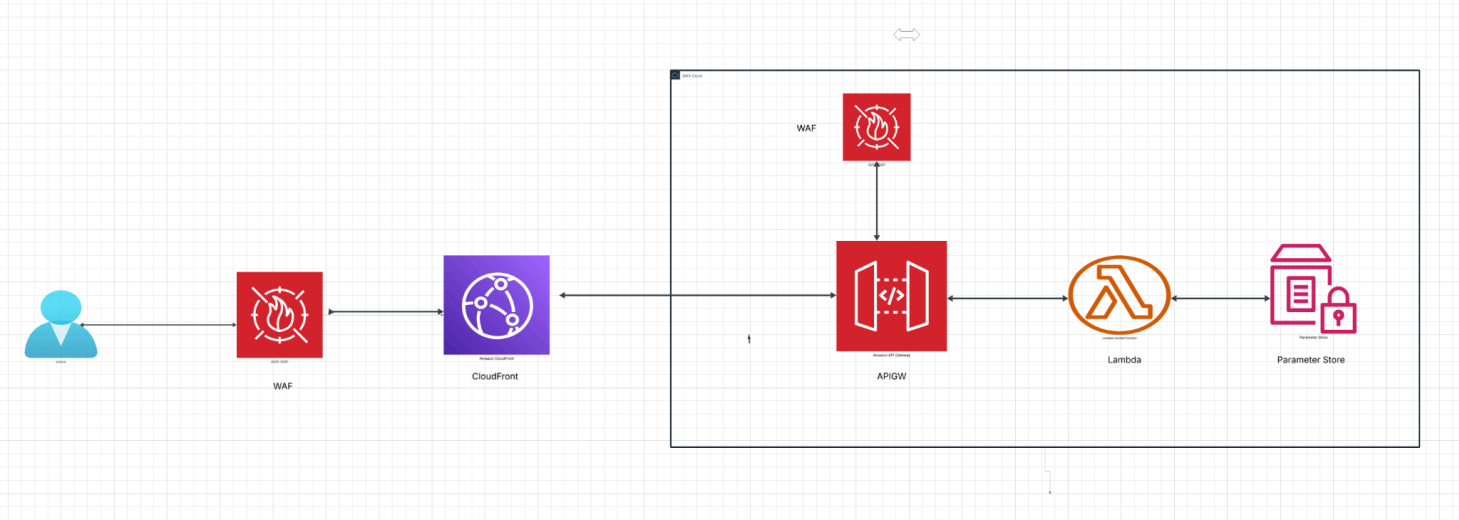
Option - 3 - AWS APIGW + Lambda function -> Parameter Store (less secure implementation)



Option - 4 - Architecture diagram

Option 4 - WAF -> CloudFront -> WAF -> APIGW - Lambda - Parameter Store

(More Secure Implementation)



Commands & how to run

Option - 3 - AWS APIGW + Lambda function -> Parameter Store (less secure implementation)

git clone https://github.com/Jagath-Wicks/Cloudfront_apigw_lambda.git

cd Cloudfront_apigw_lambda/

cd apigw_lambda/

terraform init

terraform apply

curl https://n5abilfzqi.execute-api.us-east-1.amazonaws.com/prod/html

// use the parameter store to store new value

aws ssm put-parameter --name "/dynamic_string" --value "Jagath" --type "String" --overwrite

curl https://n5abilfzqi.execute-api.us-east-1.amazonaws.com/prod/html

Terraform destroy

Option 4 - WAF -> CloudFront -> WAF -> APIGW - Lambda - Parameter Store

(More Secure Implementation)

Cd ../../cloudfront_apigw-lambda

terraform init

terraform plan

terraform apply

curl https://so8fyozx0a.execute-api.us-east-1.amazonaws.com/prod/html - **this fails as there is no direct access to the apigw**

Access CloudFront

curl https://dzjwbmqzbfk5c.cloudfront.net/html

Change the string

aws ssm put-parameter --name "/dynamic_string" --value "Jagath" --type "String" --overwrite

curl https://dzjwbmqzbfk5c.cloudfront.net/html

terraform destroy