음? 수호님 저희 서비스 403 떨어지는데요?

"우아한형제들 AWS WAF 운영 이야기"

우아콘 WOOWA CON 2021

송수호
정보보호실

시작하며

AWS WAF는

- AWS Cloud 에서 제공하는 웹 어플리케이션 방화벽
  - 어플리케이션 단계의 웹 공격으로부터 보호하는 역할
  - 웹 트래픽에 대한 가시성
  - 보안 정책에 따라 비인가 요청 탐지/차단
  - 지속적인 기능 추가에 맞춰 여러 역할을 제공 중

WAF Classic

WAFv2

- 버전 별 구분
  - 이전 버전 AWS WAF → WAF Classic
  - 신규 버전 AWS WAF → WAFv2

시작하며

우아한형제들과 AWS WAF

우아한형제들에서는 두가지 버전의 WAF를 사용하고 있었습니다

WAF Classic

- WAFv2 출시 이전에 연동된 Public 서비스들을 보호
- Custom rules + Managed rules(Marketplace) 조합

WAFv2

- WAFv2 출시 이후에 생성된 Public 서비스들을 연동하여 보호
- Custom rules + AWS Managed rules + Marketplace Managed rules 조합

시간이 지날수록

- 두 가지 버전을 모두 운영하는 상황으로 이어지고, 관리 리소스도 자연스럽게 증가
- 시간이 흐를수록 WAFv2에만 기능적/관리적 편의성에 도움을 주는 쪽으로 발전 → 마이그레이션 계획

시작하며

출시 직후 WAFv2로 이전하지 못한 이유?

- 여러 보안 환경을 담당하는 관리자의 리소스 문제
  - 서비스 별 특성을 고려한 WAF 재 적용 과정 필요
  - 서비스 성장 속도 만큼이나 새로 보호해야 할 대상 증가
- 신규 보안 정책이나 기능에 따른 오탐(False Positive) 염려
  - 오탐 등으로 인한 서비스 문제 → 장애 발생은 무섭다
  - WAF Classic에서 보호중인 서비스 → 일정기간 Count로 룰 조정
- 저렴한 보안 서비스로 다양한 효과를 얻을 수 있었음
  - 지속적인 신규 기능 확인 → WAF Classic의 기능적 한계 돌파구
  - 다양한 테스트 진행 → 탐지/차단 외 운영적인 측면 고려

→ 마이그레이션!

마이그레이션 계기

마이그레이션 계기

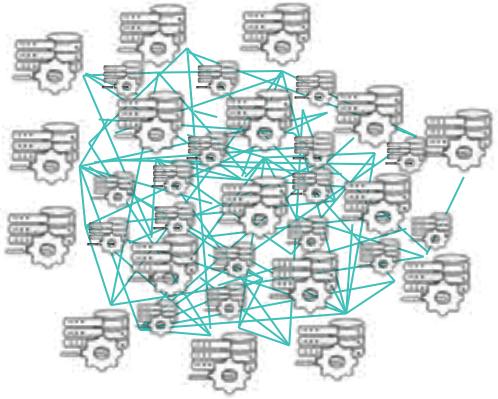
WAF Classic에서 겪은 어려움

- Microservice Architecture(MSA)
  - 우아한형제들 서비스의 수많은 환경들은 서로간의 특성 또한 너무나도 다양

2016

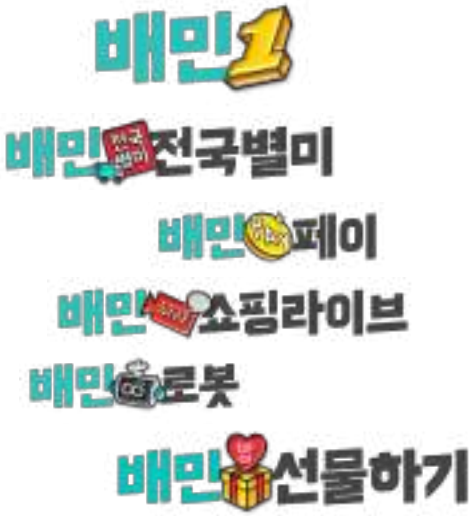
10+ Service

2020



600+ Service

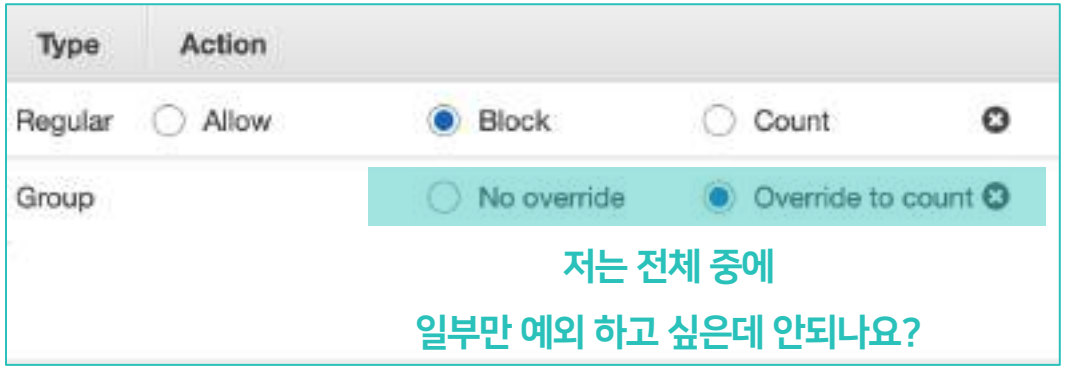
2021



마이그레이션 계기

WAF Classic에서 겪은 어려움

- 서비스 별 세심한 모니터링
  - 오탐에 의한 서비스 장애가 발생하지 않아야 함
  - 하지만, WAF Classic의 예외처리 기능이 약했음



앞의 룰 에서

A서비스는

Header “example_data” 포함 AND “/order”에 요청시 Allow

B서비스는

Src IP “x.x.x.x”이면서 “/shop_list”에 요청시 Allow

  - WebACL과 ALB가 1:1 또는 1:3 이하로 구성되는 경우 증가
  - 신규 서비스 → 리소스 증가 / 관리대상 증가 → Rate exceeded 발생
  - 연쇄적인 어려움은 고통으로 되돌아옴

마이그레이션 계기

AWS WAFv2 출시

- 신규 버전에서 본 희망

기능	WAFv2	WAF Classic
AWS 관리 규칙 그룹	신규 등장	-
WebACL 룰 적용 제한	WebACL 별 Capacity 내에서 관리	10개
WebACL 룰 그룹 적용 제한		2개
Managed rule 예외 처리	Rule 별 상세 예외처리 지원	Count or Block

출처: https://aws.amazon.com/waf

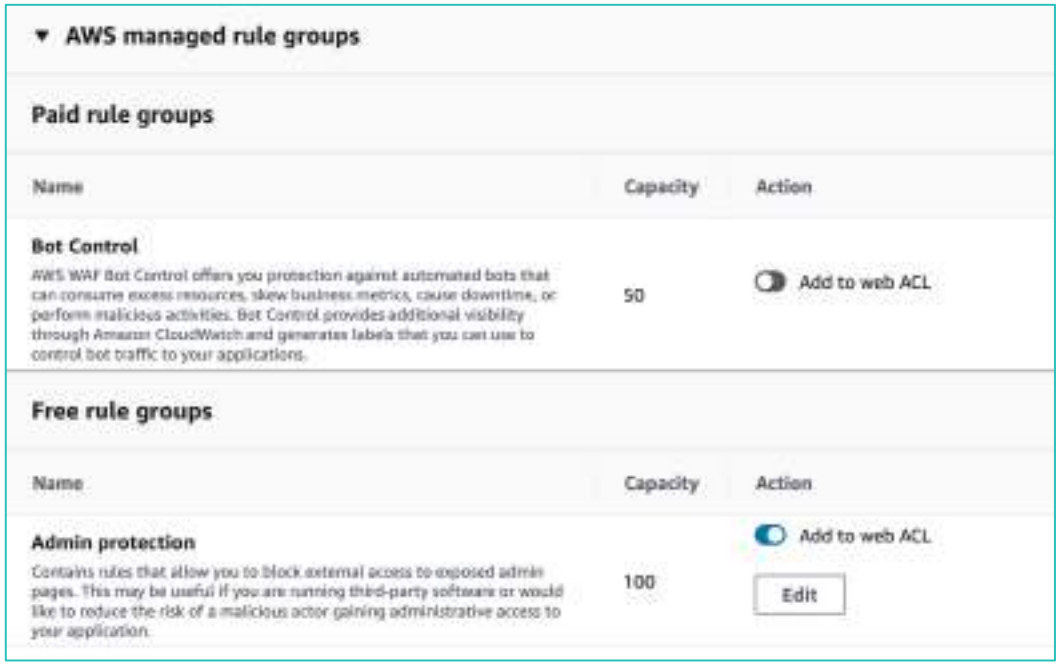
- AWS Managed rules로 규칙 생성/관리 부담 감소
- Managed rule 상세 예외처리 지원
- 운영에 큰 도움되는 기능들의 추가 → Logging configuration, API 등
- 계속되는 업데이트

대표적인 기능 변화

대표적인 기능 변화

Managed Rule

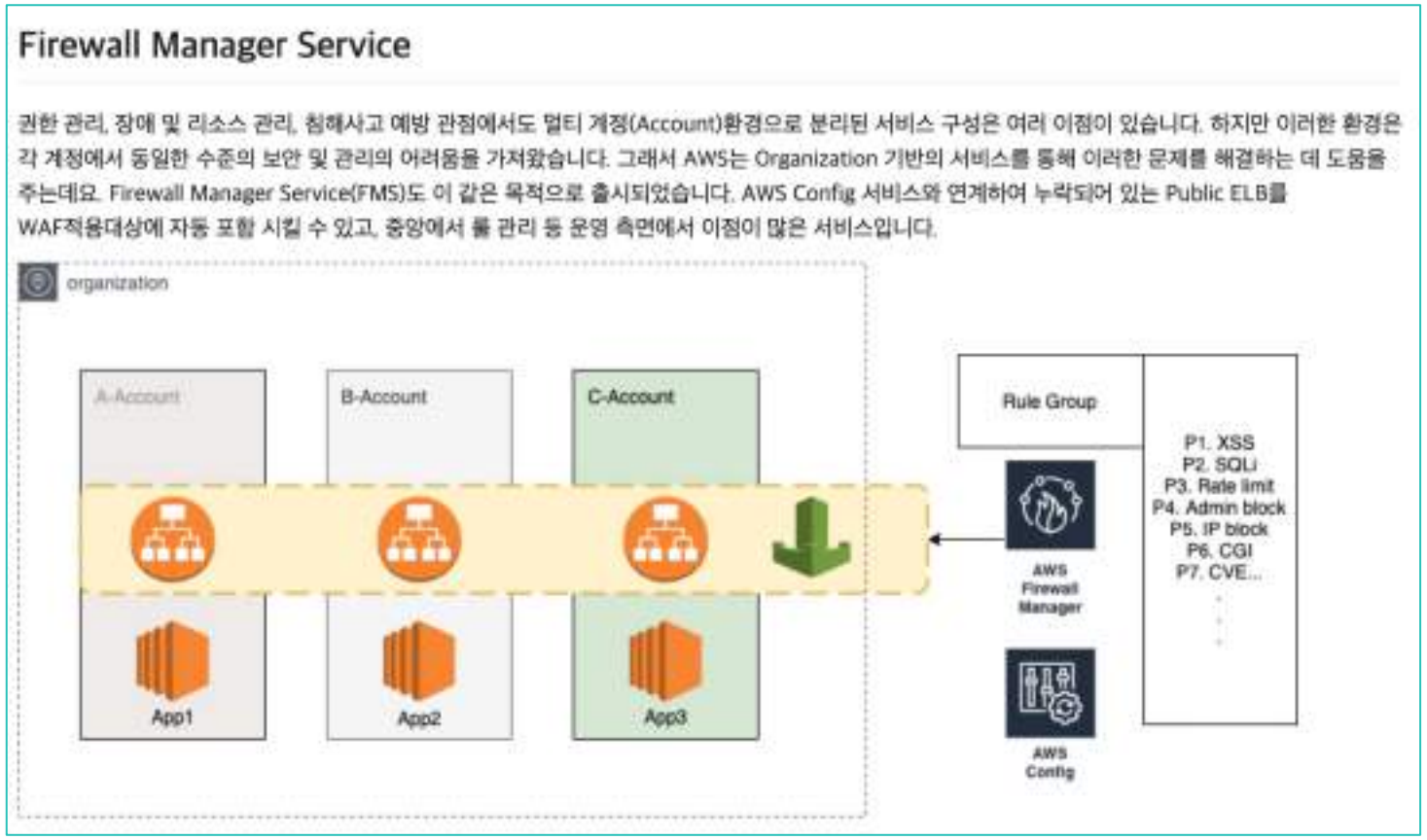
- AWS Managed rule groups 제공
  - AWS Threat research team
  - 알려진 유형의 웹 공격을 방어
  - 신규 위협 확인 시 업데이트 제공



- WAF Classic과 차이?
  - Classic에서는 구독형 Managed rule 만 제공
  - 다양한 보안 rule 적용 필요 = 여러 개의 Rule을 구독? → 비용 측면, 예외처리 걱정
  - 신규 버전의 등장으로 "비용 문제, Custom rule 추가/관리 부담 감소"
  - 동시에 Rule 선택의 폭이 넓어짐

대표적인 기능 변화

Managed Rule



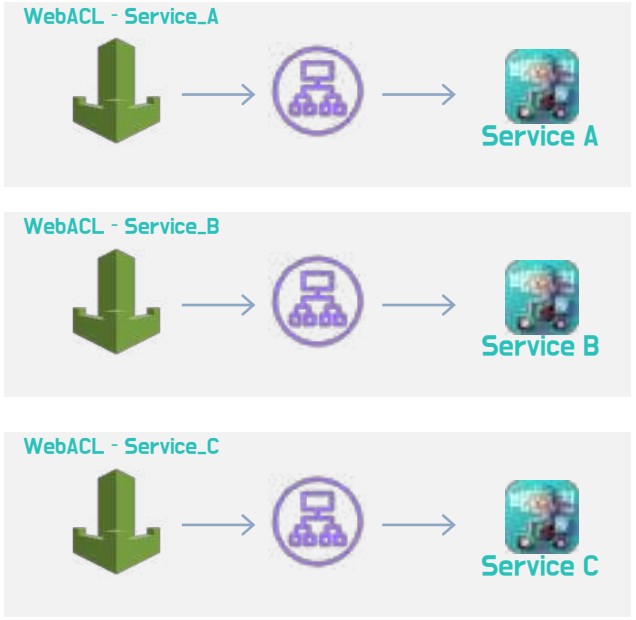
- 보호대상을 일괄적으로 동일한 수준의 환경으로 보호할 수 있다면 Best!
- 서비스 특성, 개발 언어, 환경 등 여러가지 상황으로 일괄적인 보호 환경을 제공하는 것은 어렵다

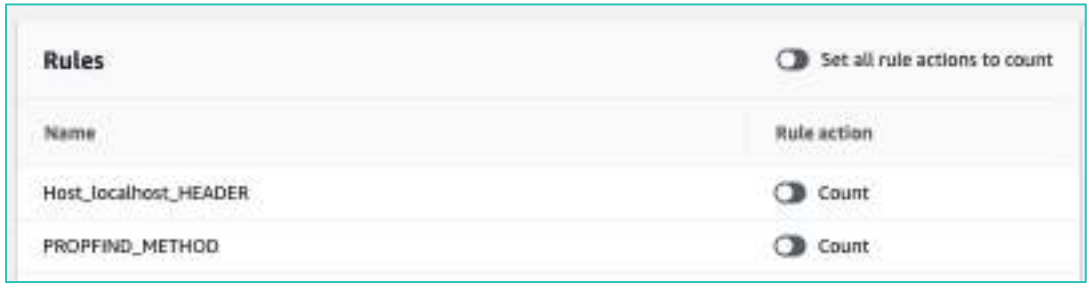
대표적인 기능 변화

Managed Rule

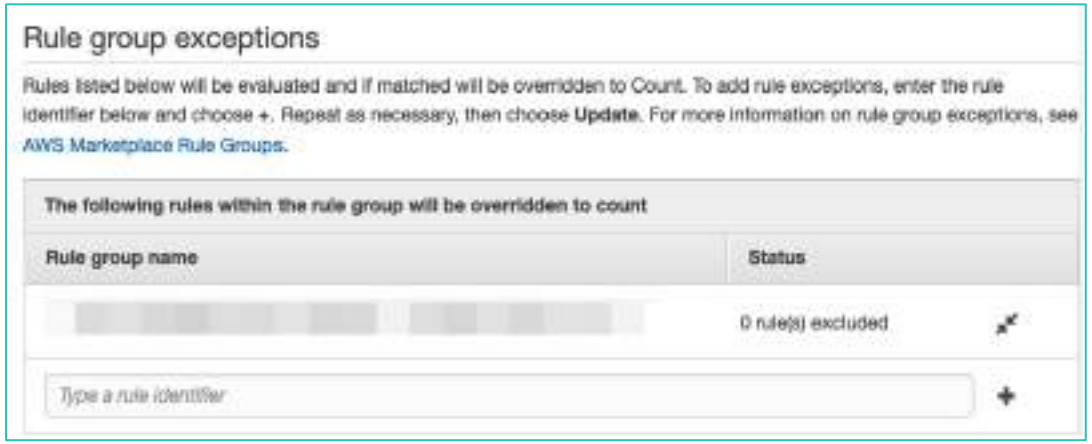
- 유연한 예외 처리 지원

Type	Action		
Regular	Allow	Block	Count
Group		No override	Override to count





WAFv2

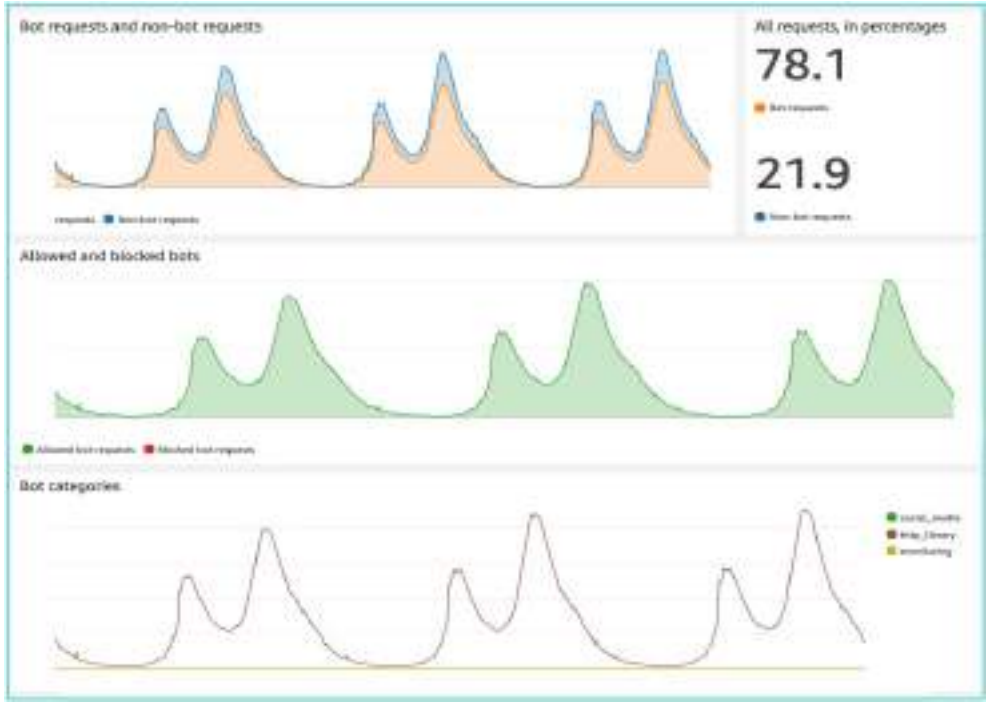


WAF Classic

대표적인 기능 변화

Managed Rule

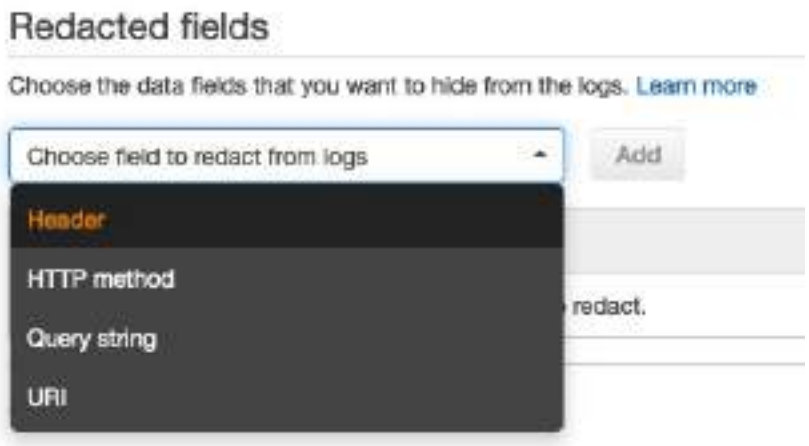
- Bot Control
  - 21년 4월 봇 성격의 요청 트래픽(스크래퍼, 스캐너, 크롤러 등) 에 대한 가시성과 대응을 위한 기능
  - 구독형 Managed Rule에서도 일부 제공 → 요청의 Signature, IP 기반 탐지
  - 봇 행동의 범주를 기준으로 탐지 결과 제공(소셜 미디어, HTTP Library등)



대표적인 기능 변화

로깅 상세 설정 지원

- 변경된 로그 설정 지원

WAF Classic	WAFv2
	

어떻게 쓰고 있을까

어떻게 쓰고 있을까

그룹화된 WebACL 운영

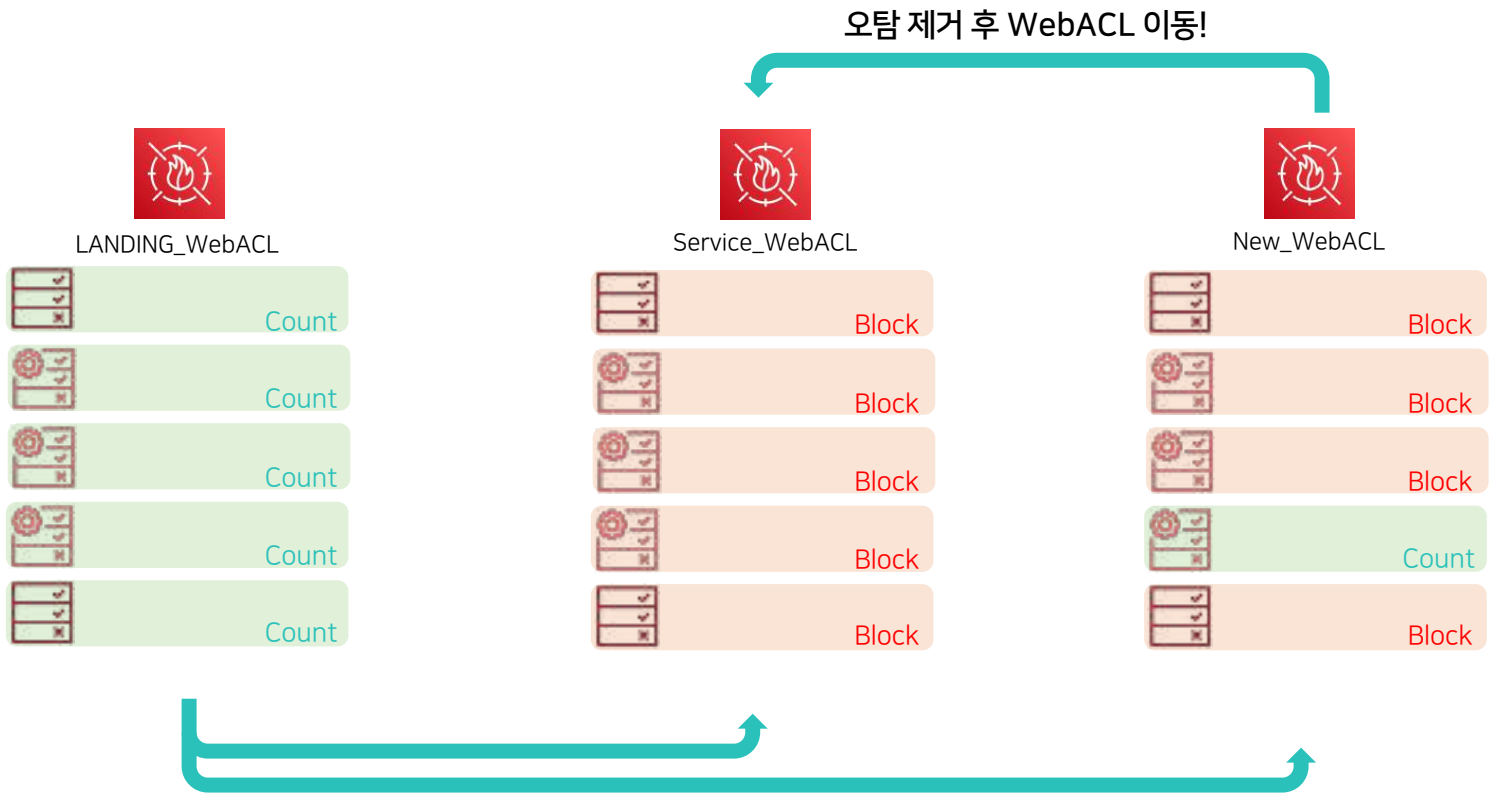
- 서비스 목적 별로 WebACL 운영
  - 예외 처리가 필요한 환경만 분리

WebACL 개수
50%
이상 감소

어떻게 쓰고 있을까

Landing WebACL

- WAF 정류장



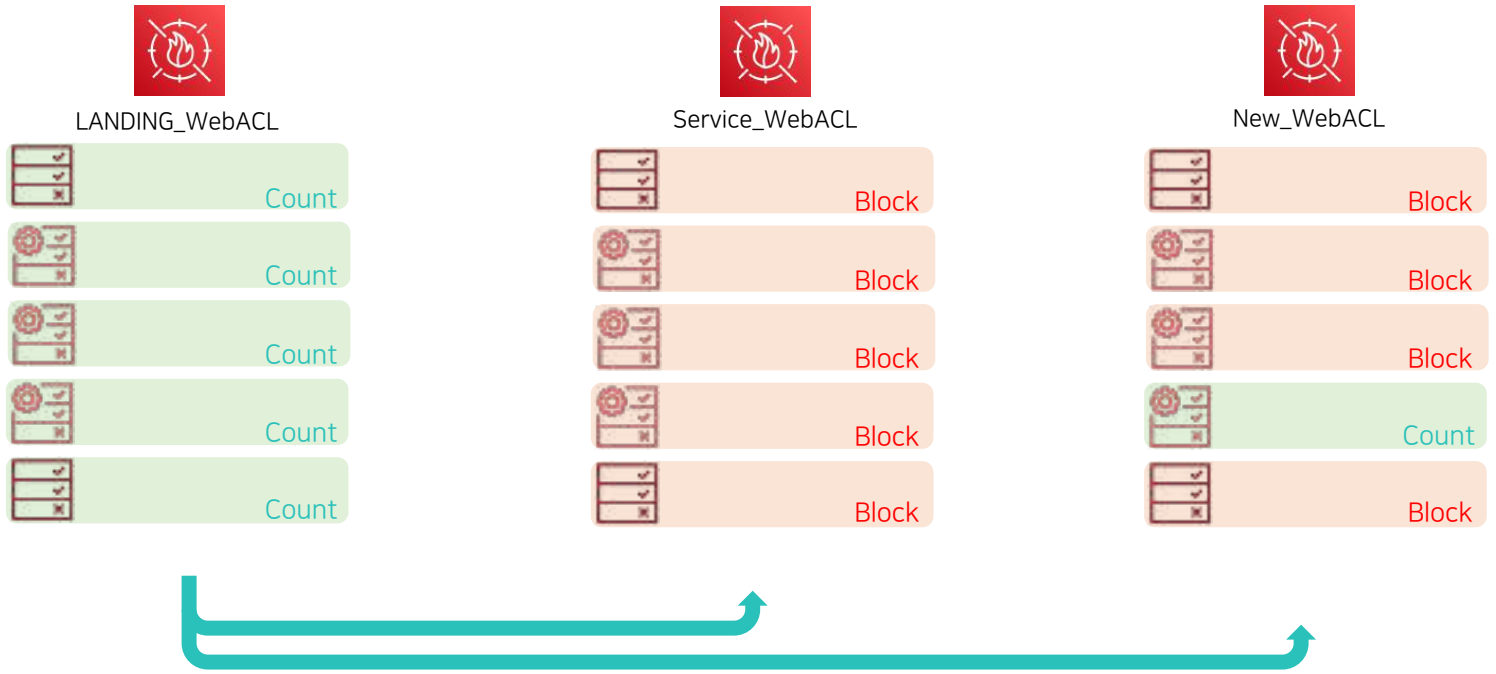
1. 수정이 근 시일내 가능한 경우

→ 수정 후 모니터링하여 다시 연동

어떻게 쓰고 있을까

Landing WebACL

- WAF 정류장



2. 수정이 근 시일내 어려운 경우

→ Count 룰 모니터링 방안 고려

어떻게 쓰고 있을까

Landing WebACL

- WAF 정거장

LANDING_WebACL

Count
Count
Count
Count
Count

Service_WebACL

Block
Block
Block
Block
Block

OR

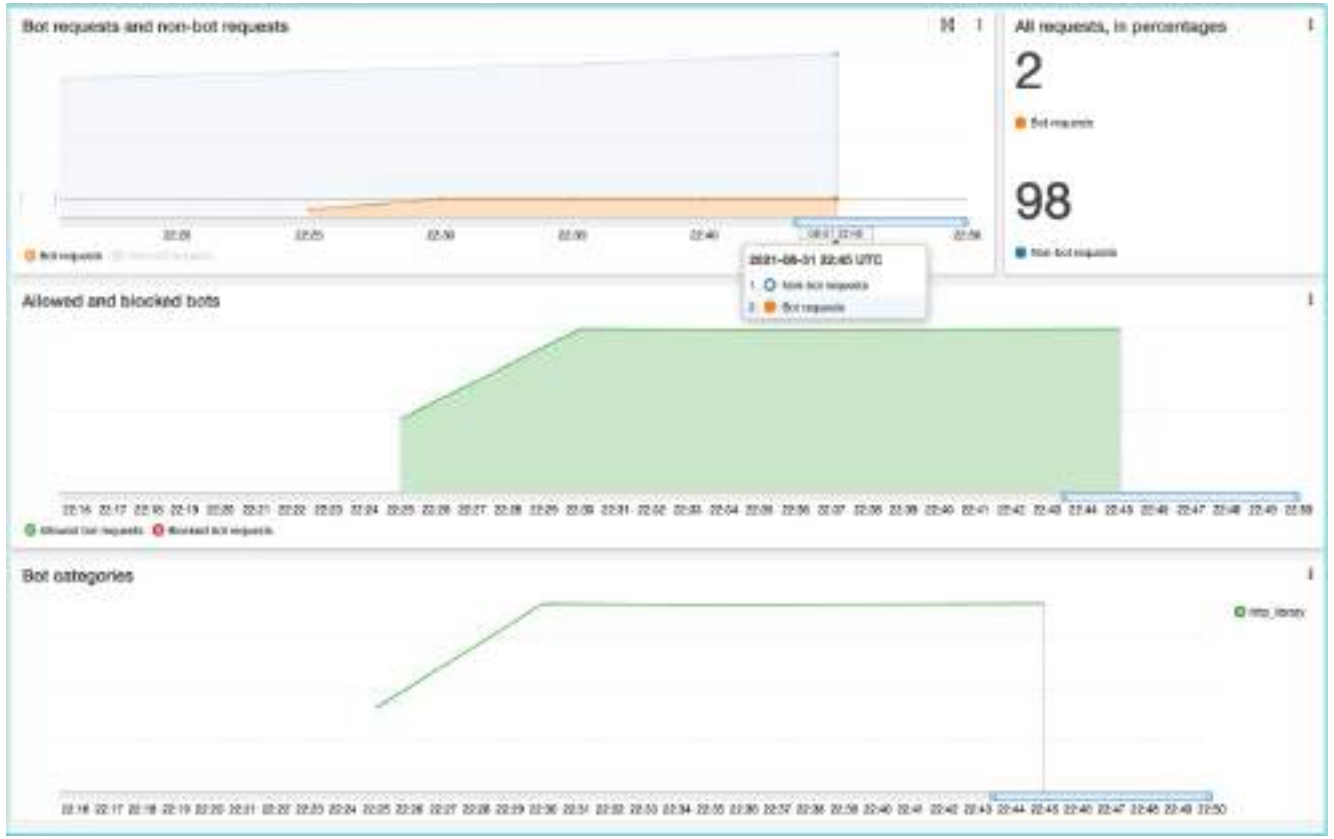
New_WebACL

Block
Block
Block
Block
Count
Block

어떻게 쓰고 있을까

어뷰징 탐지/대응 사례

- Bot Control



- 서비스 대상 어뷰징 트래픽 추적
- 자동화된 접근을 추적하기 위함
- 약 2%의 어뷰징 트래픽 확인

어떻게 쓰고 있을까

어뷰징 탐지/대응 사례

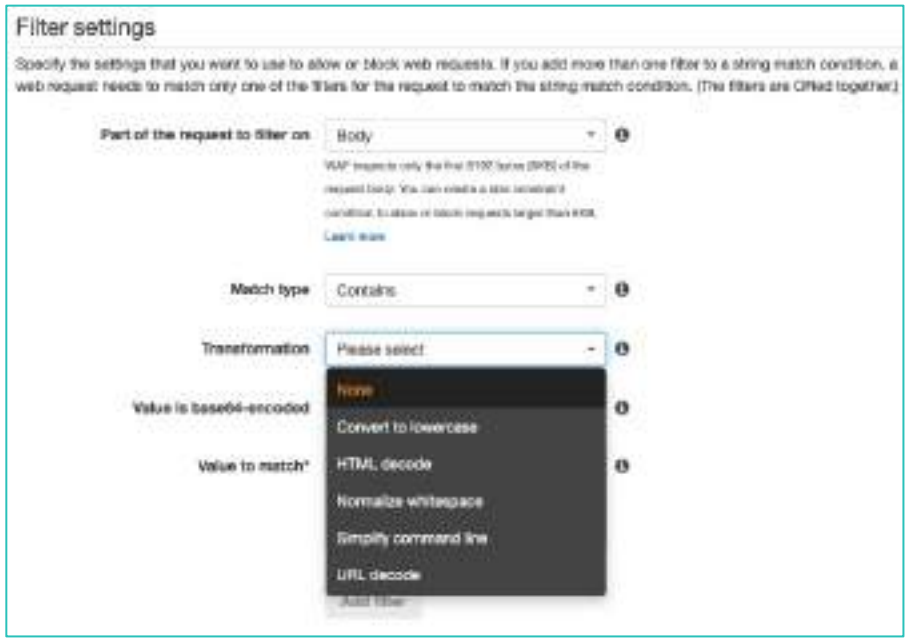
- Bot Control

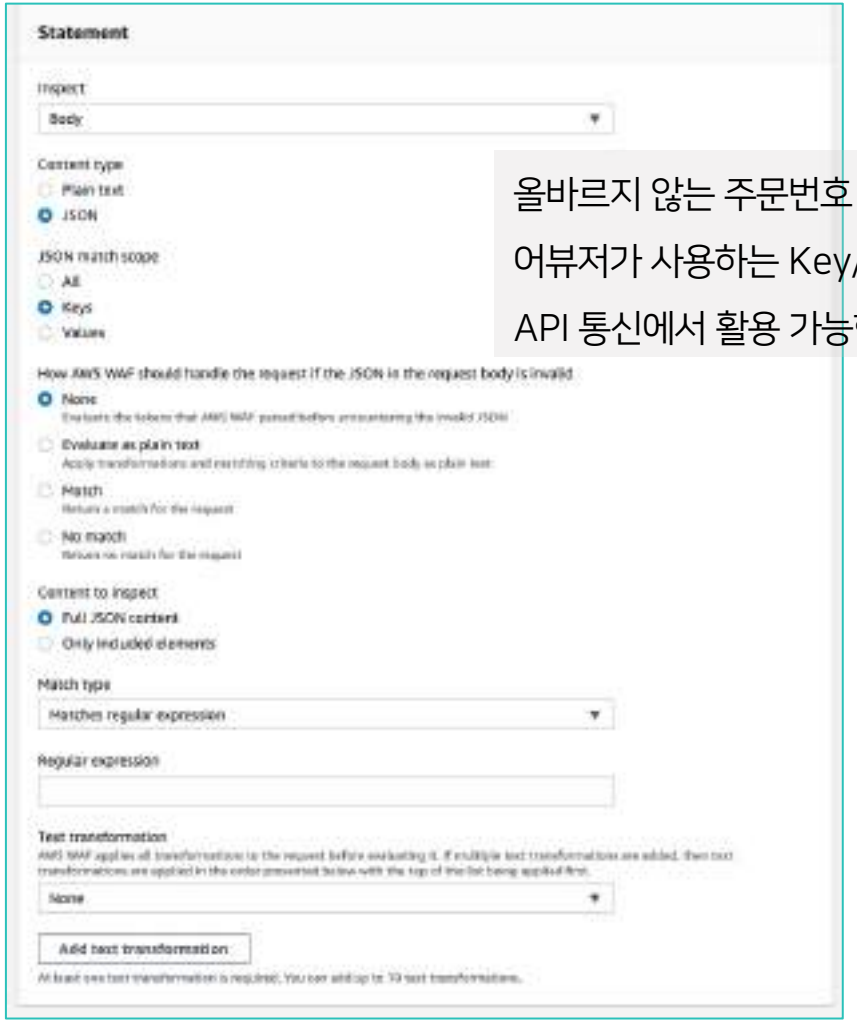
- 예외 처리에 대한 고민
- 이전과 같이 서비스 별 특성을 고려하며 순차적 적용

어떻게 쓰고 있을까

어뷰징 탐지/대응 사례

- Body JSON Match





올바르지 않는 주문번호 형식

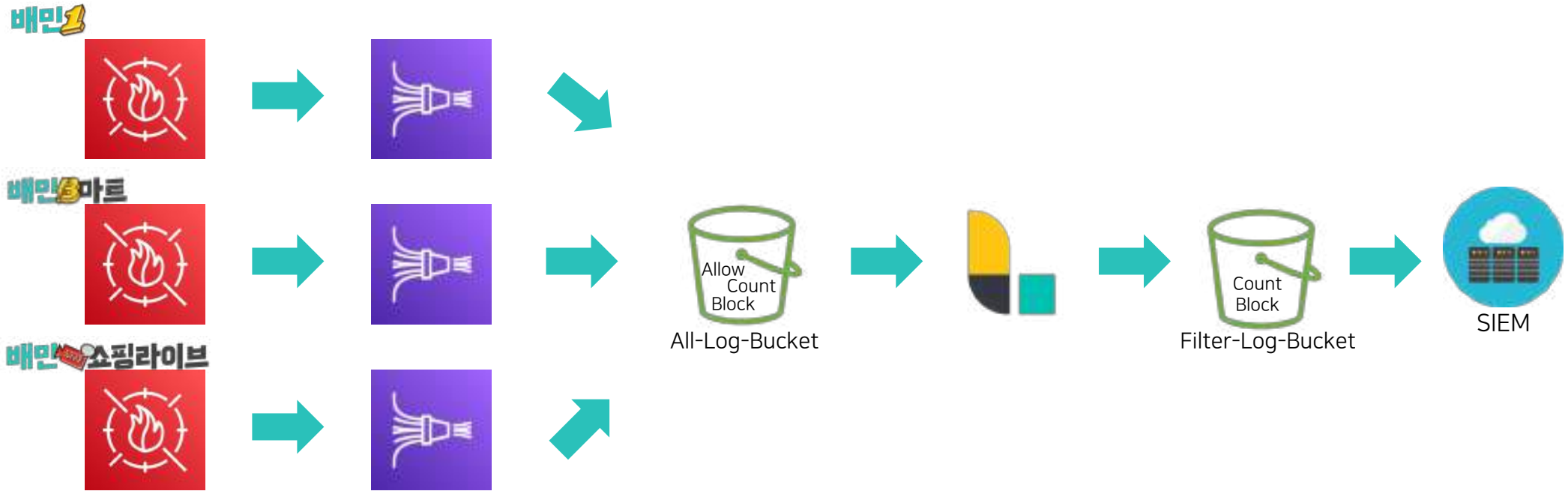
어뷰저가 사용하는 Key/Values 값 모니터링

API 통신에서 활용 가능한 방법이 다양

어떻게 쓰고 있을까

로깅 설정 변경 사례

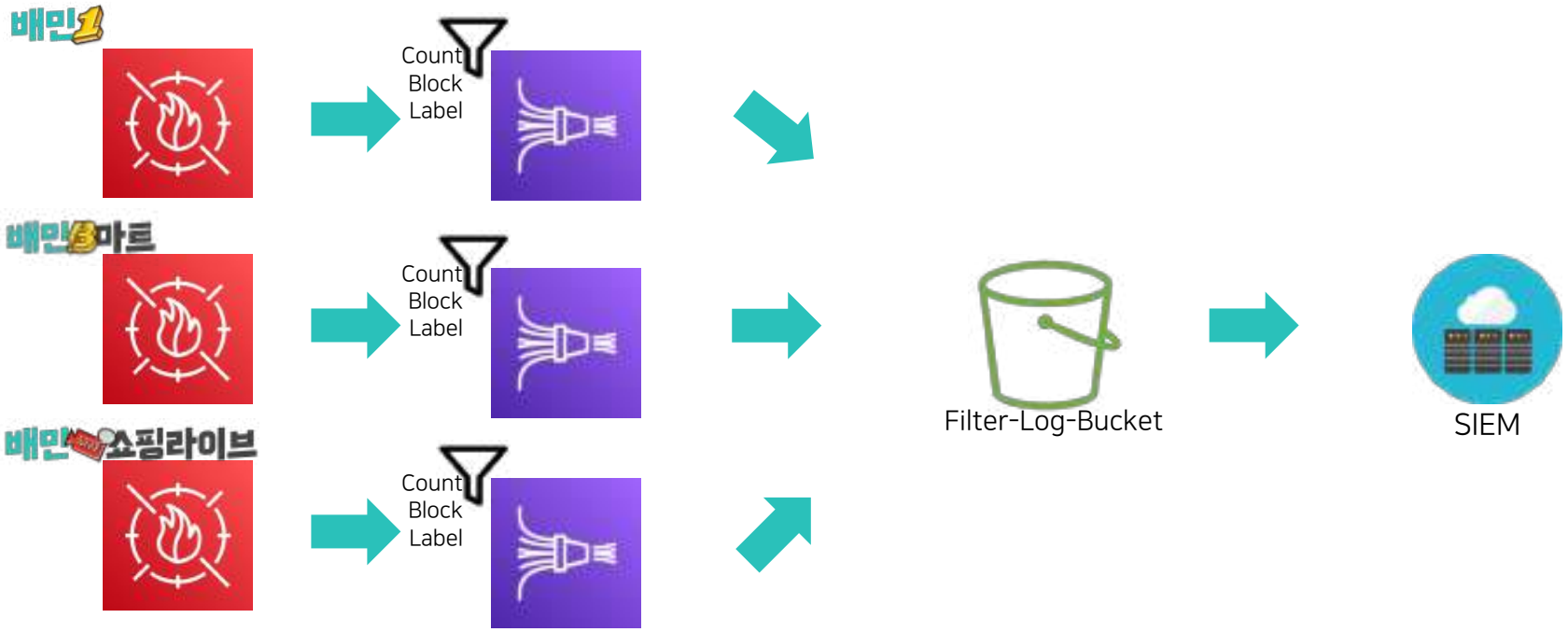
- 우아한형제들 로그 수집 환경
  - WAF Classic 에서는 Logstash를 이용하여 로그를 선별



어떻게 쓰고 있을까

로깅 설정 변경 사례

- 우아한형제들 로그 수집 환경
  - WAFv2 에서는 Filter Log 기능을 이용하여 로그를 저장



어떻게 쓰고 있을까

로깅 설정 변경 사례

- 우아한형제들 로그 수집 환경

배달의민족 주문 관련 서비스

2020년 9월

```
2020-10-01 09:04:17  393.5 KiB
2020-10-01 09:04:49  412.4 KiB
2020-10-01 09:04:54  431.1 KiB
2020-10-01 09:04:58  427.3 KiB
2020-10-01 09:05:01  419.2 KiB

Total Objects: 137480
   Total Size: 592.4 GiB
```

2021년 9월

```
2021-10-01 08:50:37  774 Bytes
2021-10-01 08:50:44  814 Bytes
2021-10-01 08:57:58  693 Bytes
2021-10-01 08:58:03  653 Bytes
2021-10-01 09:02:58  825 Bytes

Total Objects: 296572
   Total Size: 11.3 GiB
```

배달의민족 상품 관련 서비스

2020년 9월

```
2020-10-01 08:59:27   32.3 KiB
2020-10-01 09:00:43   33.3 KiB
2020-10-01 09:03:43   44.1 KiB
2020-10-01 09:04:01   39.5 KiB
2020-10-01 09:04:28   36.9 KiB

Total Objects: 18619
   Total Size: 26.6 GiB
```

2021년 9월

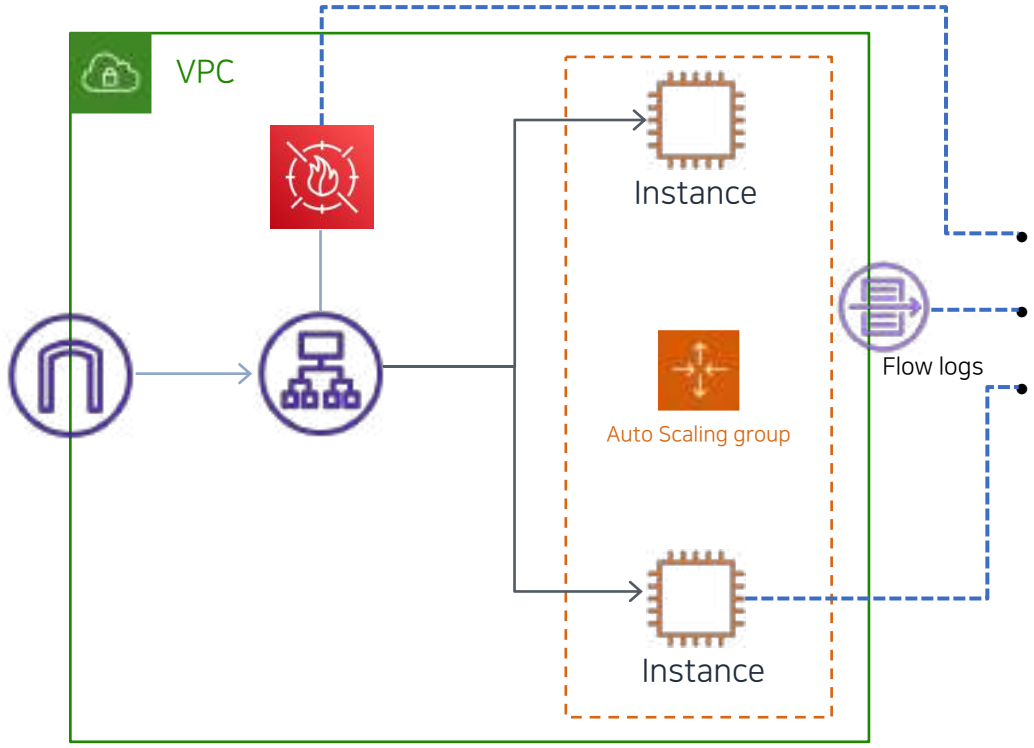
```
2021-10-01 08:50:26  533 Bytes
2021-10-01 08:53:10  502 Bytes
2021-10-01 08:53:13  745 Bytes
2021-10-01 09:00:35  677 Bytes
2021-10-01 09:00:45  497 Bytes

Total Objects: 8101
   Total Size: 5.9 MiB
```

어떻게 쓰고 있을까

로깅 설정 변경 사례

- 전체적인 로그 관리 관점에서 보았을 때



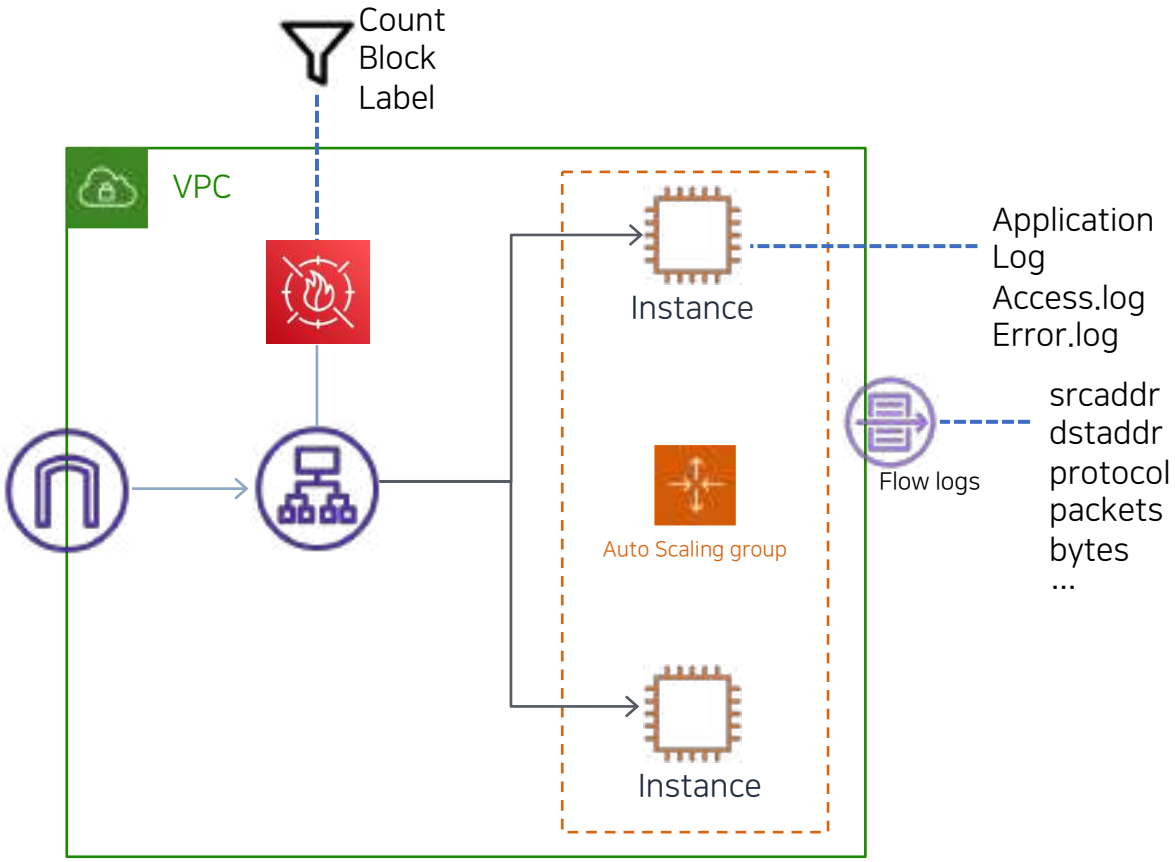
- 모든 로그에 대해 저장 (Allow, Count, Block 등)
- VPC Flow Logs로 송수신 트래픽 로깅
- Endpoint에서 서비스 로그 전체 저장(Access, Error)

어떻게 쓰고 있을까

로깅 설정 변경 사례

- 침해사고 분석 관점에서 보았을 때
  - 더 나은 로깅 설정이 무엇일지 고민
    - 사고 분석에서는 다양한 관점에서의 로그가 필요함
    - 어디에서 어떤 로그를 남길 것인가 고민 필요
    - AWS WAF 로그보다 더 상세한 로깅

      → 선택적으로 활용하여 보강



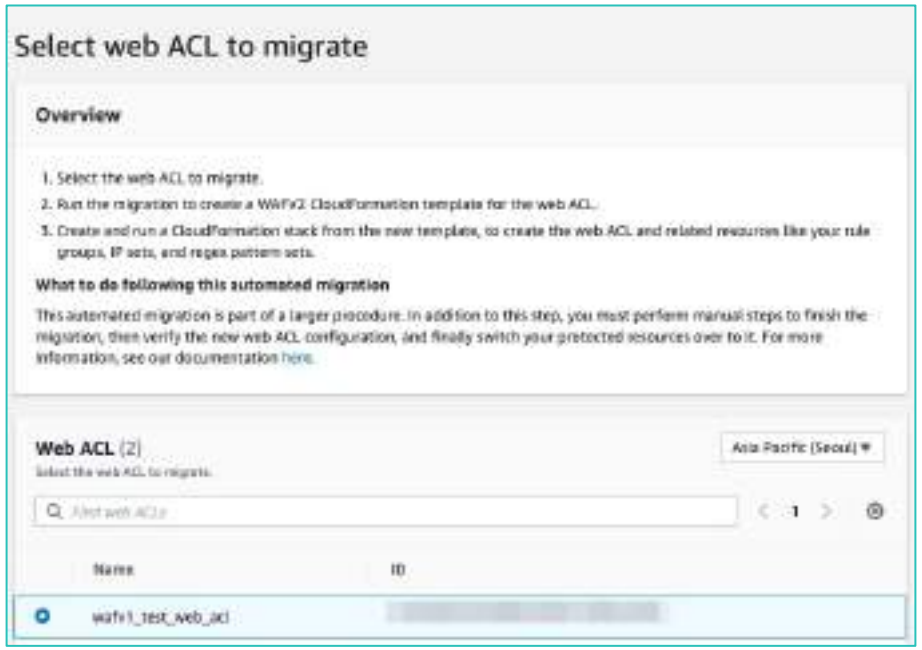
마이그레이션 참고 사항

마이그레이션 참고사항

Migration wizard

- 이전 대상을 편리하게 옮겨보자

WAF Classic

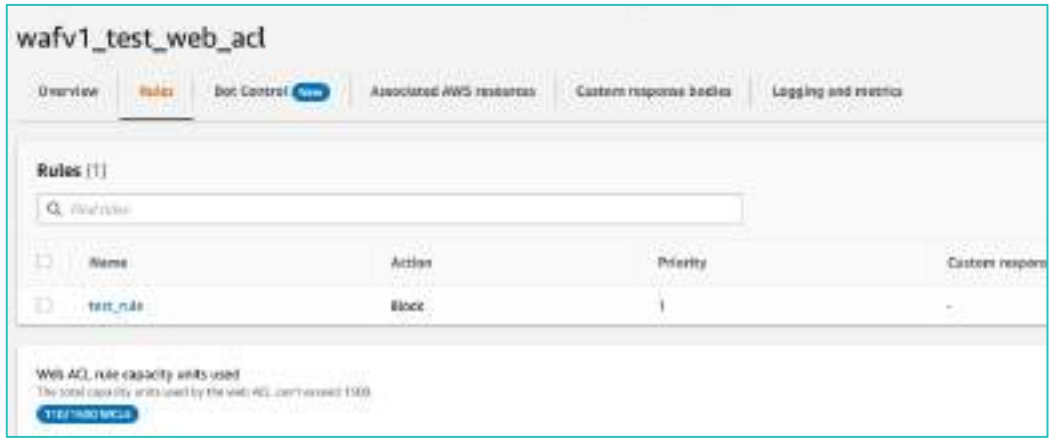


WAFv2

마이그레이션 참고사항

Migration wizard

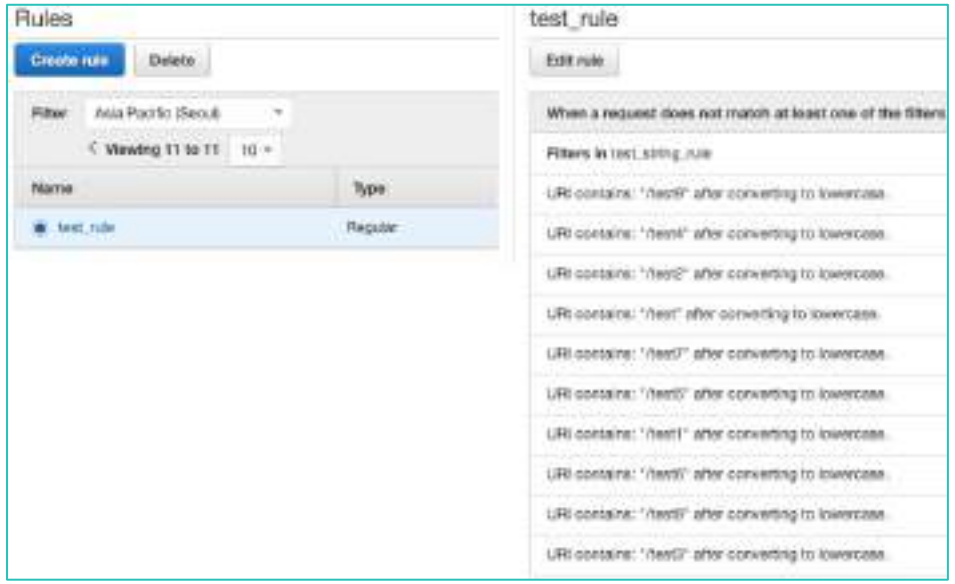
- 이전 대상을 편리하게 옮겨보자

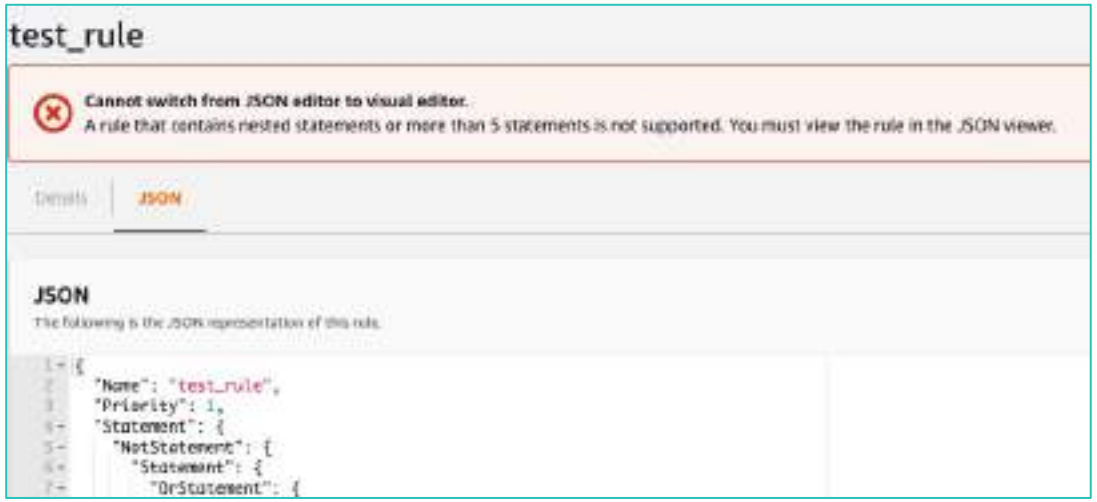


마이그레이션 참고사항

Migration wizard

- 이전 대상을 편리하게 옮겨보자
  - 변화된 정책에 대해서 충분한 검토와 테스트가 필요함



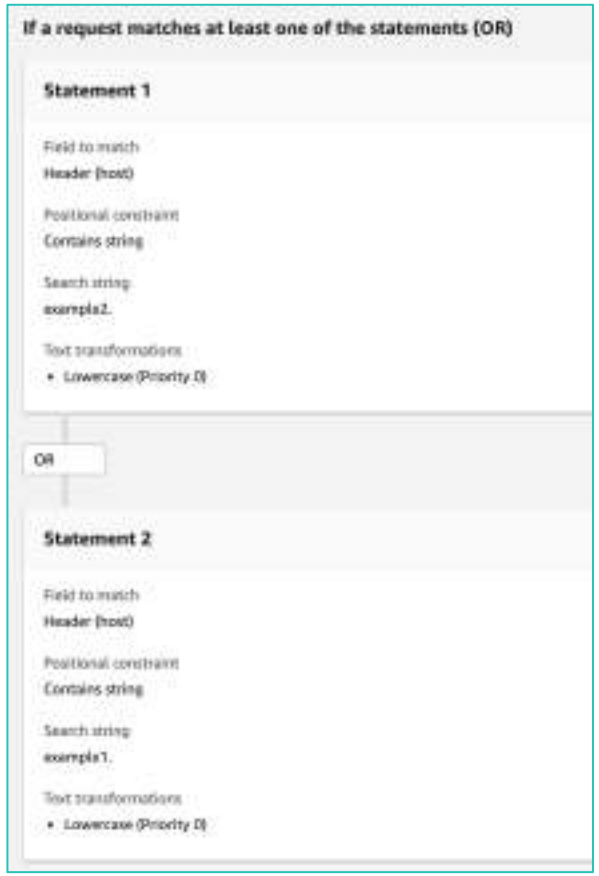


마이그레이션 참고사항

Migration wizard

- 오류만 없으면 괜찮겠지?
  - 예시 - 베타 환경에서 여러 도메인을 검증하기 위한 룰

Request header host에서
- “example1.” 또는 “example2.” 가 없으면 차단!



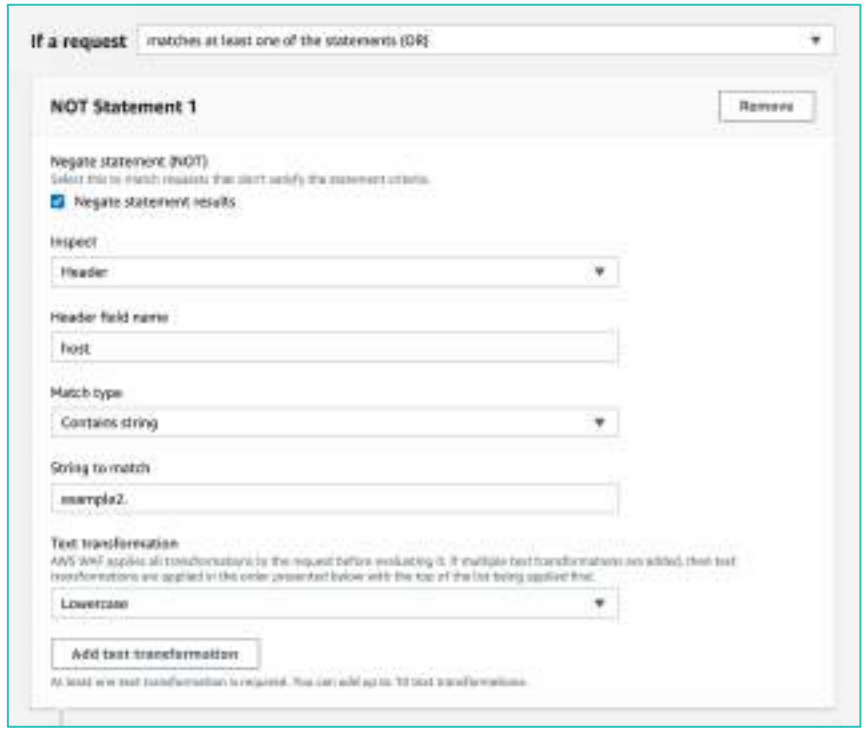
Deny!
403 Forbidden

Deny!
403 Forbidden

마이그레이션 참고사항

Migration wizard

- 오류만 없으면 괜찮겠지?
  - 기능 보강 - Negate statement (NOT) 기능 추가



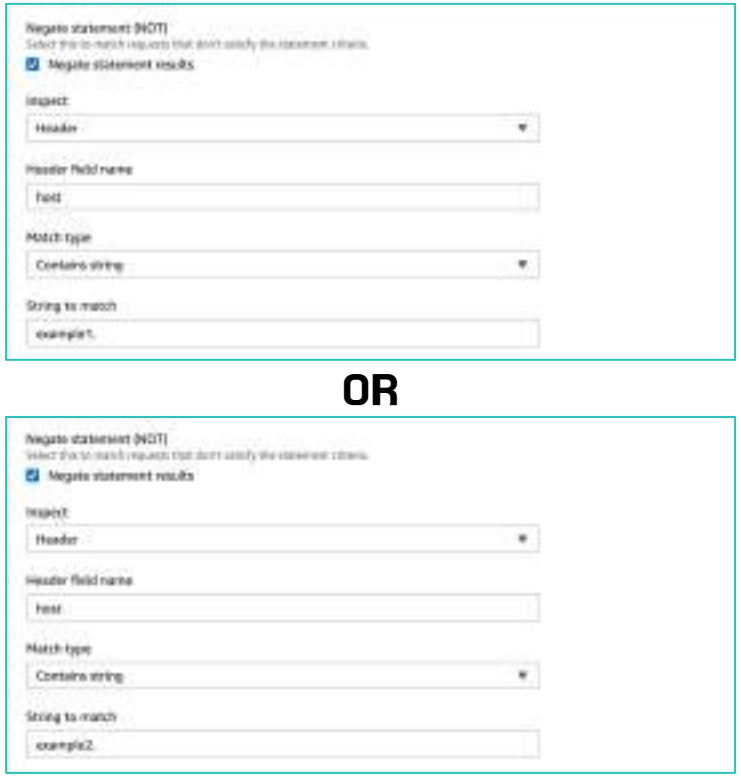
다시 예상한 결과

Request Header Host에서
- example1. 또는 example2. 가 없으면 차단!

마이그레이션 참고사항

Migration wizard

- 오류만 없으면 괜찮겠지?
  - 기능 보강 - Negate statement (NOT) 기능 추가



Deny!

service.example1.com

service.example2.com

Allow!

example1\. |. example2\.

마이그레이션 참고사항

켜진 룰도 다시 보자

- 보안 솔루션 일수록 주의가 필요하다
  - 잘못된 보안 정책으로 침해사고 발생
  - 잘못된 보안 정책으로 서비스 장애

“당연한 것은 없다”

들어주셔서
감사합니다

Thank you!

C 우아콘
WOOWA
N 2021