stormit powered by aws

Sharp EDGE Budapest

2025

📅 Thursday, March 6, 2025

📍 Larus Event & Catering, Budapest

INTRO

How Stormit Helps Businesses Optimize AWS Edge Services

ADAM NOVOTNÝ
AWS Solutions Architect

adam@stormit.cloud

Sharp Edge Budapest

AGENDA

Focused on real-world scenarios from our customers

01 Intro

02 AWS WAF and Shield Advanced

03 Amazon CloudFront and Edge Functions

The AWS Edge Services Knowledge Gap

- → Many AWS users are unfamiliar with the AWS Edge Services portfolio
- → Specifically deeper functionality, use cases, and cost optimization

Common AWS Edge Service Challenges

- → Underutilization of features
- → Over-provisioning, resulting in unnecessary costs
- → Finding the best, easiest, and most cost-effective solution for a specific need.

WAFR

AWS Well-Architected Review with Stormit

aws

PARTNER
Well-Architected
Partner Program

Find ways to optimize costs, improve application performance and mitigate any security risks.

01

Identify risks and improvement opportunities

02

Identify opportunities for cost savings

03

Improve your security posture by reviewing your workloads

04

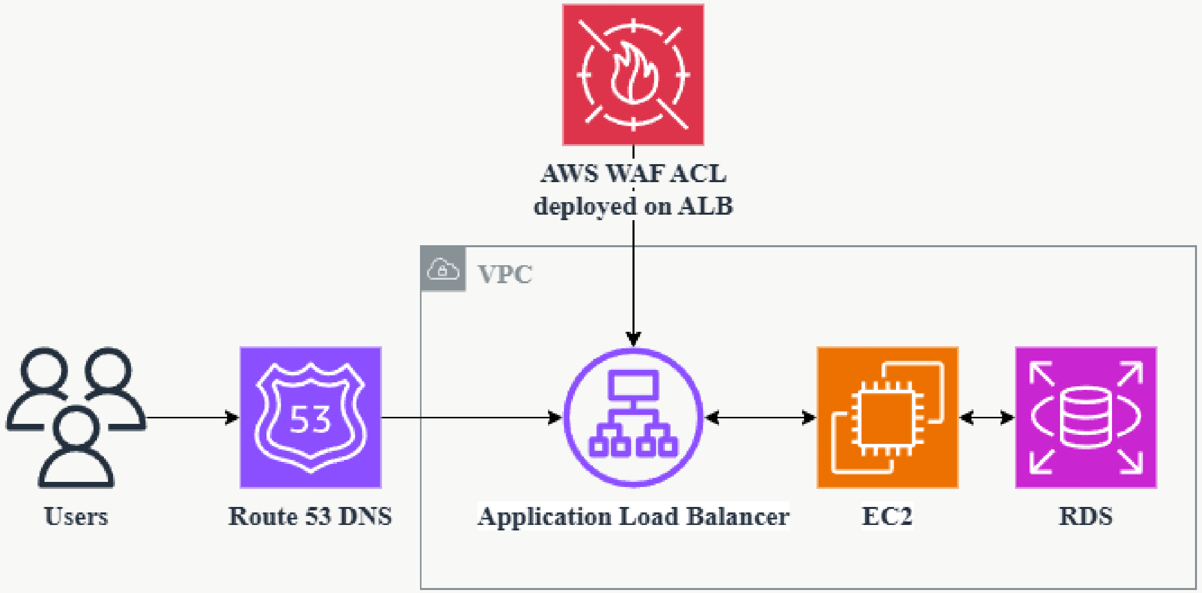
Design and implement improvement plans that fit your business needs

WAF CS 1

AWS WAF - Bot Control Case Study 1

News portal with 500k readers - challenge intro:

- High traffic peaks impacted availability
- The EC2 instance became unresponsive, requiring manual restarts.
- Already using AWS WAF with a couple of managed rules



WAF CS 1

AWS WAF - Bot Control Case Study 1

Solution:

→ High traffic impacting availability:

→ Well-Architected Review identified gaps in their solution

- Optimized current AWS WAF rules
- We implemented AWS WAF Bot Control
- Implemented Amazon CloudFront CDN for caching

→ The EC2 instance became unresponsive, requiring manual restarts:

→ Automated instance restarts to resolve EC2 unresponsiveness

WAF CS 1

AWS WAF - Bot Control Case Study 1

Solution:

→ **We implemented AWS WAF Bot Control**

What you can get in WAF Bot Control:

- **Pre-configured bot categories:** Identifies and classifies bot traffic into categories like Social Media, AI, Search Engine, SEO...
- **Unverified and verified bots**
- **Managed rule sets:** Provides ready-to-use rules to block or monitor bot traffic.
- **Comprehensive bot detection:** Uses signature-based detection, behavioral analysis, and machine learning.

WAF CS 1

AWS WAF - Bot Control Case Study 1

Solution:

→ **We implemented AWS WAF Bot Control**

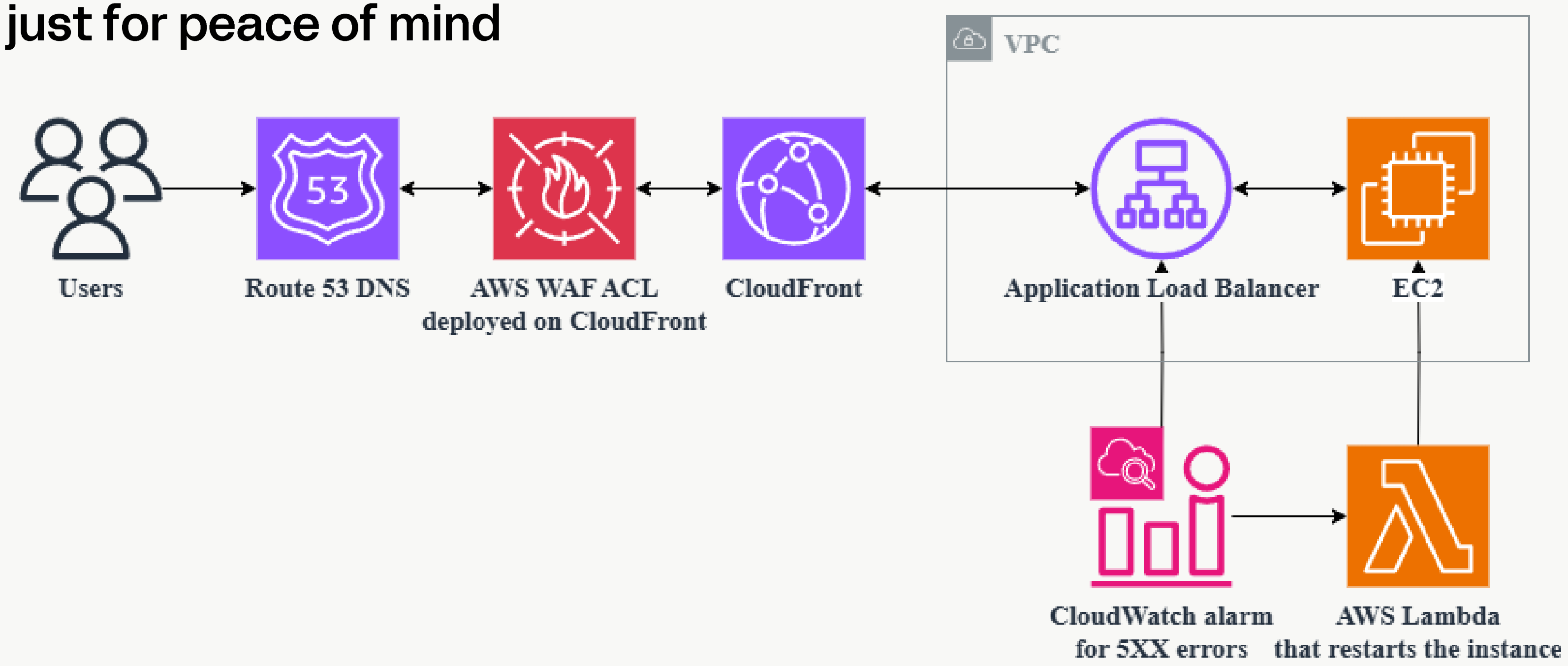
1. Implemented AWS WAF Bot Control to only check how many bots are going to the website and what they do.
2. Almost 15% of all traffic was bot traffic.
3. Majority was unverified bots getting to a specific path of the website.
4. We created a special rule that blocked this kind of traffic.

WAF CS 1

AWS WAF - Bot Control Case Study 1

Results:

- A stable website for the readers thanks to WAF Bot Control and CloudFront caching
- Has been running smoothly for a couple of years without issues
- Lambda stayed there just for peace of mind



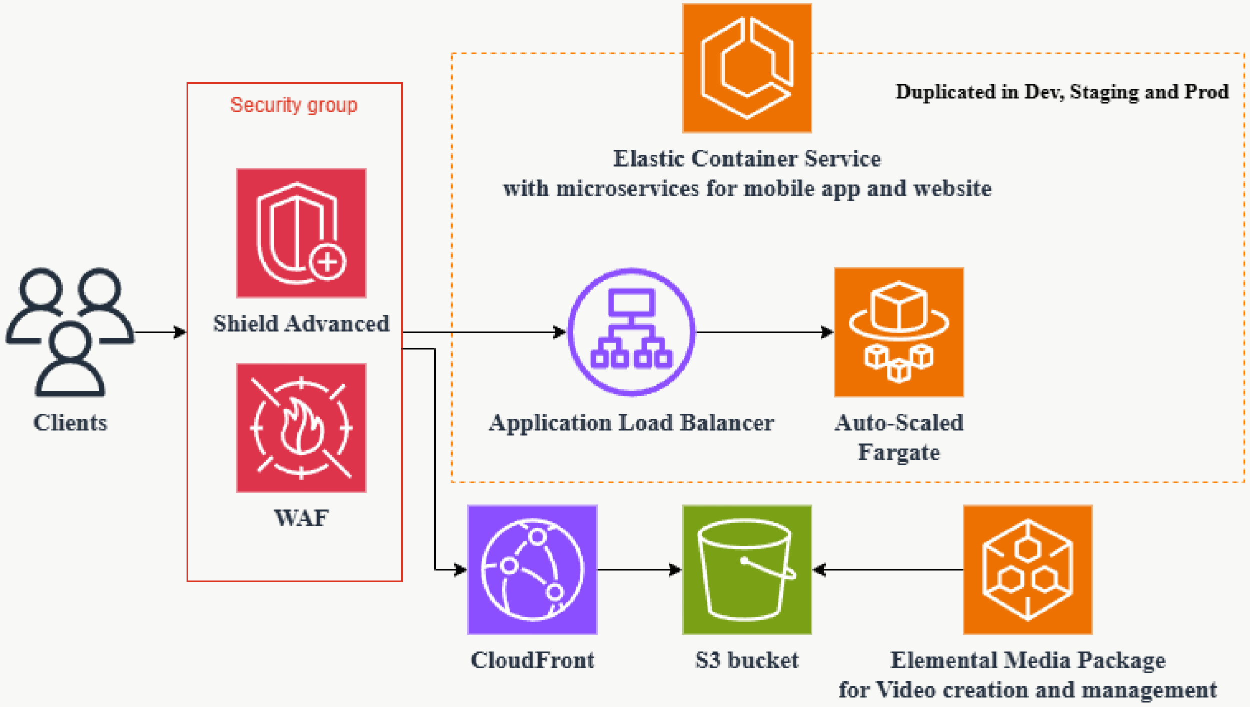
VPC
Users
Route 53 DNS
AWS WAF ACL deployed on CloudFront
CloudFront
Application Load Balancer
EC2
CloudWatch alarm for 5XX errors
AWS Lambda that restarts the instance

CS 2

AWS Shield Advanced + AWS WAF Case Study 2

News portal II with millions of readers - challenge intro:

- Past DDoS attacks impacted availability
- ECS (Elastic Container Services scalability in production and development environment)
- Already using AWS WAF with specific rules and AWS Shield Advanced for DDoS protection on multiple services, including ALB, S3 and CloudFront.



Security group
Shield Advanced
WAF
Clients
Duplicated in Dev, Staging and Prod
Elastic Container Service
with microservices for mobile app and website
Application Load Balancer
Auto-Scaled
Fargate
CloudFront
S3 bucket
Elemental Media Package
for Video creation and management

CS 2

AWS Shield Advanced + AWS WAF Case Study 2

Solution:

→ **Well-Architected Review showed us exact gaps in their solution**

→ **Past DDoS attacks impacting availability**

- **Implementing better rate limiting for certain website paths**
- **Cost savings achieved through targeted protection**

→ **ECS scalability in the production and development environment**

- **Scaled down the staging and dev environment, achieving 25% savings on the overall AWS bill**

CS 2

AWS Shield Advanced + AWS WAF Case Study 2

Solution:

→ Well-Architected Review showed us the exact gaps in their solution

→ Already using AWS WAF with specific rules and Shield Advanced for DDoS protection.

- Tested Bot Control, but without good results, only 0,5% of the whole traffic was Bots. Bot control would increase the bill by a couple of %.

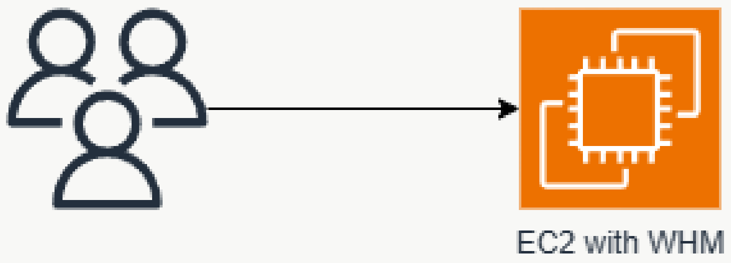
- Disabled AWS Shield Advanced protection on the CloudFront distribution with an S3 bucket behind it.

CS 3

Amazon CloudFront Case Study 3

The company taking care of 50 WP websites challenge intro:

- → Slow website loading times globally
- → All websites are hosted on a single EC2 instance.

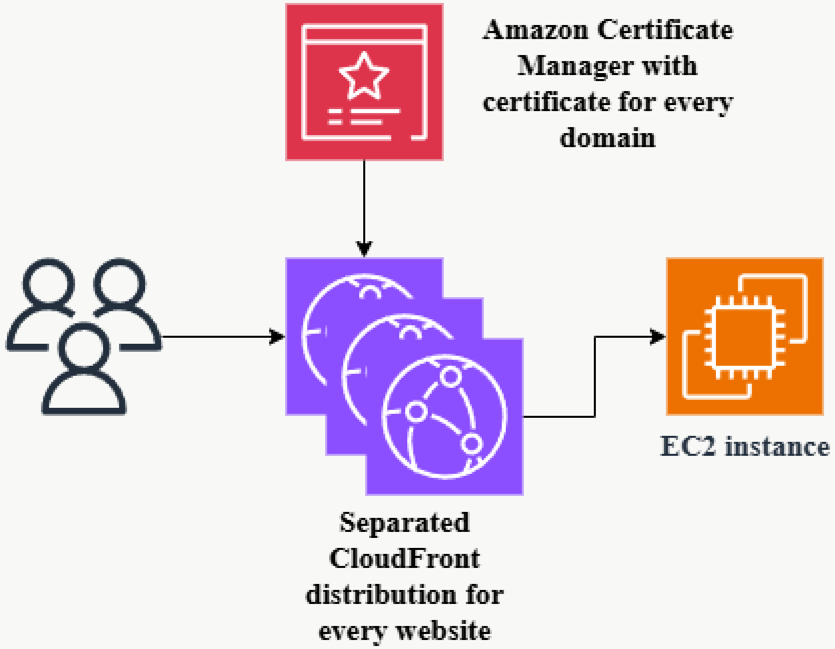


CS 3

Amazon CloudFront Case Study 3

Solution:

- → **Well-Architected Review showed us the exact gaps in their solution**
  - → Slow loading of the websites
    - We implemented a separate Amazon CloudFront distribution for every website
    - Required a special setup for each website
    - Deployed through Infrastructure as Code


Amazon Certificate Manager with certificate for every domain
Separated CloudFront distribution for every website
EC2 instance

CS 3

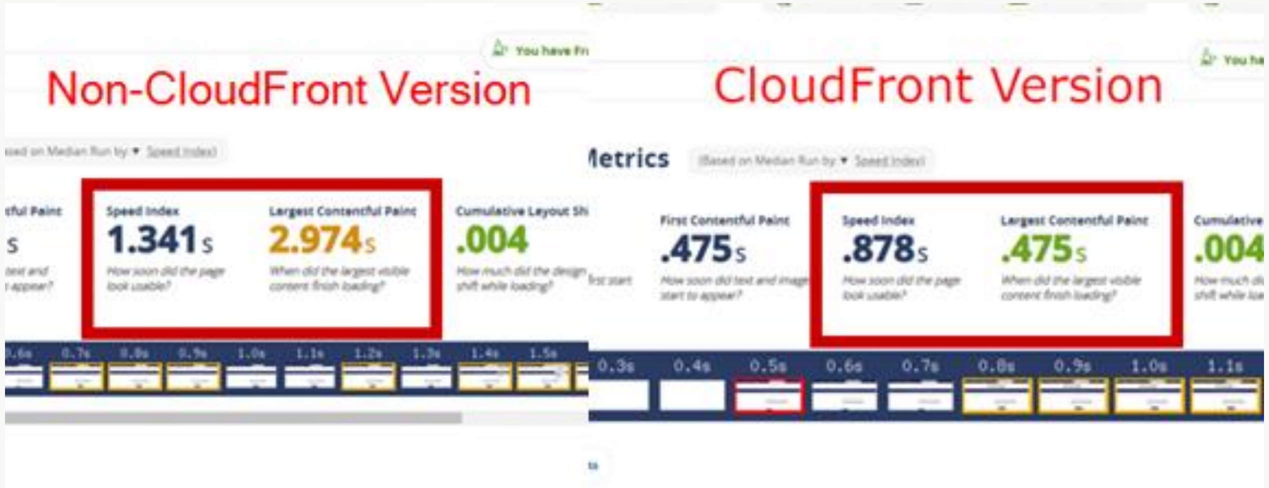
Amazon CloudFront Case Study 3

Results:

→ **Faster loading of the websites**

- Visible change in loading times, thanks to CloudFront pricing, even lower bill
- EC2 instance stayed the same

50% faster



Non-CloudFront Version

sed on Median Run by: ▾ Speed Index)

ntful Paint
s
text and
appear?

Speed Index
1.341s
How soon did the page look usable?

Largest Contentful Paint
2.974s
When did the largest visible content finish loading?

Cumulative Layout Sh
.004
How much did the design shift while loading?

0.6s 0.7s 0.8s 0.9s 1.0s 1.1s 1.2s 1.3s 1.4s 1.5s

CloudFront Version

Metrics (Based on Median Run by: ▾ Speed Index)

First Contentful Paint
.475s
How soon did text and images start to appear?

Speed Index
.878s
How soon did the page look usable?

Largest Contentful Paint
.475s
When did the largest visible content finish loading?

Cumulative
.004
How much di
shift while loa

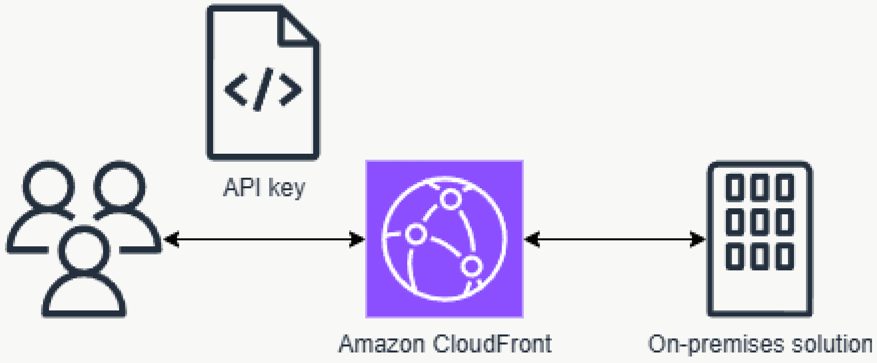
0.3s 0.4s 0.5s 0.6s 0.7s 0.8s 0.9s 1.0s 1.1s

CS 4

Edge functions - Lambda@Edge Case Study 4

The company serves millions of users with map files challenge intro:

- New CloudFront deployment for global accessibility and better latency
- Need a security feature to only allow users with special API KEY
- Using outdated solutions to connect to this service, with no ability to implement alternative authentication methods beyond a simple API key in the URL.



Edge functions - Lambda@Edge Case Study 4

CS 4

Solutions:

→ **Need a security feature to only allow users with special API KEY**

- example.com/file1.gzip?apikey=xzyxzyxzy
- Not possible to cache based on the key
- Sending the key to the origin for validation on every request would prevent effective caching
- We have to check the key in CloudFront, using Lambda@Edge

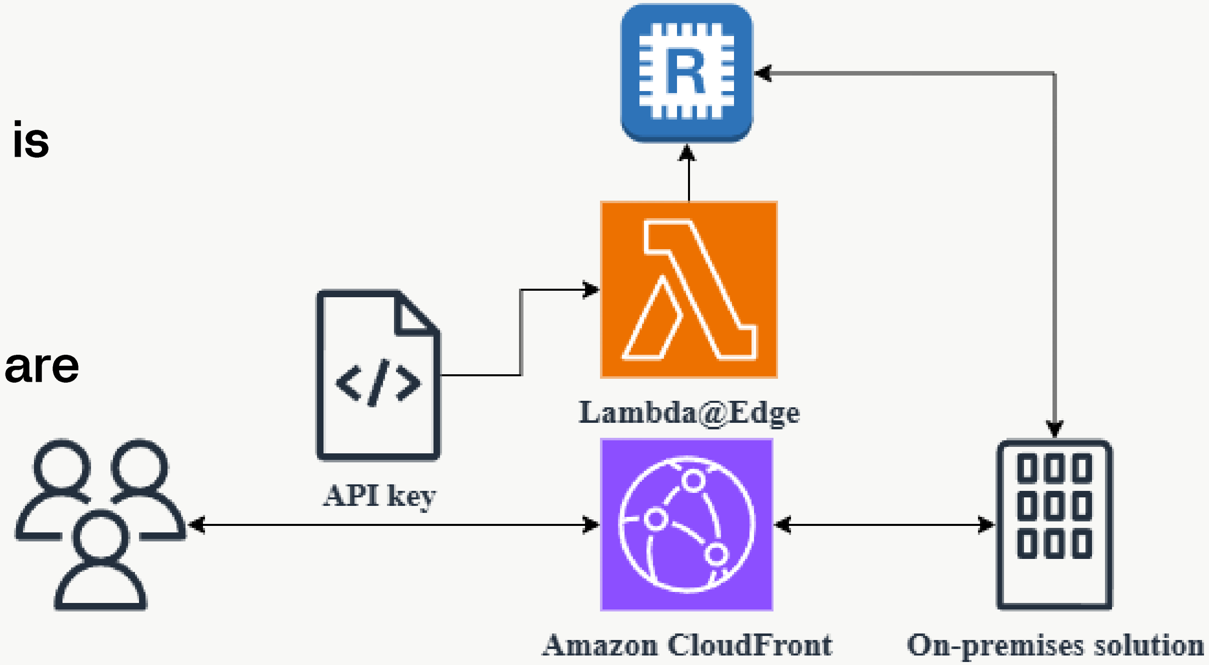
CS 4

Edge functions - Lambda@Edge Case Study 4

Results:

→ **API key validation with Lambda@Edge:**

- If the key exists in the database, access is granted.
- All validation occurs at the edge.
- If the files are already in the cache, they are served from the CloudFront cache



Conclusion

→ Investing time in implementing AWS Edge Services delivers significant benefits without added overhead.

→ All services are fully managed.

→ You don’t have to worry about downtime caused by faulty updates.

stormit.cloud

Contact us

stormit.cloud
info@stormit.cloud