

CLOUD



AWS Security

박경규

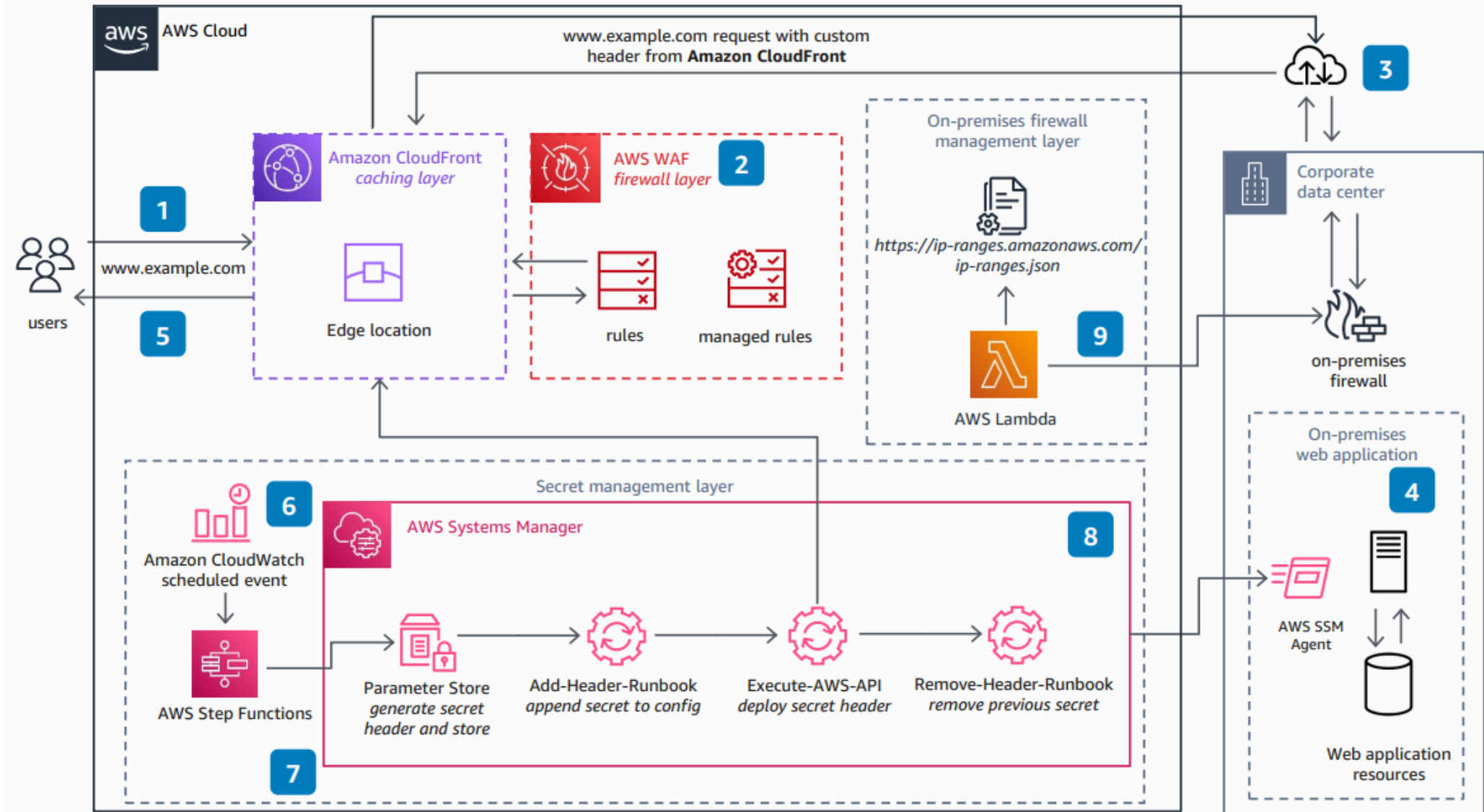
Contents

<u>1. Reference Architecture Review</u>02
<u>2. Hands-on Lab</u>13
<u>3. Design Architecture : 아키텍처 구성</u>27
<u>3. Design Architecture : 아키텍처 Pipeline Flow 분석</u>37

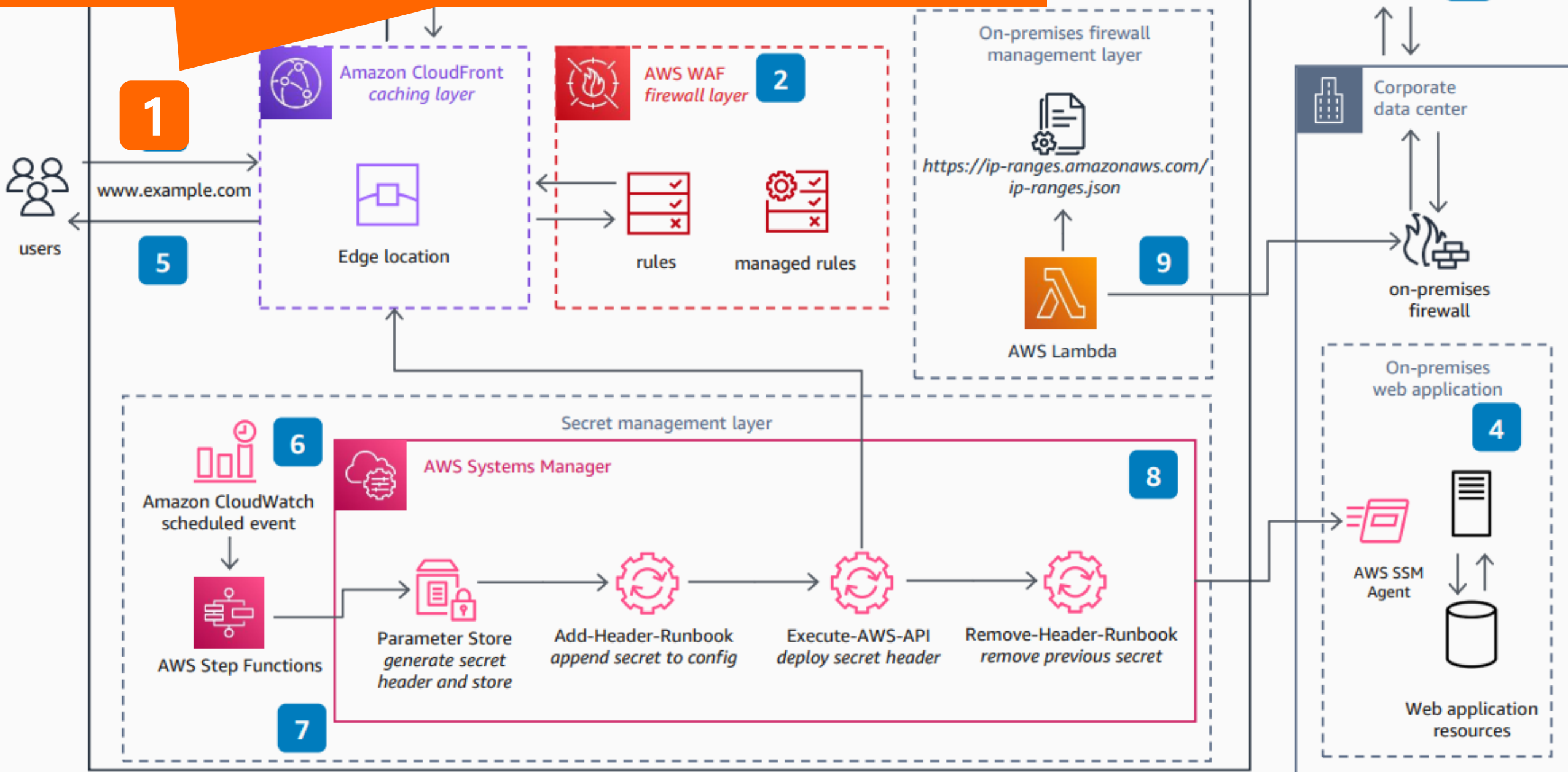
1. Reference Architecture Review

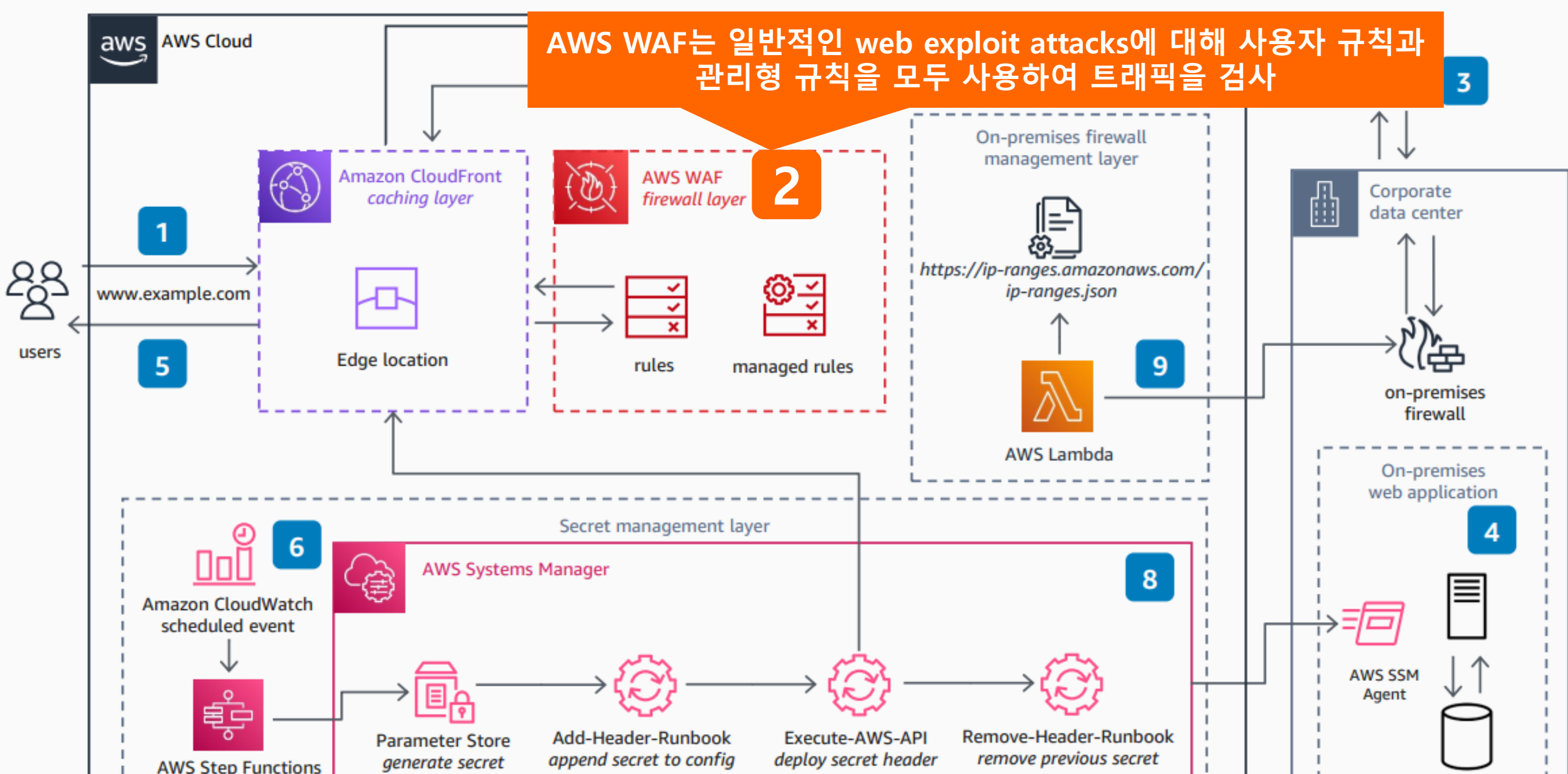
1. Reference Architecture Review

Cloud Front의 사용자 지정 오리진 및 사용자 지정 암호 헤더를 활용하여, 일반적인 웹 취약점으로부터 Endpoint를 보호하는 아키텍처

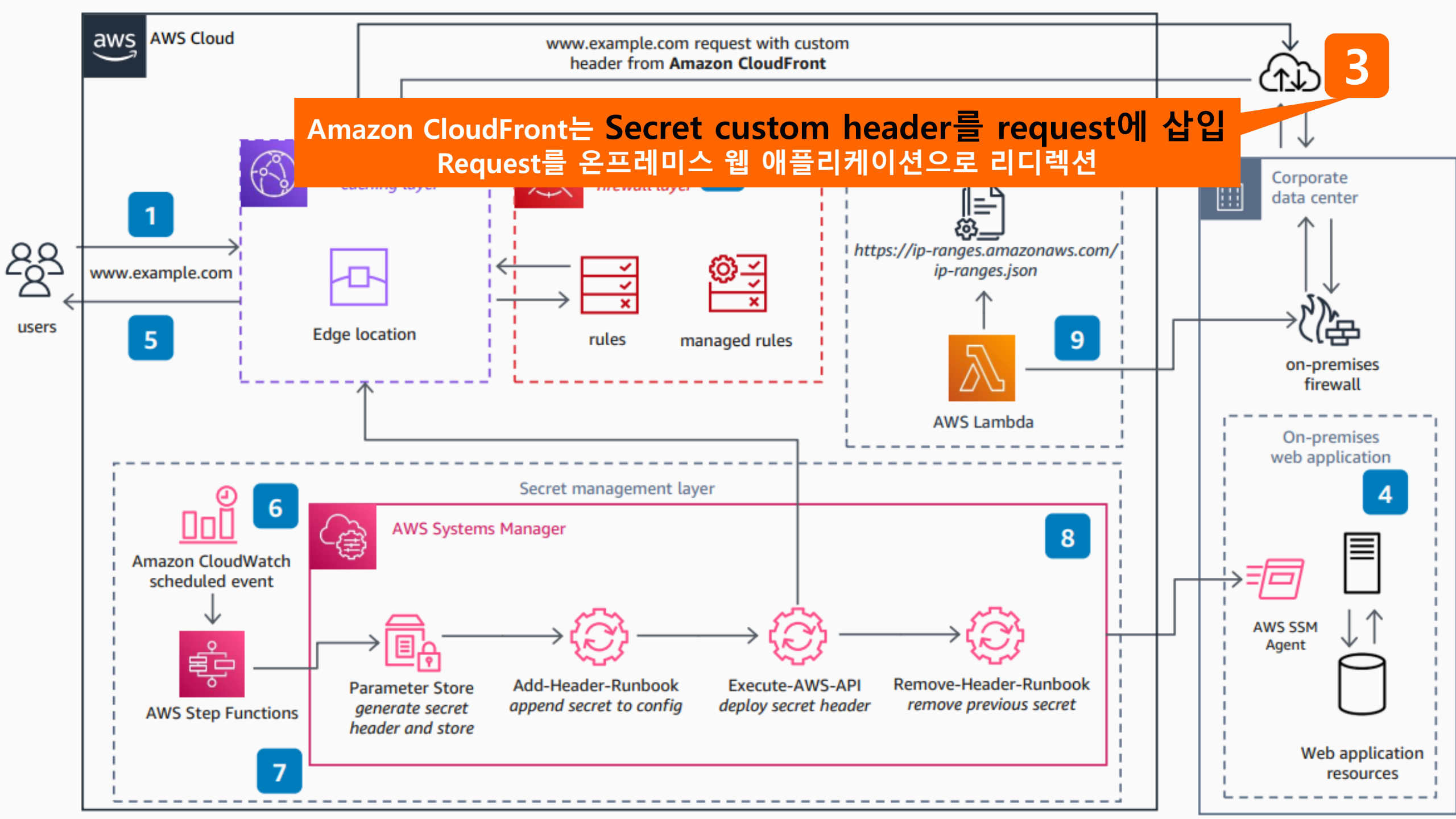


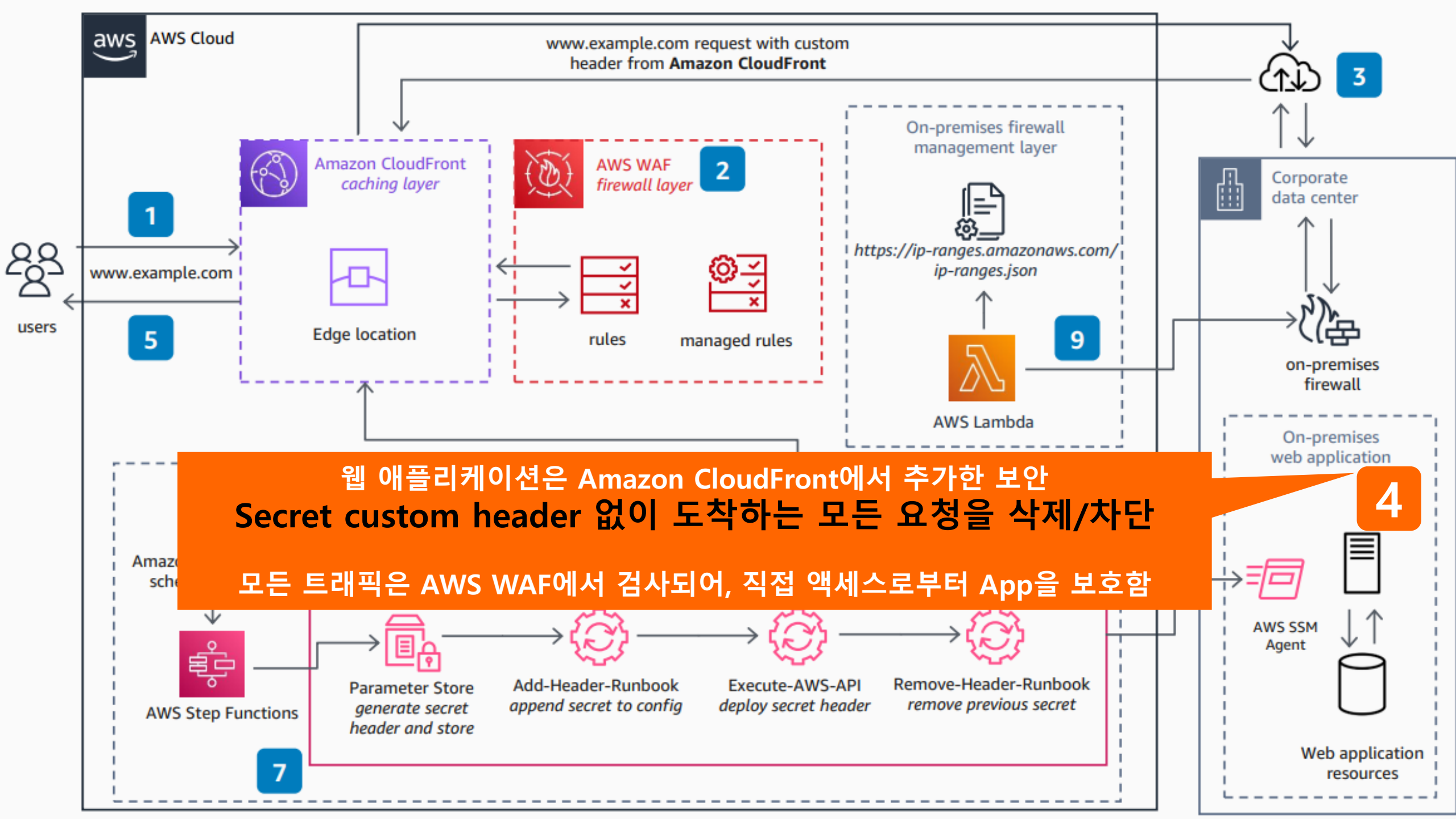
사용자 웹 접속(Request to the web application)
DNS records는 가장 가까운 CloudFront 엣지 로케이션 안내합니다.





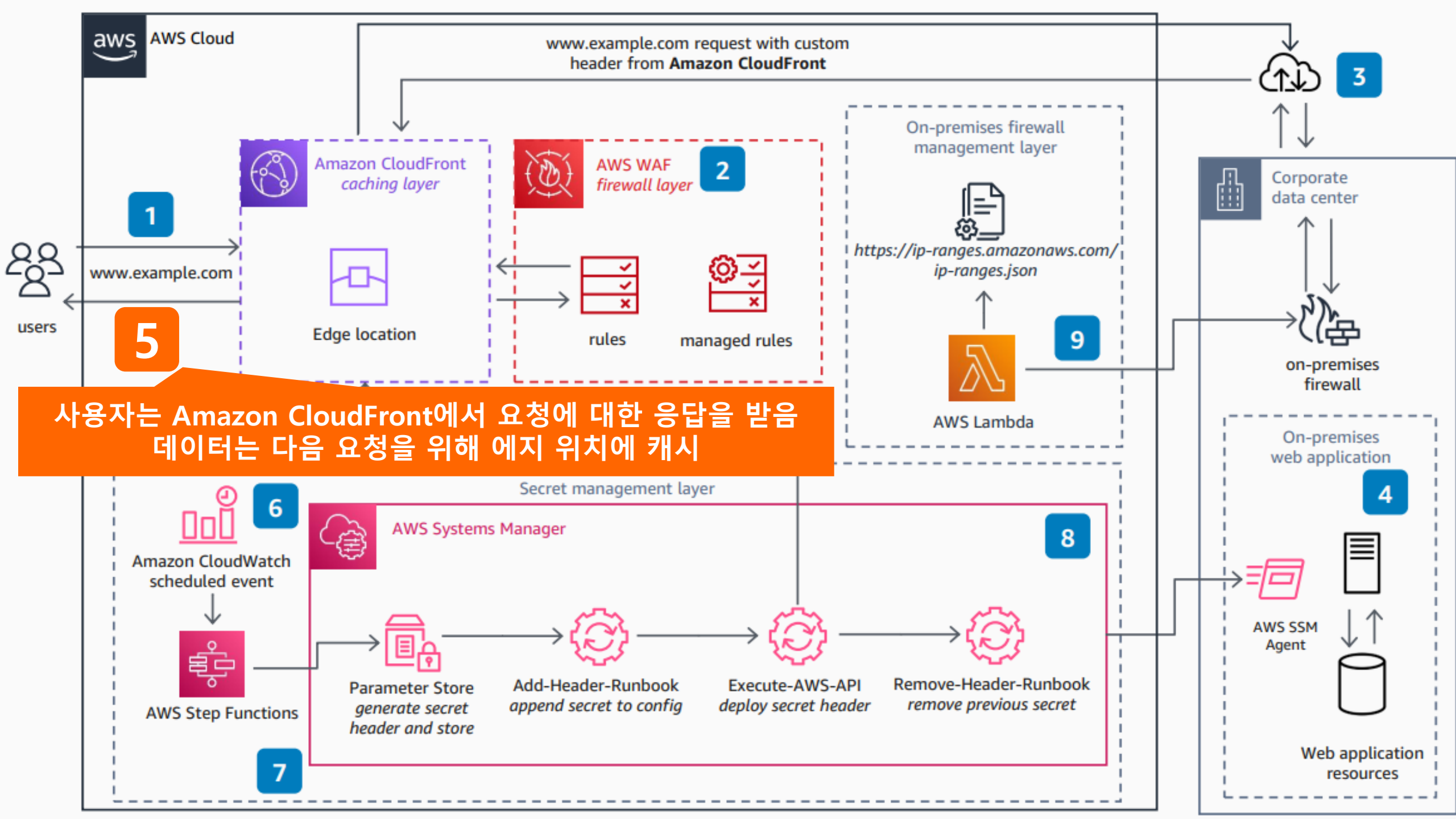
- **Web exploitation :** 웹 기반 애플리케이션의 취약점을 악용하여 민감한 데이터에 액세스하거나 앱을 제어하는 프로세스임
- 공격자는 취약점을 악용하여 앱을 장악하거나 민감한 데이터를 훔치거나 앱을 사용하여 다른 시스템에 대한 공격을 할수 있음

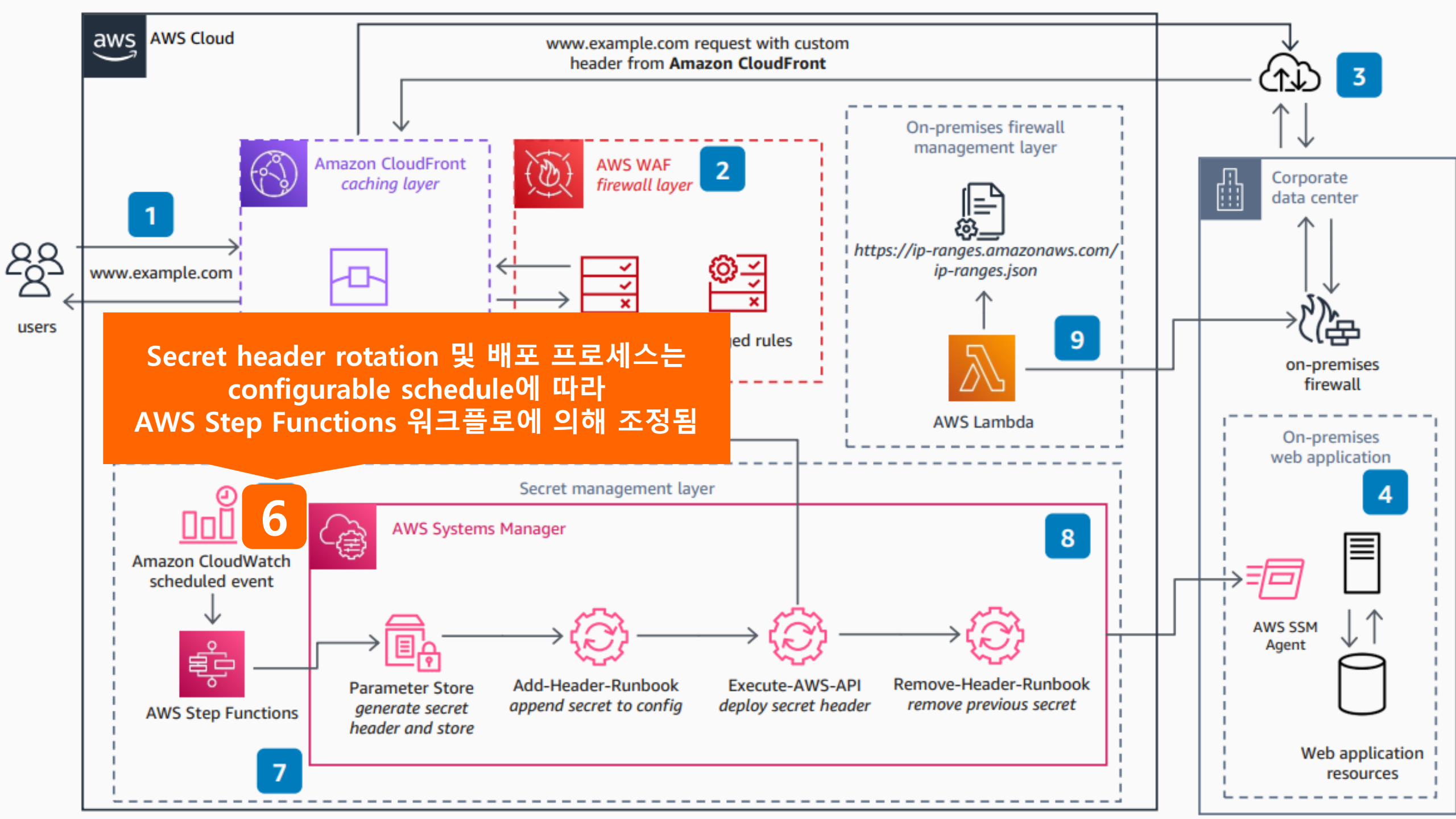


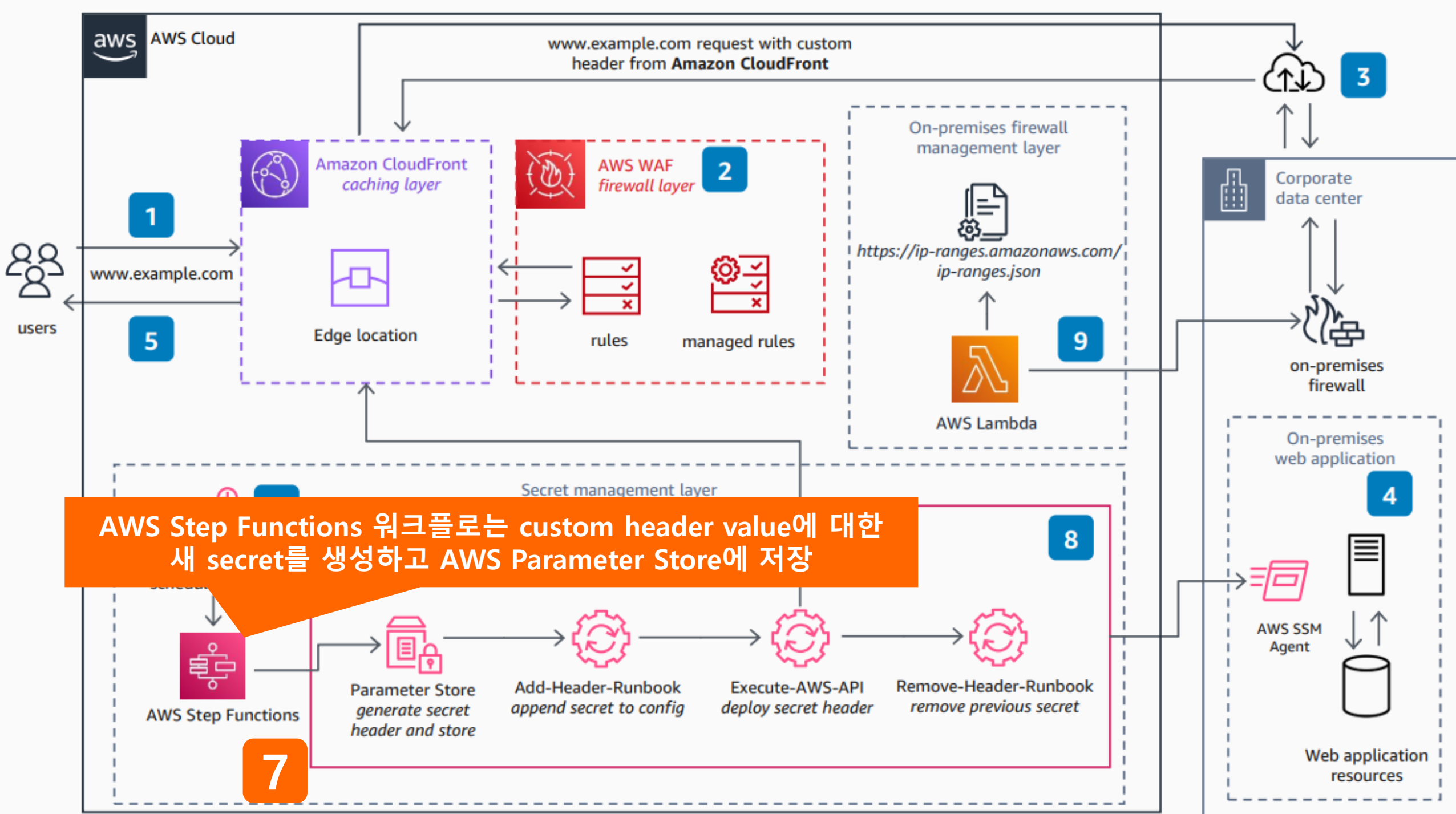


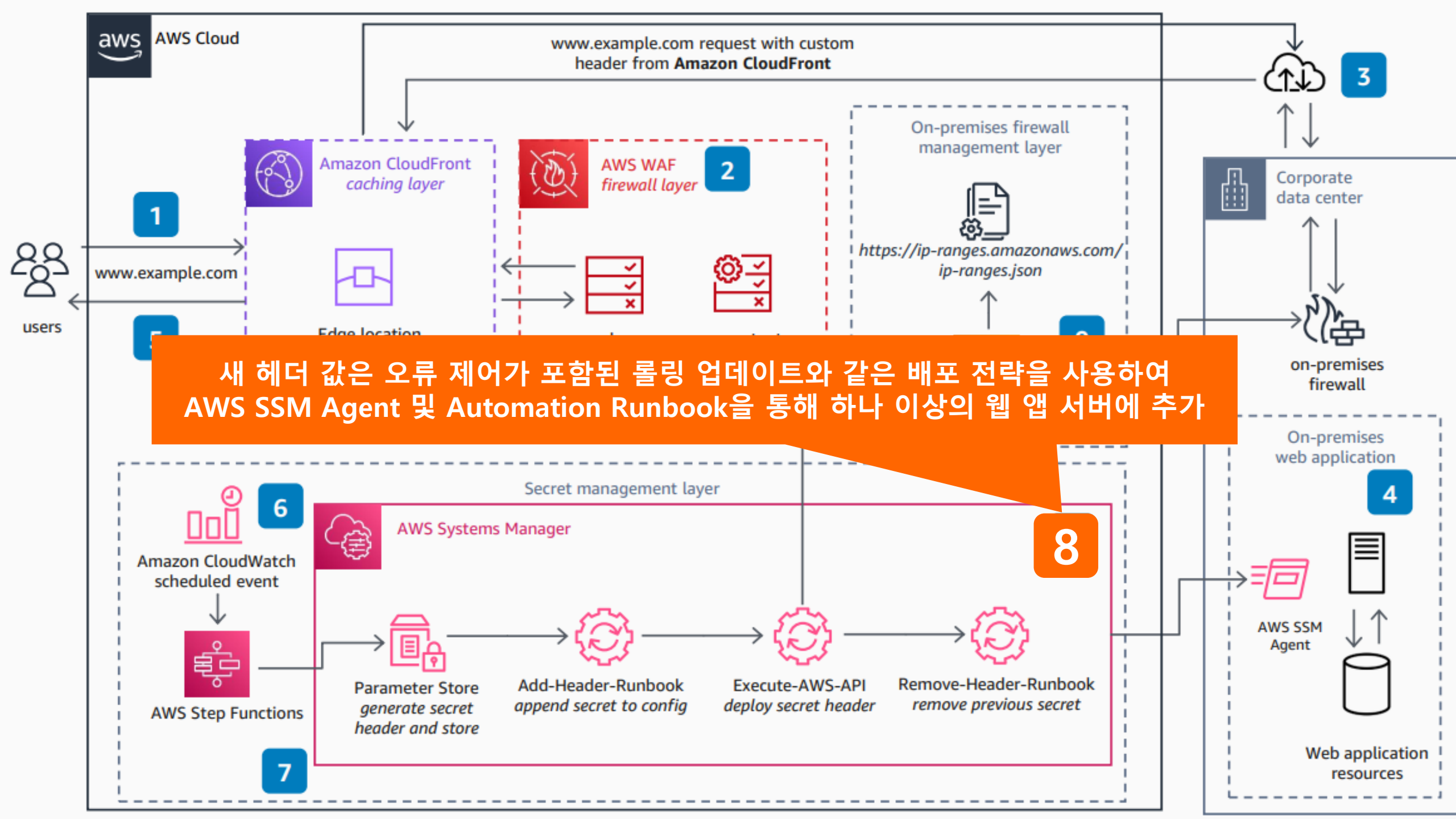
웹 애플리케이션은 Amazon CloudFront에서 추가한 보안
Secret custom header 없이 도착하는 모든 요청을 삭제/차단

모든 트래픽은 AWS WAF에서 검사되어, 직접 액세스로부터 App을 보호함

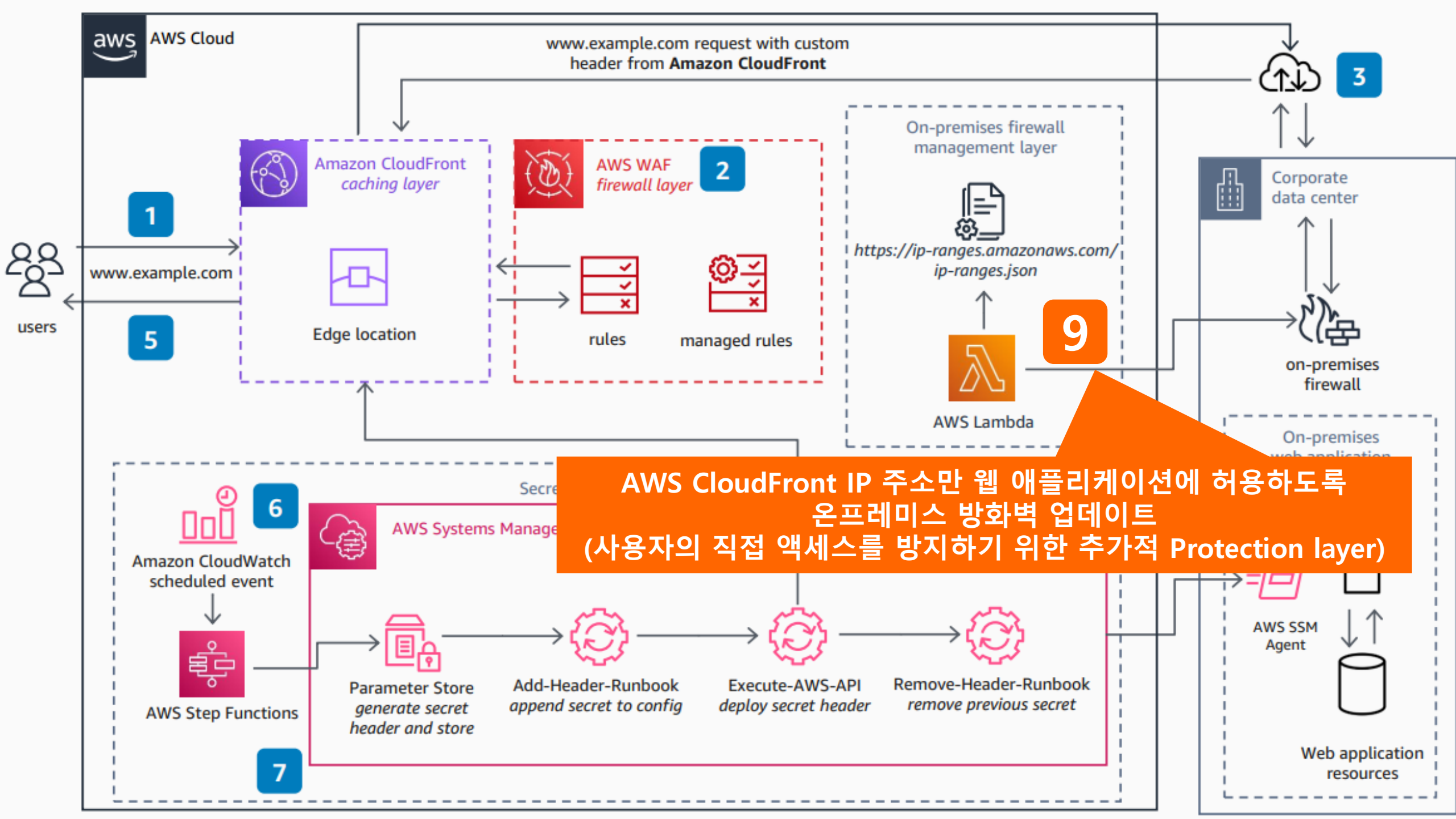








새 헤더 값은 오류 제어가 포함된 롤링 업데이트와 같은 배포 전략을 사용하여 AWS SSM Agent 및 Automation Runbook을 통해 하나 이상의 웹 앱 서버에 추가



AWS CloudFront IP 주소만 웹 애플리케이션에 허용하도록
온프레미스 방화벽 업데이트
(사용자의 직접 액세스를 방지하기 위한 추가적 Protection layer)

2. Hands-on Lab

2. Hands-on Lab - WAF

■ AWS WAF 핵심 실습

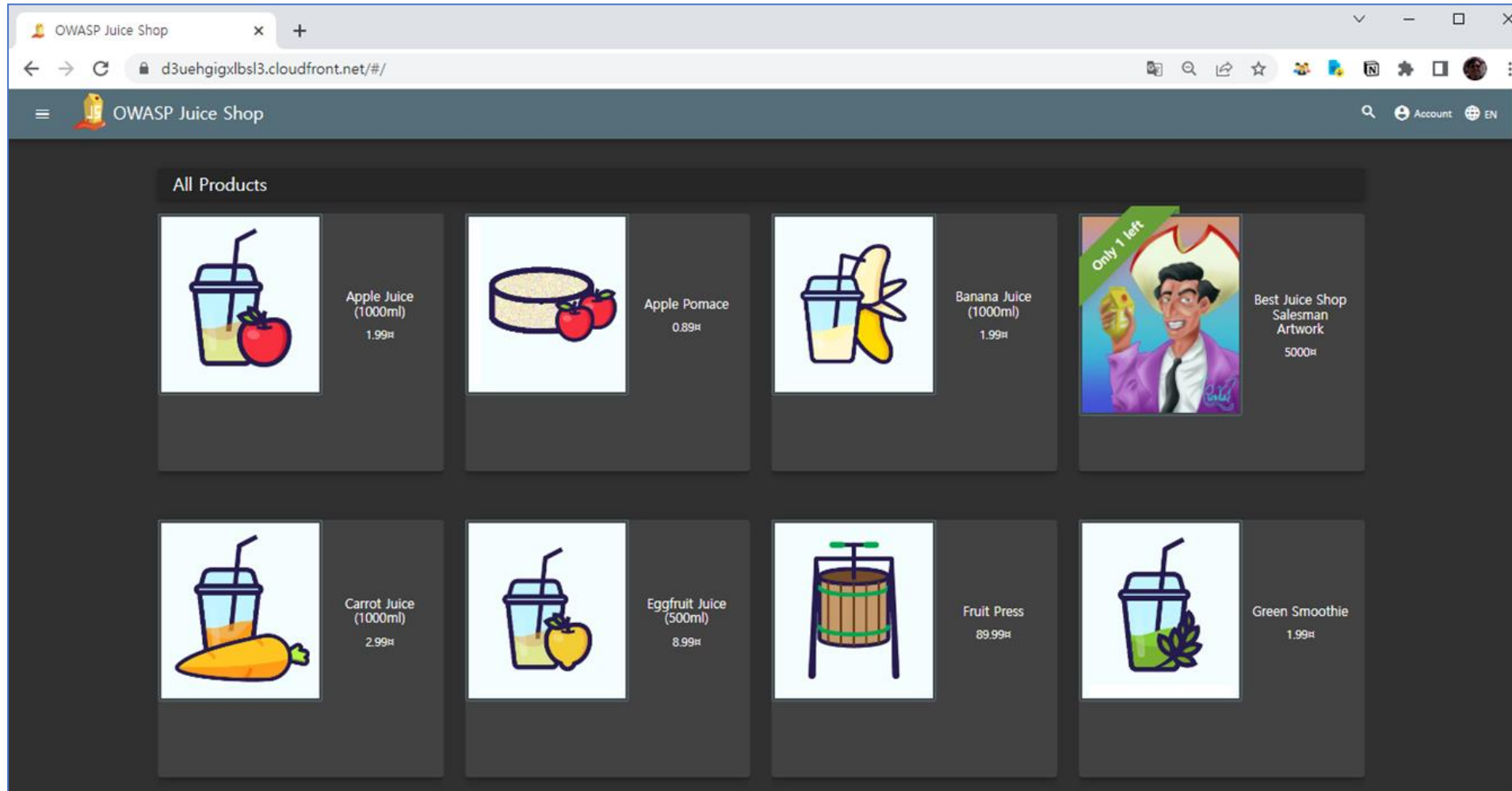
<https://catalog.us-east-1.prod.workshops.aws/workshops/c2f03000-cf61-42a6-8e62-9eaf04907417/en-US>

■ 지능형 위협탐지가 가능한 Amazon GuardDuty 실습

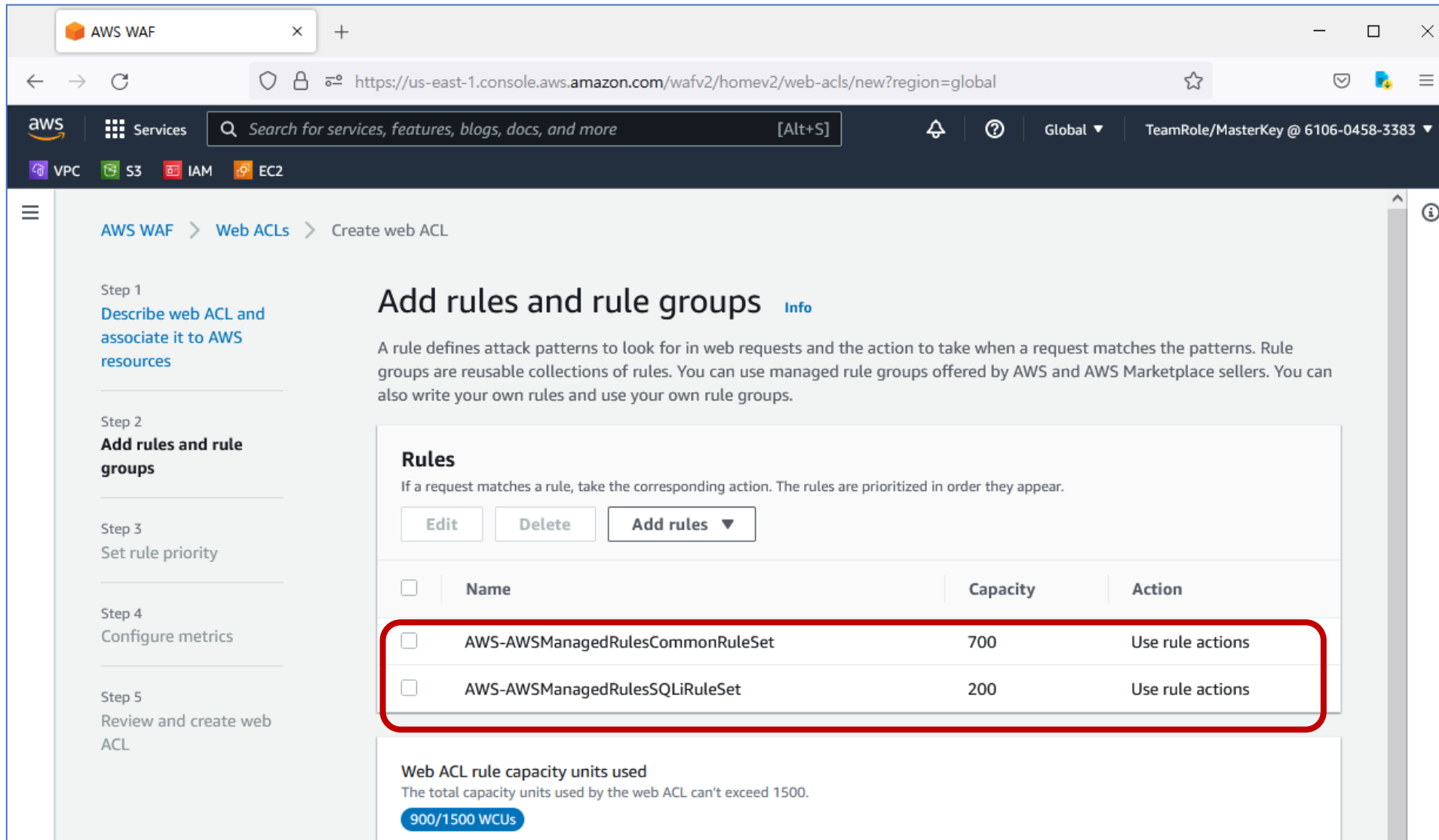
<https://catalog.workshops.aws/guardduty/en-US/introduction>

2. Hands-on Lab - WAF : OWASP Juice Shop Web

<https://d3uehgigxlb3l3.cloudfront.net/>



2. Hands-on Lab - WAF : Web ACL Managed Rules



The screenshot shows the AWS WAF console interface for creating a new web ACL. The breadcrumb navigation indicates the path: AWS WAF > Web ACLs > Create web ACL. The left sidebar shows the progress of the wizard, with Step 2, 'Add rules and rule groups', currently active.

Add rules and rule groups [Info](#)

A rule defines attack patterns to look for in web requests and the action to take when a request matches the patterns. Rule groups are reusable collections of rules. You can use managed rule groups offered by AWS and AWS Marketplace sellers. You can also write your own rules and use your own rule groups.

Rules
If a request matches a rule, take the corresponding action. The rules are prioritized in order they appear.

[Edit](#) [Delete](#) [Add rules ▼](#)

<input type="checkbox"/>	Name	Capacity	Action
<input type="checkbox"/>	AWS-AWSManagedRulesCommonRuleSet	700	Use rule actions
<input type="checkbox"/>	AWS-AWSManagedRulesSQLiRuleSet	200	Use rule actions

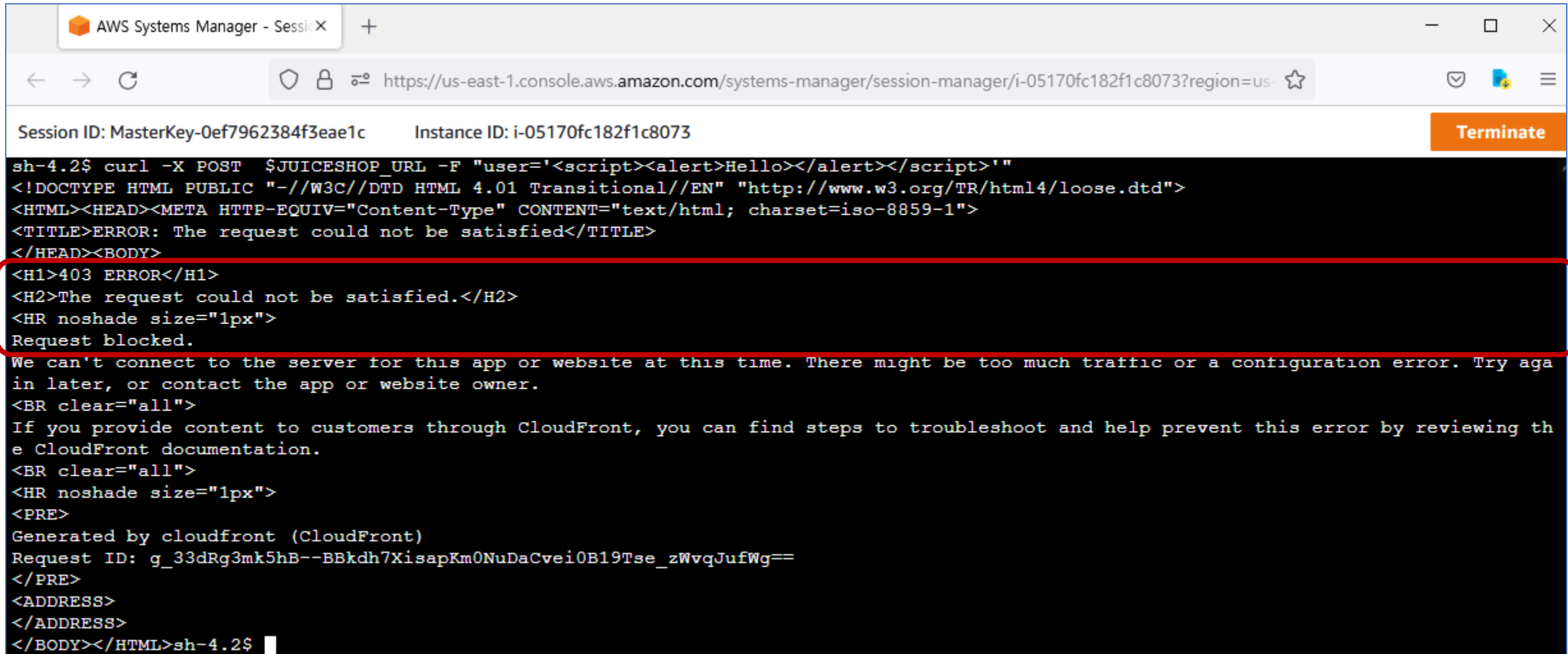
Web ACL rule capacity units used
The total capacity units used by the web ACL can't exceed 1500.

900/1500 WCUs

2. Hands-on Lab - WAF : Web ACL Managed Rules

```
export JUICESHOP_URL=d3uehgigxlbsl3.cloudfront.net
```

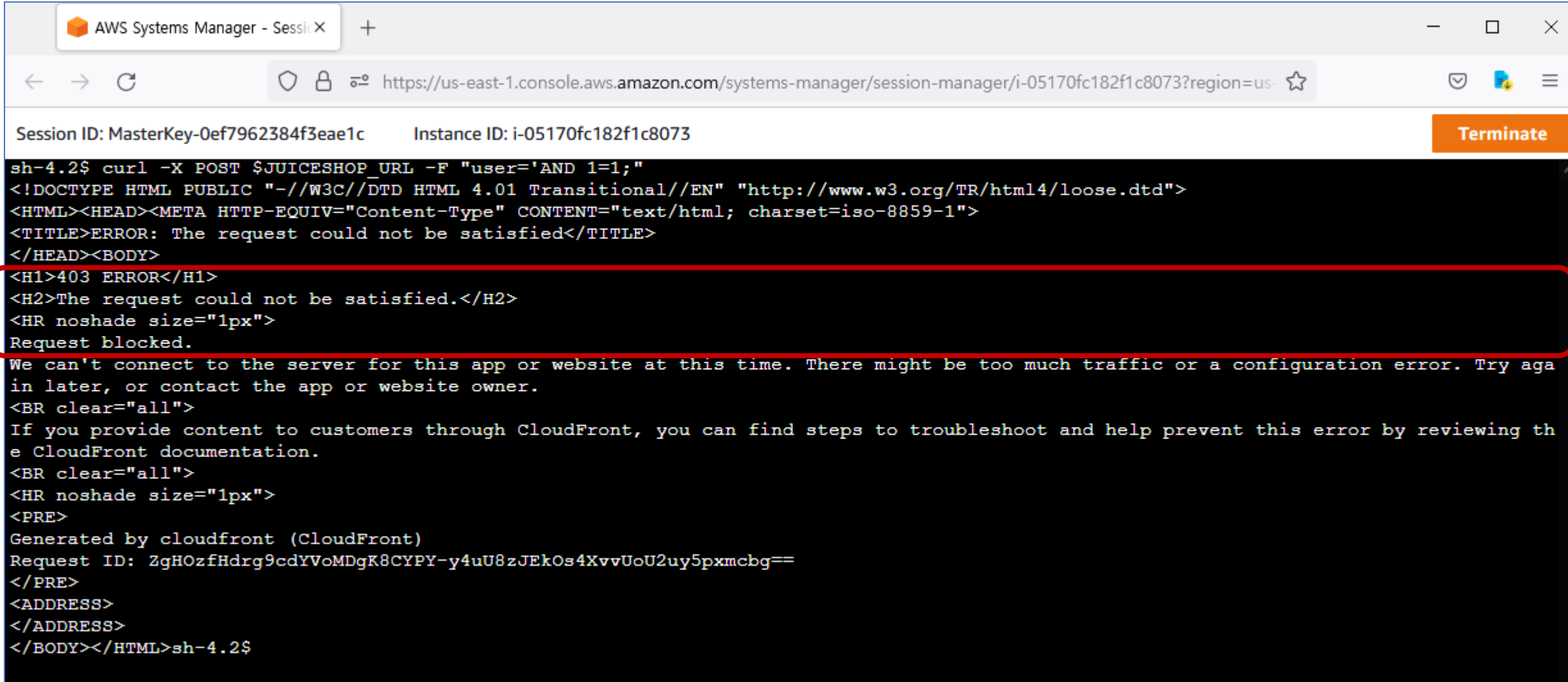
```
curl -X POST $JUICESHOP_URL -F "user='<script><alert>Hello></alert></script>'"
```

A screenshot of the AWS Systems Manager console. The browser tab is titled 'AWS Systems Manager - Session Manager'. The address bar shows the URL 'https://us-east-1.console.aws.amazon.com/systems-manager/session-manager/i-05170fc182f1c8073?region=us-'. The console displays session details: 'Session ID: MasterKey-0ef7962384f3eae1c' and 'Instance ID: i-05170fc182f1c8073'. A 'Terminate' button is visible in the top right. The terminal window shows a command prompt 'sh-4.2\$' followed by the curl command. The output is an HTML error page with a 403 status and the message 'The request could not be satisfied'. A red rectangle highlights the error details: '<H1>403 ERROR</H1>', '<H2>The request could not be satisfied.</H2>', '<HR noshade size="1px">', and 'Request blocked.'. Below this, there is a message about connecting to the server and a link to CloudFront documentation. The terminal ends with 'Generated by cloudfront (CloudFront)', 'Request ID: g_33dRg3mk5hB--BBkdh7XisapKm0NuDaCvei0B19Tse_zWvqJufWg==', and some HTML tags. The prompt 'sh-4.2\$' is visible at the bottom.

```
sh-4.2$ curl -X POST $JUICESHOP_URL -F "user='<script><alert>Hello></alert></script>'"
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<HTML><HEAD><META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
<TITLE>ERROR: The request could not be satisfied</TITLE>
</HEAD><BODY>
<H1>403 ERROR</H1>
<H2>The request could not be satisfied.</H2>
<HR noshade size="1px">
Request blocked.
We can't connect to the server for this app or website at this time. There might be too much traffic or a configuration error. Try again later, or contact the app or website owner.
<BR clear="all">
If you provide content to customers through CloudFront, you can find steps to troubleshoot and help prevent this error by reviewing the CloudFront documentation.
<BR clear="all">
<HR noshade size="1px">
<PRE>
Generated by cloudfront (CloudFront)
Request ID: g_33dRg3mk5hB--BBkdh7XisapKm0NuDaCvei0B19Tse_zWvqJufWg==
</PRE>
<ADDRESS>
</ADDRESS>
</BODY></HTML>sh-4.2$
```

2. Hands-on Lab - WAF : Web ACL Managed Rules

```
curl -X POST $JUICESHOP_URL -F "user='AND 1=1;'"
```

The screenshot shows the AWS Systems Manager console interface. At the top, there's a browser window with the URL 'https://us-east-1.console.aws.amazon.com/systems-manager/session-manager/i-05170fc182f1c8073?region=us-'. Below the browser window, the console displays 'Session ID: MasterKey-0ef7962384f3eae1c' and 'Instance ID: i-05170fc182f1c8073'. A red box highlights a terminal output showing a 403 error. The terminal text is as follows:

```
sh-4.2$ curl -X POST $JUICESHOP_URL -F "user='AND 1=1;'"
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<HTML><HEAD><META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
<TITLE>ERROR: The request could not be satisfied</TITLE>
</HEAD><BODY>
<H1>403 ERROR</H1>
<H2>The request could not be satisfied.</H2>
<HR noshade size="1px">
Request blocked.
We can't connect to the server for this app or website at this time. There might be too much traffic or a configuration error. Try again later, or contact the app or website owner.
<BR clear="all">
If you provide content to customers through CloudFront, you can find steps to troubleshoot and help prevent this error by reviewing the CloudFront documentation.
<BR clear="all">
<HR noshade size="1px">
<PRE>
Generated by cloudfront (CloudFront)
Request ID: ZgHOzfHdrg9cdYVoMDgK8CYPY-y4uU8zJEkOs4XvvUoU2uy5pxmcbg==
</PRE>
<ADDRESS>
</ADDRESS>
</BODY></HTML>sh-4.2$
```

2. Hands-on Lab - WAF : Web ACL Custom Rules

The screenshot shows the AWS WAF console interface. At the top, there's a navigation bar with the AWS logo, 'Services' link, a search bar, and user information. Below this is a sidebar with icons for VPC, S3, IAM, and EC2. The main content area has a header with a note about the JSON editor. The 'Rule' configuration form is highlighted with a red border. It includes fields for 'Name' (TomatoAttackRuleSet), 'Type' (Regular rule selected), 'If a request' (matches the statement), 'Statement' (Inspect), 'Inspect' (Single header), 'Header field name' (x-tomatoattack), 'Match type' (Size greater than or equal to), and 'Size in bytes' (0). A 'Validate' button is located to the right of the form.

Rule

Name

TomatoAttackRuleSet

The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

Type

☒ Regular rule

☐ Rate-based rule

If a request matches the statement

Statement

Inspect

Single header

Header field name

x-tomatoattack

Match type

Size greater than or equal to

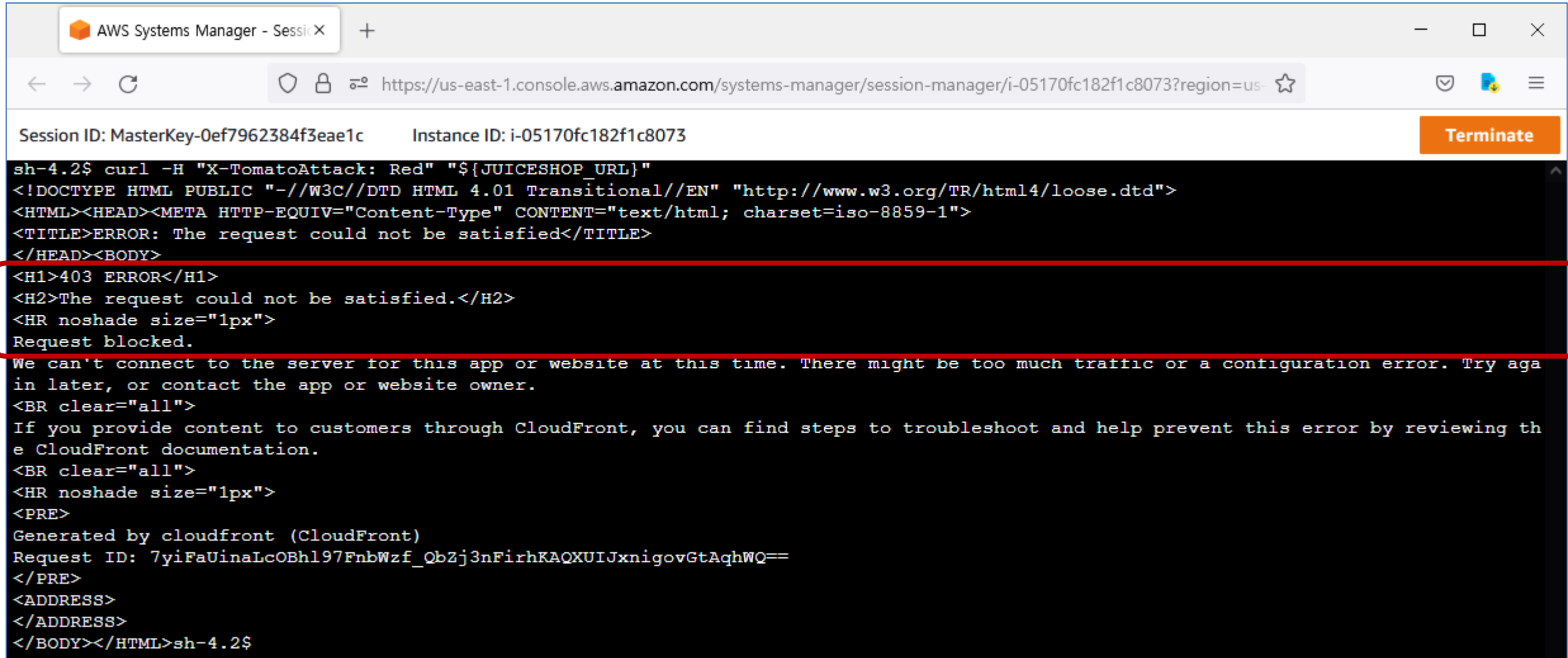
Size in bytes

0

Validate

2. Hands-on Lab - WAF : Web ACL Managed Rules

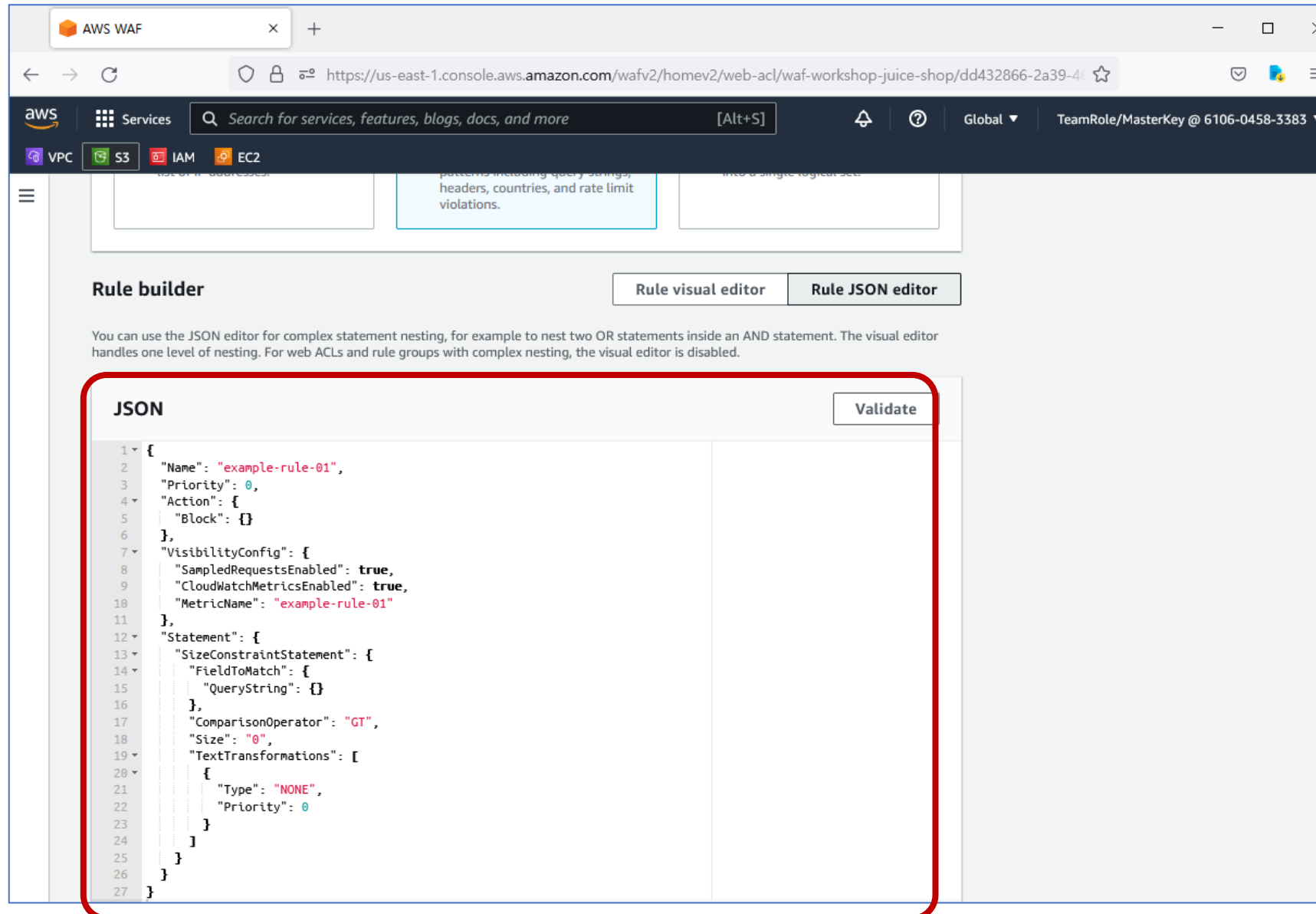
```
curl -H "X-TomatoAttack: Red" "${JUICESHOP_URL}"  
curl -H "X-TomatoAttack: Green" "${JUICESHOP_URL}"
```



The screenshot shows the AWS Systems Manager console interface. At the top, there's a browser window with the URL `https://us-east-1.console.aws.amazon.com/systems-manager/session-manager/i-05170fc182f1c8073?region=us-`. Below the browser window, the session details are displayed: Session ID: MasterKey-0ef7962384f3eae1c and Instance ID: i-05170fc182f1c8073. A red box highlights the terminal output, which shows a 403 error response from the web application. The terminal output is as follows:

```
sh-4.2$ curl -H "X-TomatoAttack: Red" "${JUICESHOP_URL}"  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  
<HTML><HEAD><META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">  
<TITLE>ERROR: The request could not be satisfied</TITLE>  
</HEAD><BODY>  
<H1>403 ERROR</H1>  
<H2>The request could not be satisfied.</H2>  
<HR noshade size="1px">  
Request blocked.  
We can't connect to the server for this app or website at this time. There might be too much traffic or a configuration error. Try again later, or contact the app or website owner.  
<BR clear="all">  
If you provide content to customers through CloudFront, you can find steps to troubleshoot and help prevent this error by reviewing the CloudFront documentation.  
<BR clear="all">  
<HR noshade size="1px">  
<PRE>  
Generated by cloudfront (CloudFront)  
Request ID: 7yiFaUinaLcOBh197FnbWzf_QbZj3nFirhKAQXUIJxnigovGtAqhWQ==  
</PRE>  
<ADDRESS>  
</ADDRESS>  
</BODY></HTML>sh-4.2$
```

2. Hands-on Lab - WAF : Web ACL Custom Rules



The screenshot shows the AWS WAF console interface. At the top, there's a navigation bar with the AWS logo, a search bar, and various service icons (VPC, S3, IAM, EC2). Below the navigation bar, there's a section titled "Rule builder" with two tabs: "Rule visual editor" and "Rule JSON editor". The "Rule JSON editor" is selected, and it displays a JSON rule configuration. The JSON is as follows:

```
1 {
2   "Name": "example-rule-01",
3   "Priority": 0,
4   "Action": {
5     "Block": {}
6   },
7   "VisibilityConfig": {
8     "SampledRequestsEnabled": true,
9     "CloudWatchMetricsEnabled": true,
10    "MetricName": "example-rule-01"
11  },
12  "Statement": {
13    "SizeConstraintStatement": {
14      "FieldToMatch": {
15        "QueryString": {}
16      },
17      "ComparisonOperator": "GT",
18      "Size": "0",
19      "TextTransformations": [
20        {
21          "Type": "NONE",
22          "Priority": 0
23        }
24      ]
25    }
26  }
27 }
```

A red box highlights the JSON editor area, including the "Validate" button at the top right of the editor.

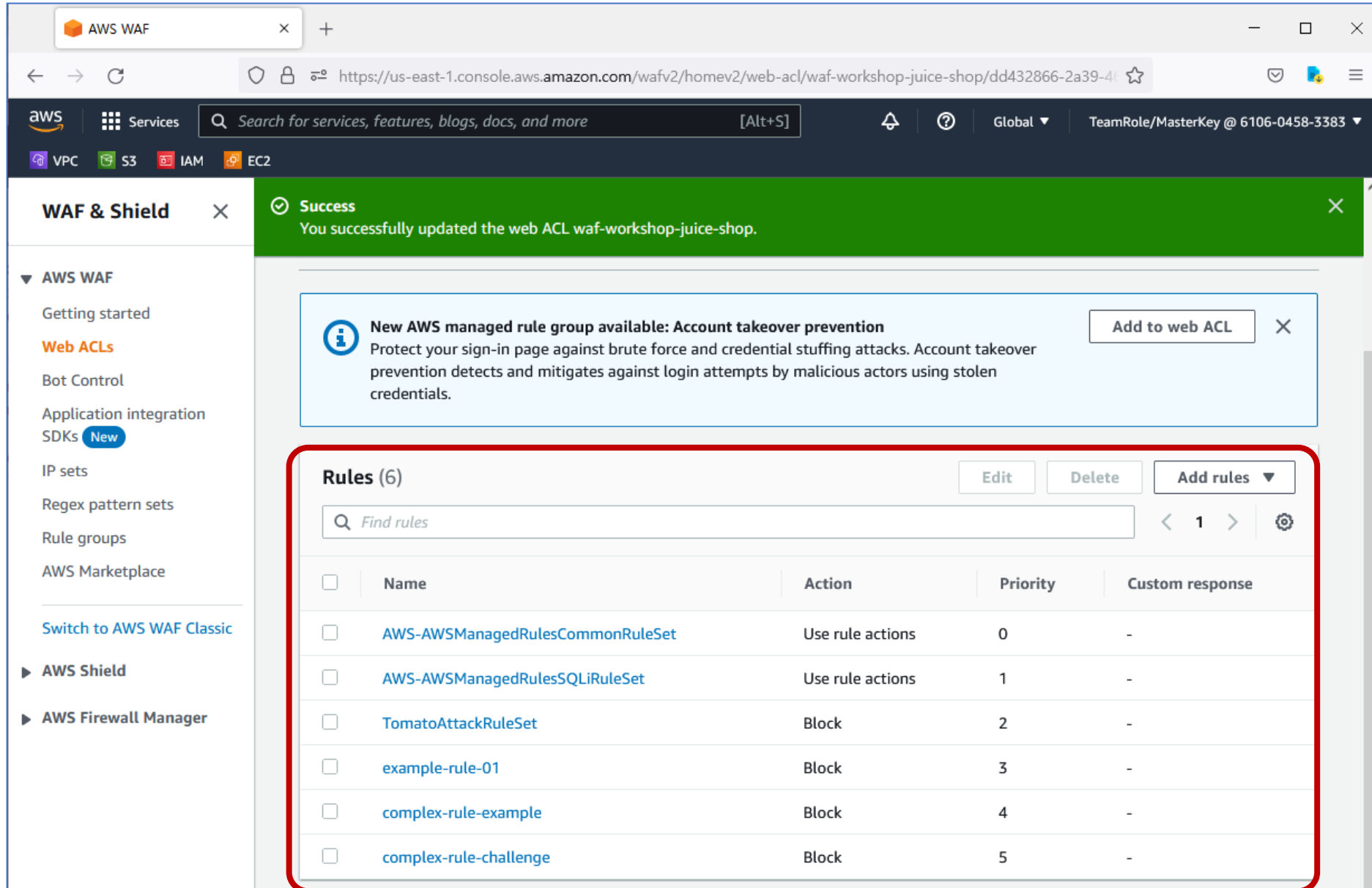


This screenshot shows the same AWS WAF console interface, but with a different JSON rule configuration. The JSON is as follows:

```
1 {
2   "Name": "complex-rule-example",
3   "Priority": 0,
4   "Action": {
5     "Block": {}
6   },
7   "VisibilityConfig": {
8     "SampledRequestsEnabled": true,
9     "CloudWatchMetricsEnabled": true,
10    "MetricName": "complex-rule-example"
11  },
12  "Statement": {
13    "AndStatement": {
14      "Statements": [
15        {
16          "SizeConstraintStatement": {
17            "FieldToMatch": {
18              "Body": {}
19            },
20            "ComparisonOperator": "GT",
21            "Size": "100",
22            "TextTransformations": [
23              {
24                "Type": "NONE",
25                "Priority": 0
26              }
27            ]
28          }
29        },
30        {
31          "NotStatement": {
32            "Statement": {
33              "ByteMatchStatement": {
34                "FieldToMatch": {
35                  "SingleHeader": {
36                    "Name": "x-upload-body"
37                  }
38                },
39                "PositionalConstraint": "EXACTLY",
40                "SearchString": "true",
41                "TextTransformations": [
42                  {
43                    "Type": "NONE",
44                    "Priority": 0
45                  }
46                ]
47              }
48            }
49          }
50        }
51      ]
52    }
53  }
54 }
```

A red box highlights the entire JSON editor area, including the "Validate" button at the top right of the editor.

2. Hands-on Lab - WAF : Web ACL Custom Rules



The screenshot shows the AWS WAF console interface. A green success banner at the top indicates that the web ACL 'waf-workshop-juice-shop' has been successfully updated. Below this, a notification box informs that a new AWS managed rule group, 'Account takeover prevention', is available for addition to the web ACL. The main section displays a list of rules for the selected web ACL, titled 'Rules (6)'. The rules are listed in a table with columns for Name, Action, Priority, and Custom response. The rules are: AWS-AWSManagedRulesCommonRuleSet (Priority 0), AWS-AWSManagedRulesSQLiRuleSet (Priority 1), TomatoAttackRuleSet (Priority 2), example-rule-01 (Priority 3), complex-rule-example (Priority 4), and complex-rule-challenge (Priority 5). The 'Rules (6)' section is highlighted with a red border.

WAF & Shield

- AWS WAF
 - Getting started
 - Web ACLs**
 - Bot Control
 - Application integration SDKs **New**
 - IP sets
 - Regex pattern sets
 - Rule groups
 - AWS Marketplace
 - [Switch to AWS WAF Classic](#)
- AWS Shield
- AWS Firewall Manager

Success
You successfully updated the web ACL waf-workshop-juice-shop.

New AWS managed rule group available: Account takeover prevention
Protect your sign-in page against brute force and credential stuffing attacks. Account takeover prevention detects and mitigates against login attempts by malicious actors using stolen credentials.
[Add to web ACL](#)

Rules (6) [Edit](#) [Delete](#) [Add rules](#)

<input type="checkbox"/>	Name	Action	Priority	Custom response
<input type="checkbox"/>	AWS-AWSManagedRulesCommonRuleSet	Use rule actions	0	-
<input type="checkbox"/>	AWS-AWSManagedRulesSQLiRuleSet	Use rule actions	1	-
<input type="checkbox"/>	TomatoAttackRuleSet	Block	2	-
<input type="checkbox"/>	example-rule-01	Block	3	-
<input type="checkbox"/>	complex-rule-example	Block	4	-
<input type="checkbox"/>	complex-rule-challenge	Block	5	-

2. Hands-on Lab - GuardDuty : Findings

The screenshot displays the AWS GuardDuty console interface. At the top, a green banner confirms that GuardDuty has been successfully enabled. Below this, a light blue notification box highlights a new feature: Amazon GuardDuty is now available in the Middle East (UAE) Region. The main content area is titled 'GuardDuty > Findings' and shows three status indicators: 0 findings, 0 suppressed findings, and 0 saved rules. The 'Findings' section includes a 'Suppress Findings' button, an 'Info' link, and a 'Saved rules' field currently showing 'No saved rules'. A filter dropdown is set to 'Current', and a search bar is available with the placeholder text 'Add filter criteria'. Below the filter bar, a table header is visible with columns for 'Finding type', 'Resource', 'L...', and 'Co...'. A light blue message box at the bottom states, 'You don't have any findings. GuardDuty continuously monitors your AWS environment and reports findings on this page. Learn more'. The left sidebar contains navigation links for 'Findings', 'Usage', 'Malware scans', 'Settings', 'Lists', 'S3 Protection', 'Kubernetes Protection', 'Malware Protection' (marked as 'New!'), 'Accounts', 'What's New', and 'Partners'.

2. Hands-on Lab - GuardDuty : Usage

The screenshot shows the AWS GuardDuty console interface. The left sidebar contains navigation links for Findings, Usage (selected), Malware scans, Settings (Lists, S3 Protection, Kubernetes Protection, Malware Protection, Accounts), What's New, and Partners. The main content area is titled 'GuardDuty > Usage' and shows the 'Usage' page. The estimated total daily cost is \$0.00. A note states that some features are still in free trial. The 'Breakdown by data source' section shows two data sources: CloudTrail and VPC Flow Logs, both with a status of 'Pending' and a free trial ending on November 11 (30 days remaining).

GuardDuty Management Console

https://us-east-1.console.aws.amazon.com/guardduty/home?region=us-east-1#/usage

Search for services, features, blogs, docs, and more [Alt+S]

N. Virginia TeamRole/MasterKey @ 6106-0458-3383

VPC S3 IAM EC2

GuardDuty

Findings

Usage

Malware scans

Settings

Lists

S3 Protection

Kubernetes Protection

Malware Protection **New!**

Accounts

What's New

Partners

GuardDuty > Usage

Usage [Info](#)

Estimated total daily cost **\$0.00**

[About GuardDuty pricing](#)

- Some features are still in free trial. You pay nothing for these features while free trials are in effect. These estimates reflect what you can expect to pay after your free trial ends.

Breakdown by data source Average daily cost

CloudTrail	Pending
Daily cost will be available 7 days after enabling data source.	Free trial ends November 11 (30 days remaining)
VPC Flow Logs	Pending
Daily cost will be available 7 days after enabling data source.	Free trial ends November 11 (30 days remaining)

3. Design Architecture

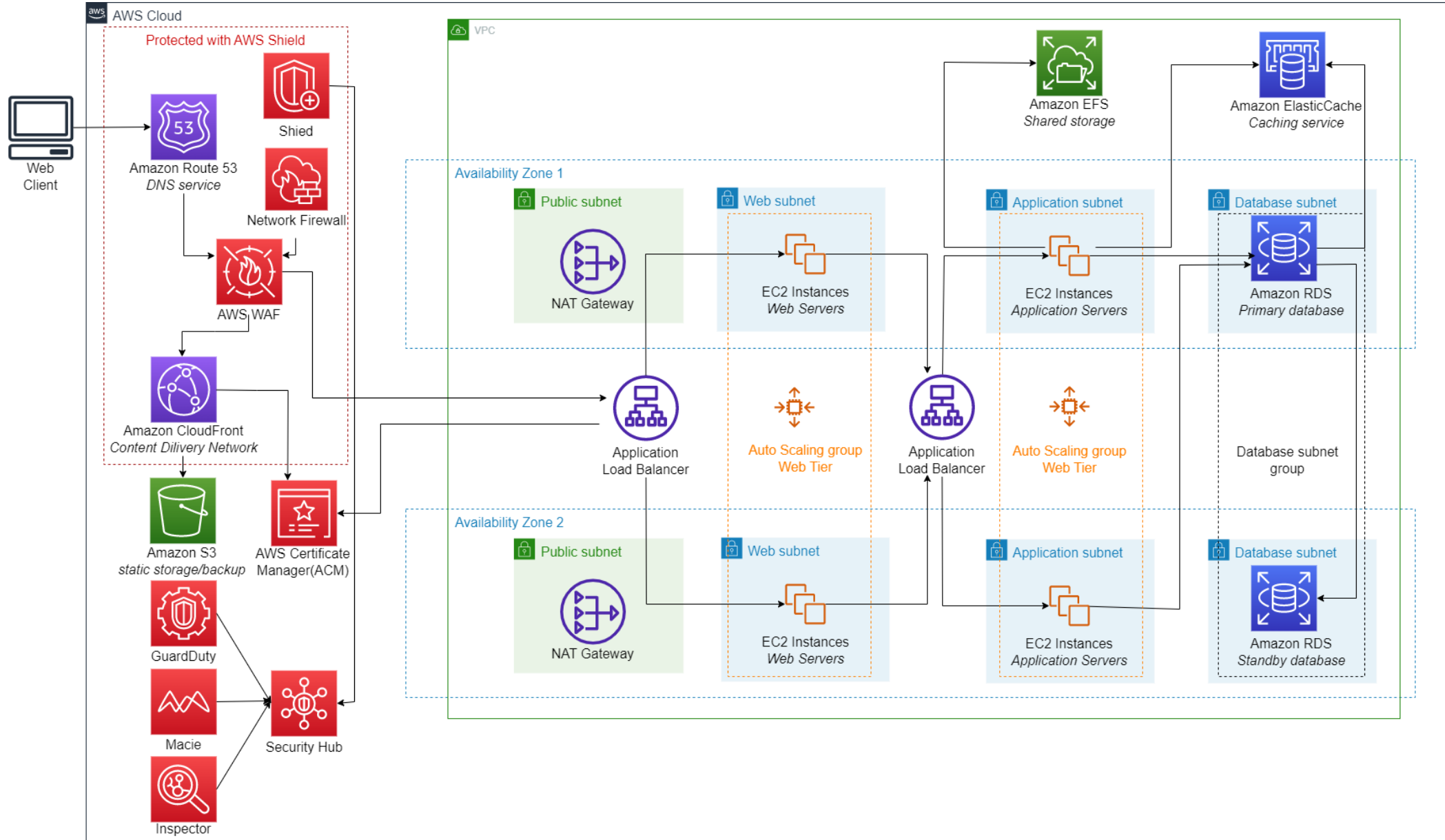
아키텍처 구성

3. Design Architecture

■ Scenario

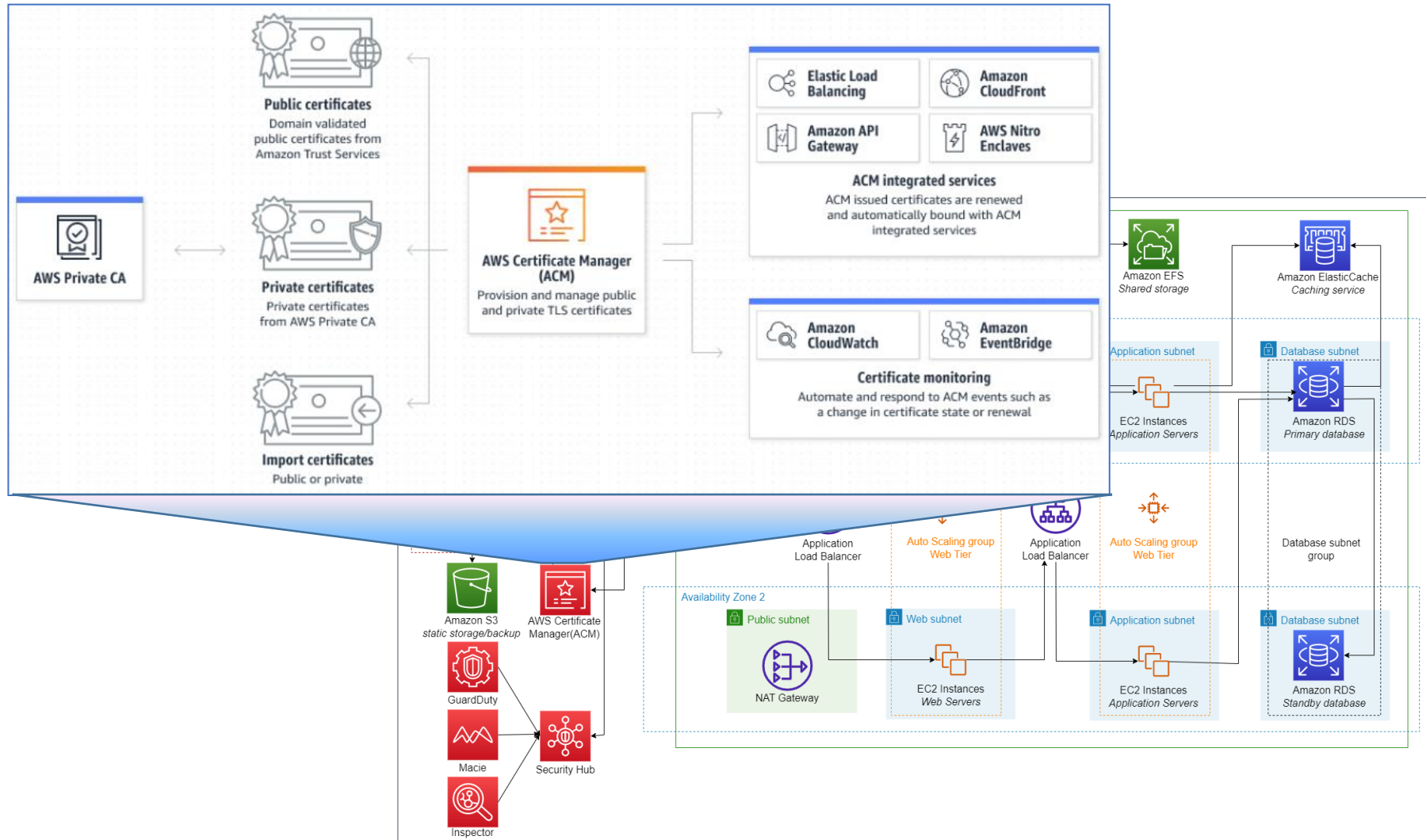
- A회사는 on-prem 에서 3-tier web app 을 운영하고 있다.
 - 신규 유저의 유입이 많아지면서 기존 서버들로 트래픽을 처리할 수 없어 AWS로의 이관을 고려 중이다.
 - 고가용성은 물론 A회사의 보안 요구사항을 충족할 수 있는 아키텍처를 구성하라.
-
- 보안 요구사항
 - 인증서관리의 자동화 (현재 서버에서 갱신 주기에 맞춰 수동 작업 중)
 - DDoS 방어
 - 웹 취약점 공격 대응
 - Outbound network traffic 필터링 기능 필요 ex. 비업무사이트에 대한 인터넷 접근 차단
 - Malware Protection
 - SIEM(Security Information and Event Management) 구성 (optional)

3. Design Architecture : Multi-tier Web App



3. Design Architecture : 인증서관리의 자동화

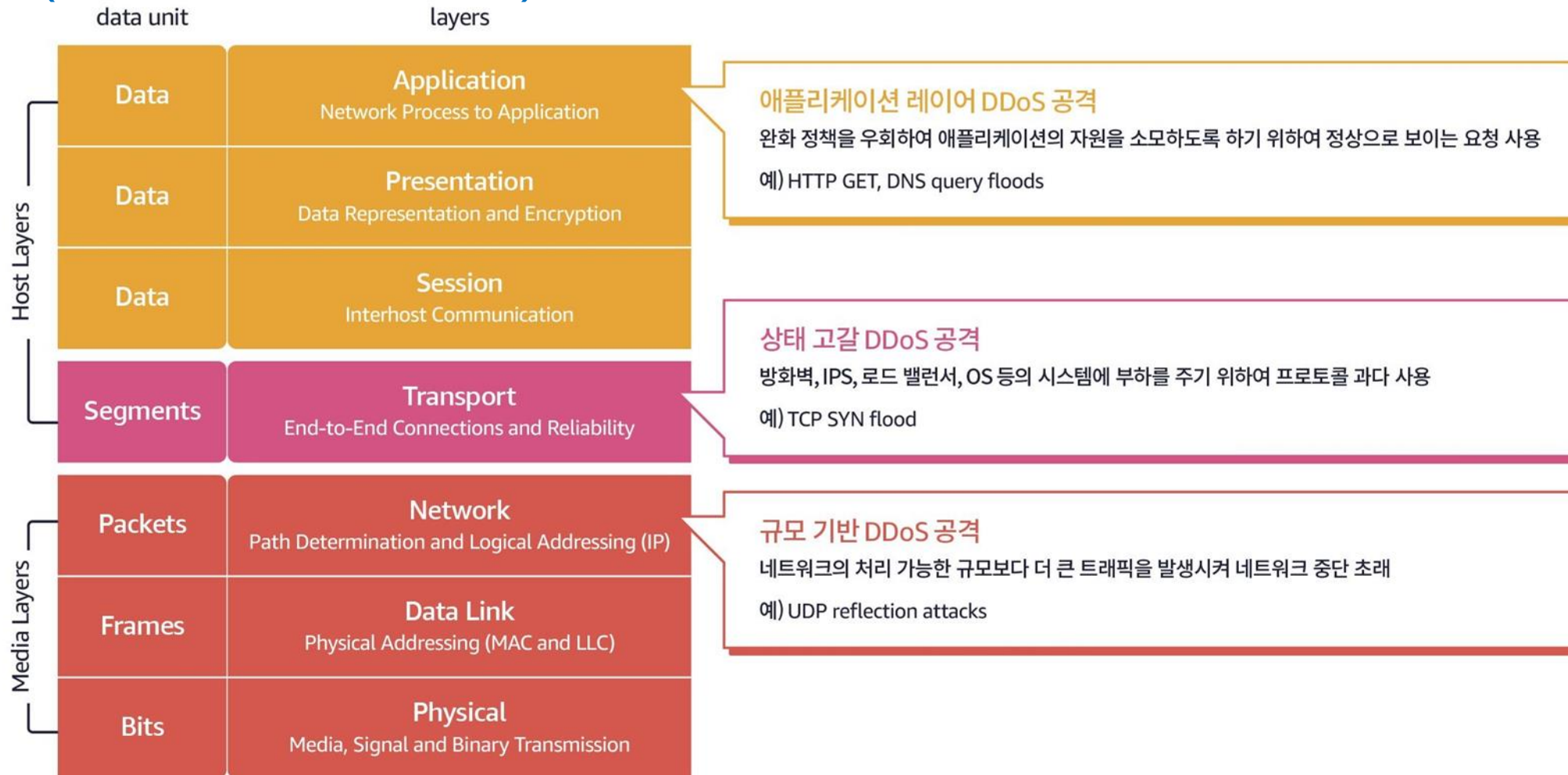
ACM(AWS Certificate Manager)는 SSL/TLS 인증서를 손쉽게 프로비저닝, 관리 및 배포할 수 있도록 지원하는 서비스
ACM은 SSL/TLS 인증서를 구매, 업로드 및 갱신하는 데 드는 시간 소모적인 수동 프로세스를 대신 처리



3. Design Architecture : DDoS 방어

DDoS는 여러 대의 공격자에 의해서 시스템을 악의적 트래픽으로 공격한 뒤 해당 시스템의 자원을 부족하게 만드는 공격
DDoS 공격의 경우, 공격 벡터로 불리는 다양한 형태의 공격 유형을 가지고 있음

DDoS(Distributed Denial of Service) 공격 패턴 및 현황



3. Design Architecture : DDoS 방어

DDoS 방어를 위해 제공되는 AWS 서비스

AWS Shield

AWS Shield **Standard**

- 일반적인 레이어 3/4 공격으로부터 보호 (SYN/UDP Floods, Reflection Attacks, 등등)
- 자동으로 감지 및 완화
- AWS 서비스에 빌트인
- 모든 사용자에게 무료로 제공

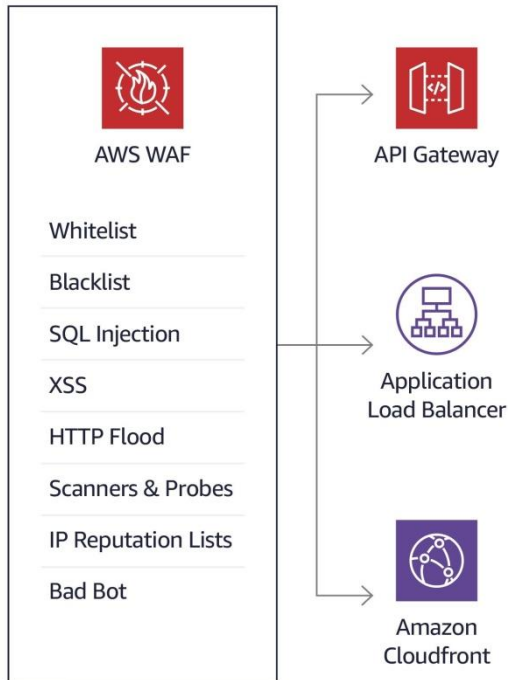
AWS Shield **Advanced**

- 상시 모니터링 및 감지
- 추가적인 레이어 3/4/7 공격으로부터 보호
- 공격 지표, 경고 및 리포트
- 24X7로 DDoS Response Team (DRT) 지원
- 추가 비용 없이 AWS WAF 사용
- DDoS 대응에 사용된 추가 자원 비용 보존
- 1년 약정 방식

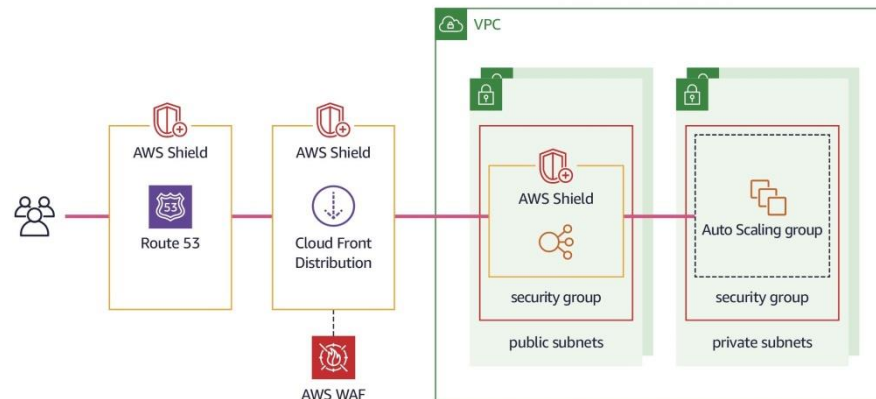
AWS Global Accelerator



AWS Web Application Firewall

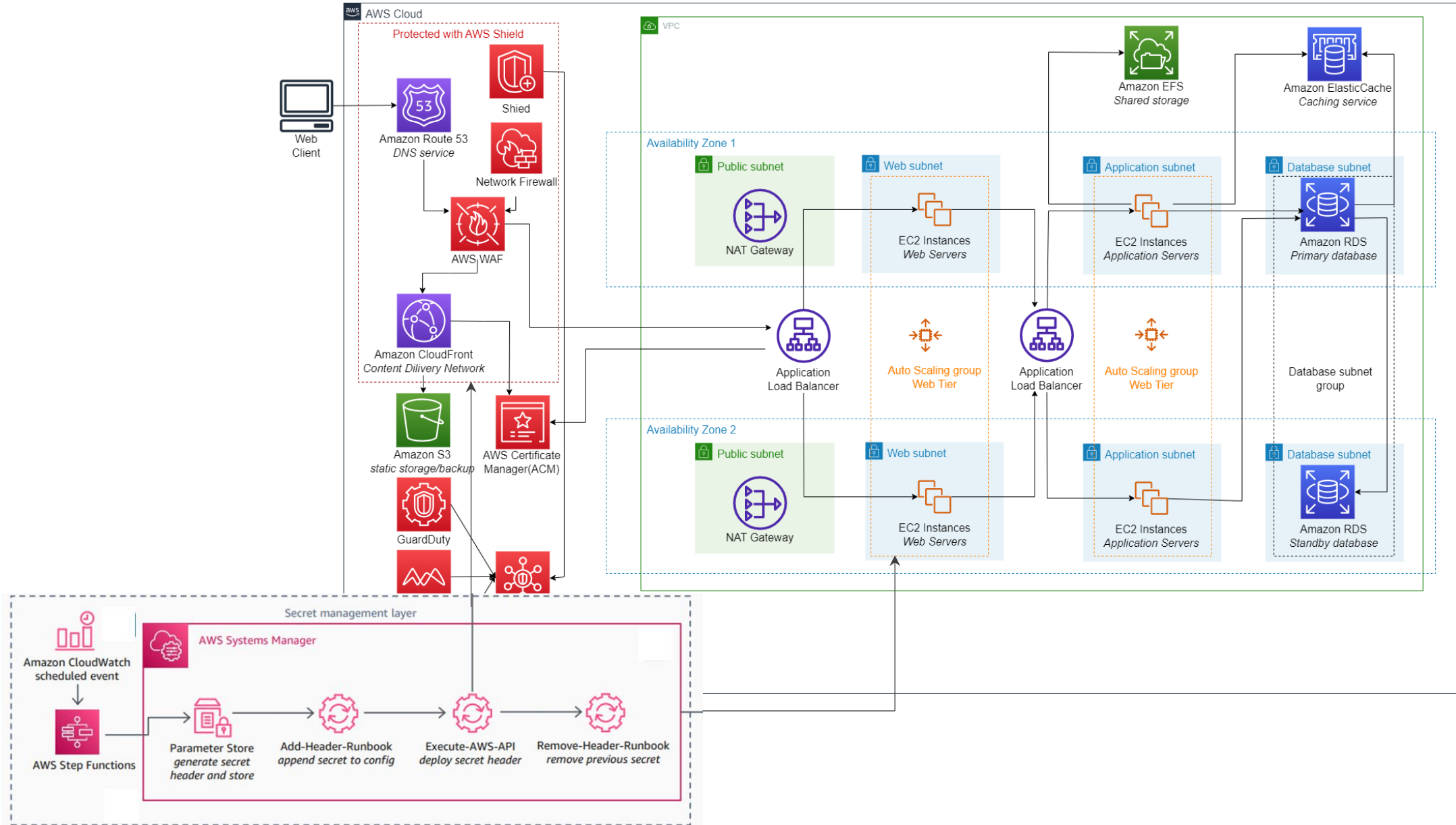


DDoS 방어를 위한 아키텍처



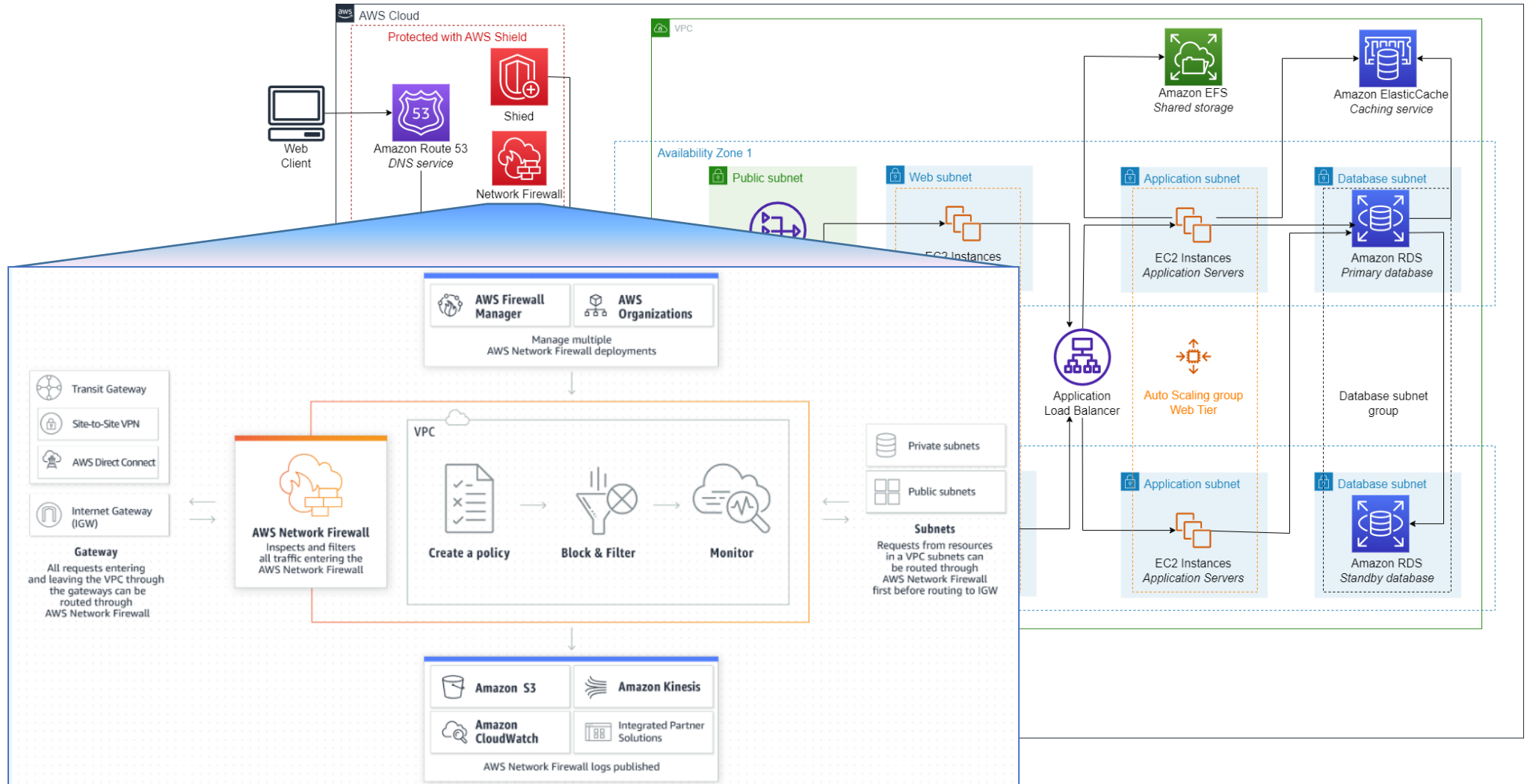
- 확장이 가능한 서비스 구조를 준비
- 공개적으로 노출되는 서비스 엔드포인트를 최소화
- 웹 애플리케이션 방화벽을 사용
- 부하 상황에서의 서비스 안정성을 미리 검증
- 서비스의 동작을 모니터링하고 경보를 등록
- DDoS가 발생했을 때 어떻게 대응할지에 대한 런북을 작성
- 클라이언트에서 접근 엔드포인트 정보를 DNS 를 사용

3. Design Architecture : 웹 취약점 공격 대응



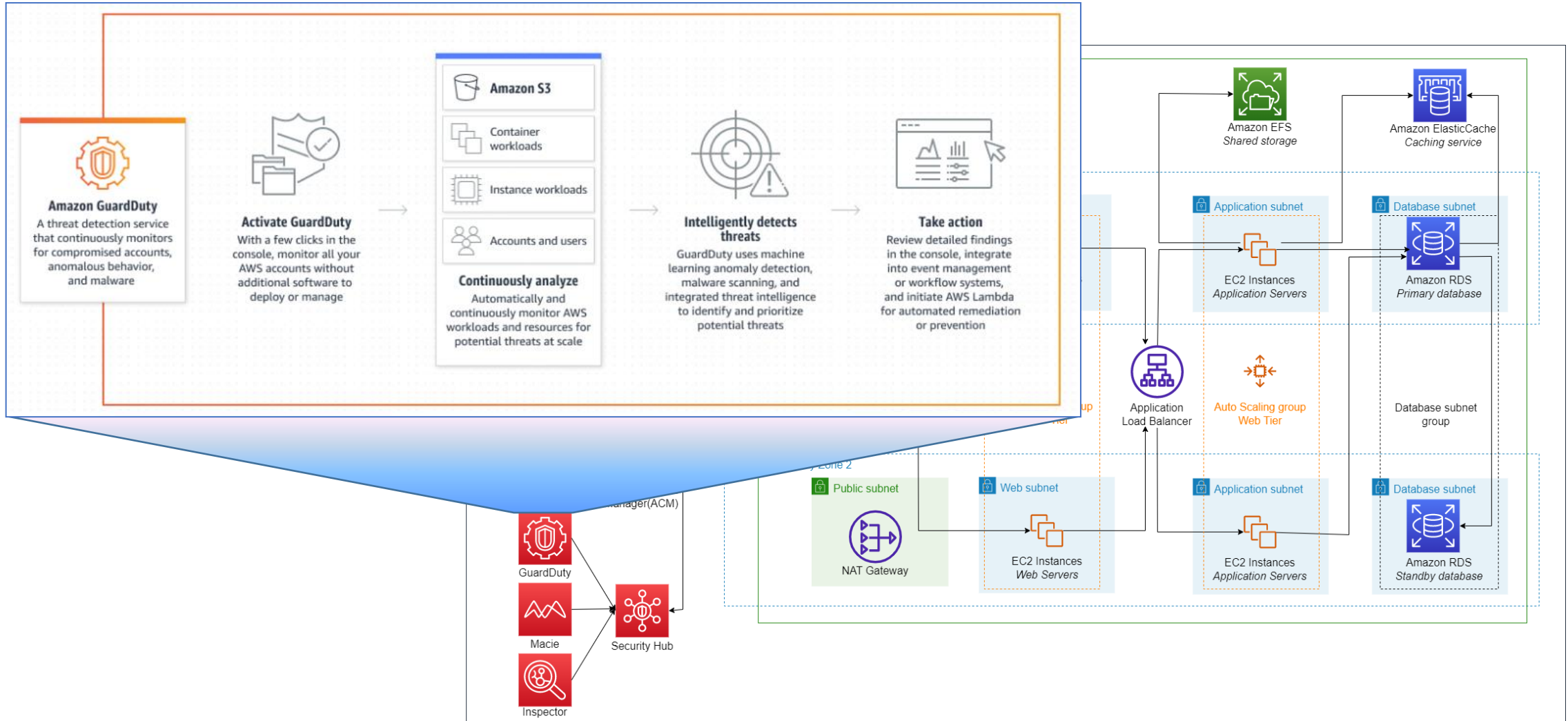
3. Design Architecture : Outbound network traffic 필터링 기능

AWS Network Firewall의 유연한 규칙 엔진은 악성 활동의 확산을 방지하기 위해 아웃바운드 SMB(서버 메시지 블록) 요청을 차단하는 등 네트워크 트래픽을 세밀하게 제어할 수 있는 방화벽 규칙을 정의할 수 있게 함



3. Design Architecture : Malware Protection

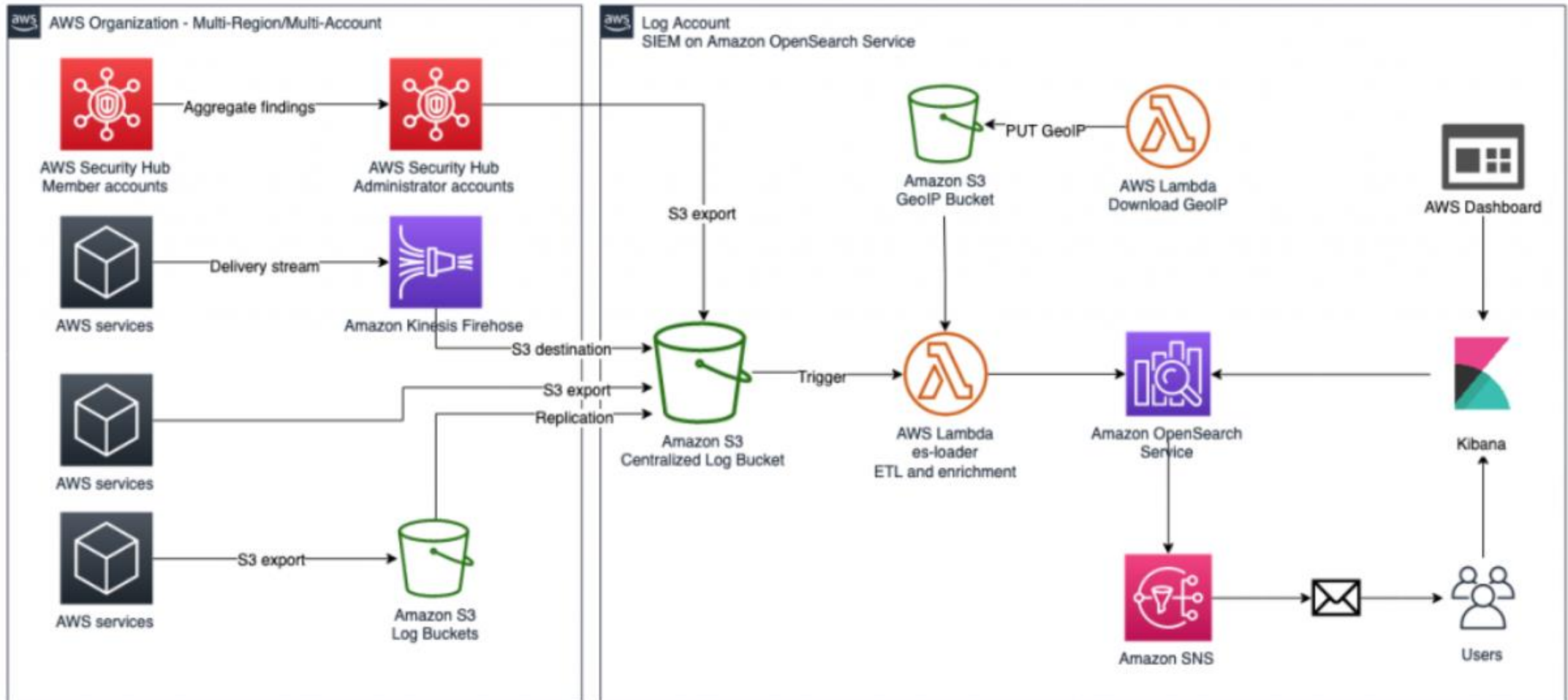
GuardDuty Malware Protection을 사용하여 EC2 또는 컨테이너 워크로드에 상주하는 악성 파일을 탐지할 수 있음
위협이 감지되면 GuardDuty Malware Protection은 보안 결과를 Security Hub, EventBridge, Detective에 자동으로 전송



3. Design Architecture : SIEM(Security Information and Event Management)

Amazon OpenSearch Service를 SIEM으로 사용하고, Security Hub를 통합

Amazon OpenSearch Service는 다른 AWS 서비스들의 로그를 직접 로드할 수도 있고, Kinesis를 연동하거나, Beats-Logstash와 에이전트 기반으로 로그를 수집하고 Kibana를 통한 시각화를 구현하는 방식과 같이 다양하게 확장될 수 있음

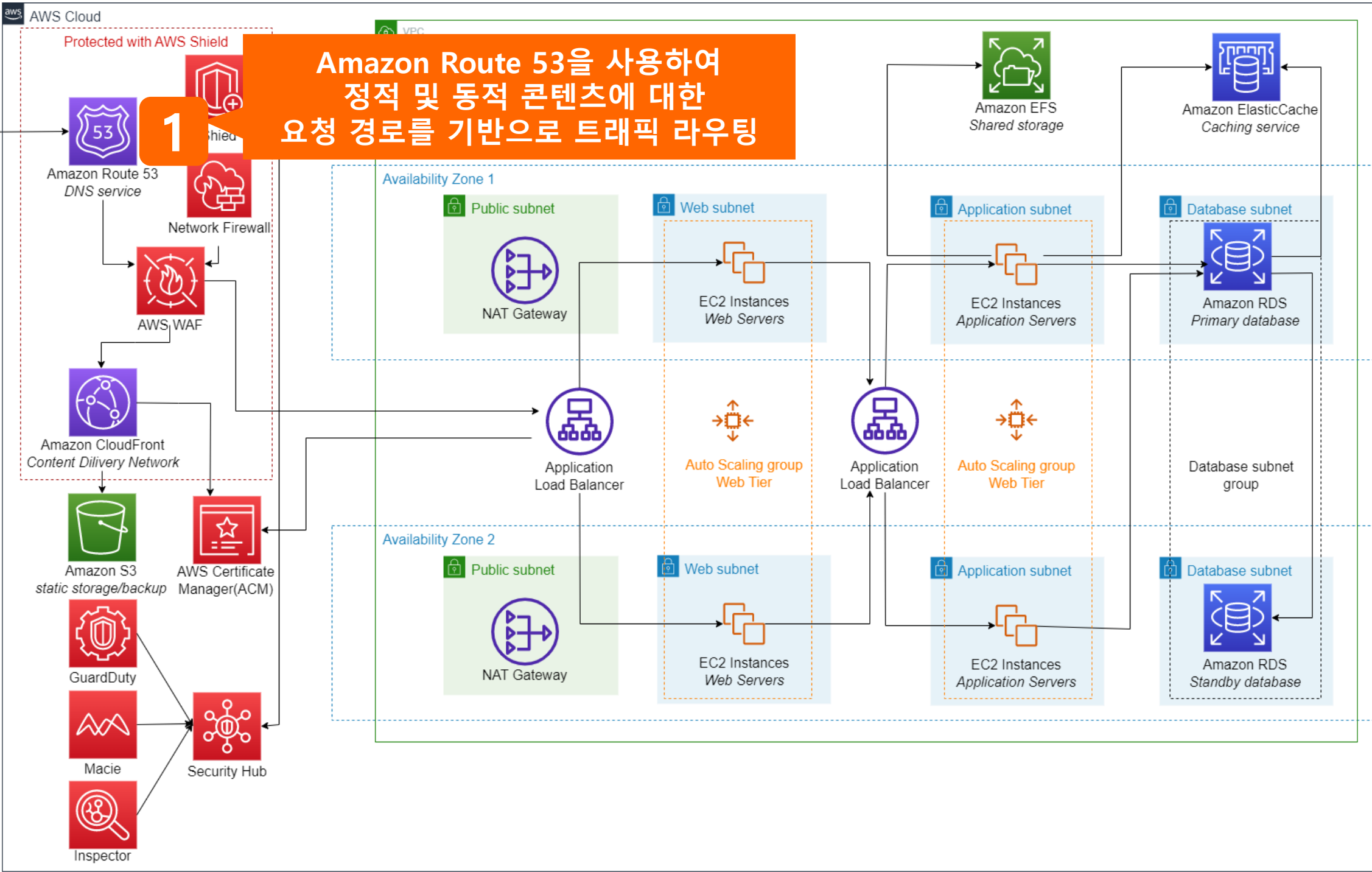


3. Design Architecture

아키텍처 Pipeline Flow 분석



Web Client

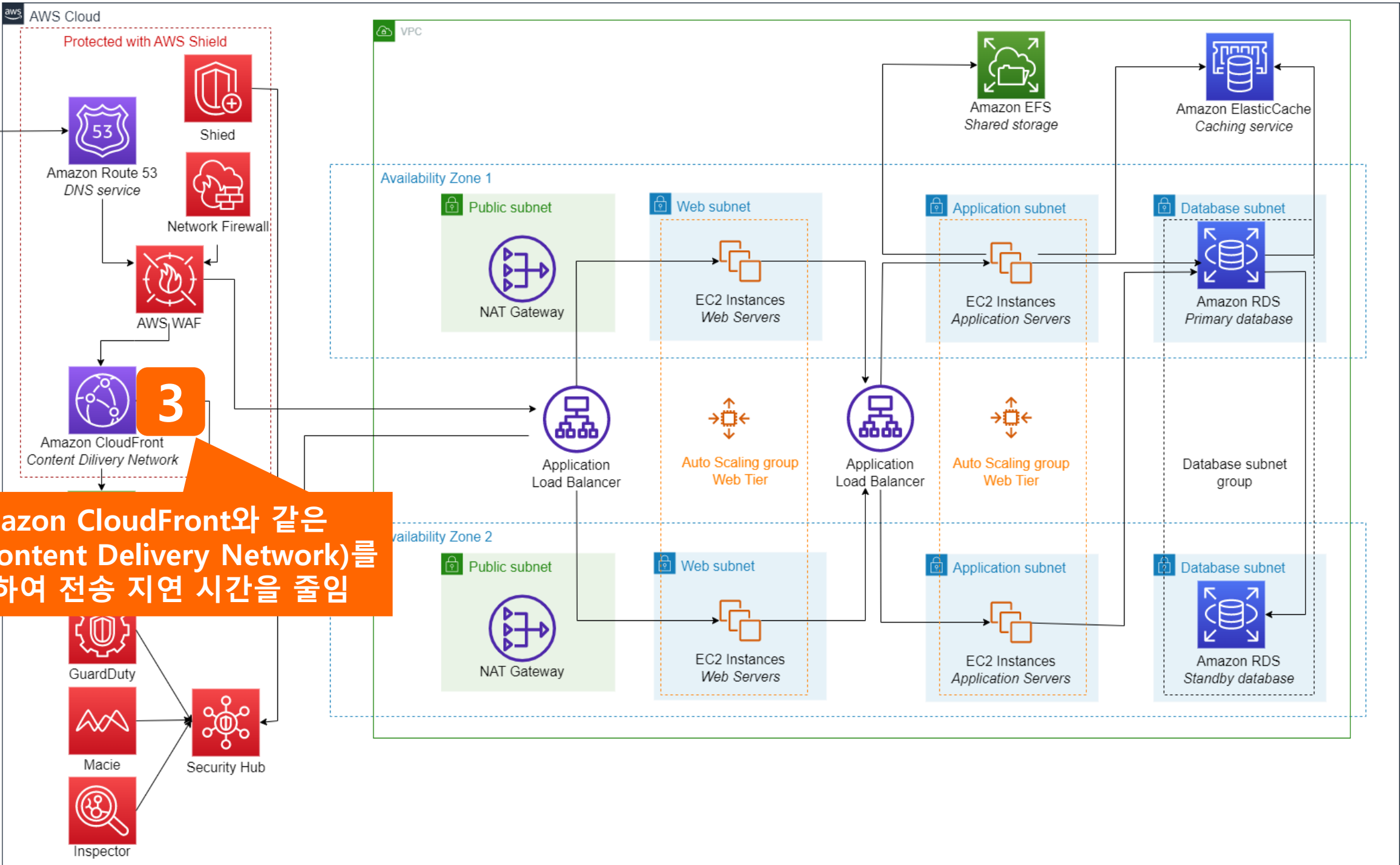


Amazon Route 53을 사용하여
정적 및 동적 콘텐츠에 대한
요청 경로를 기반으로 트래픽 라우팅

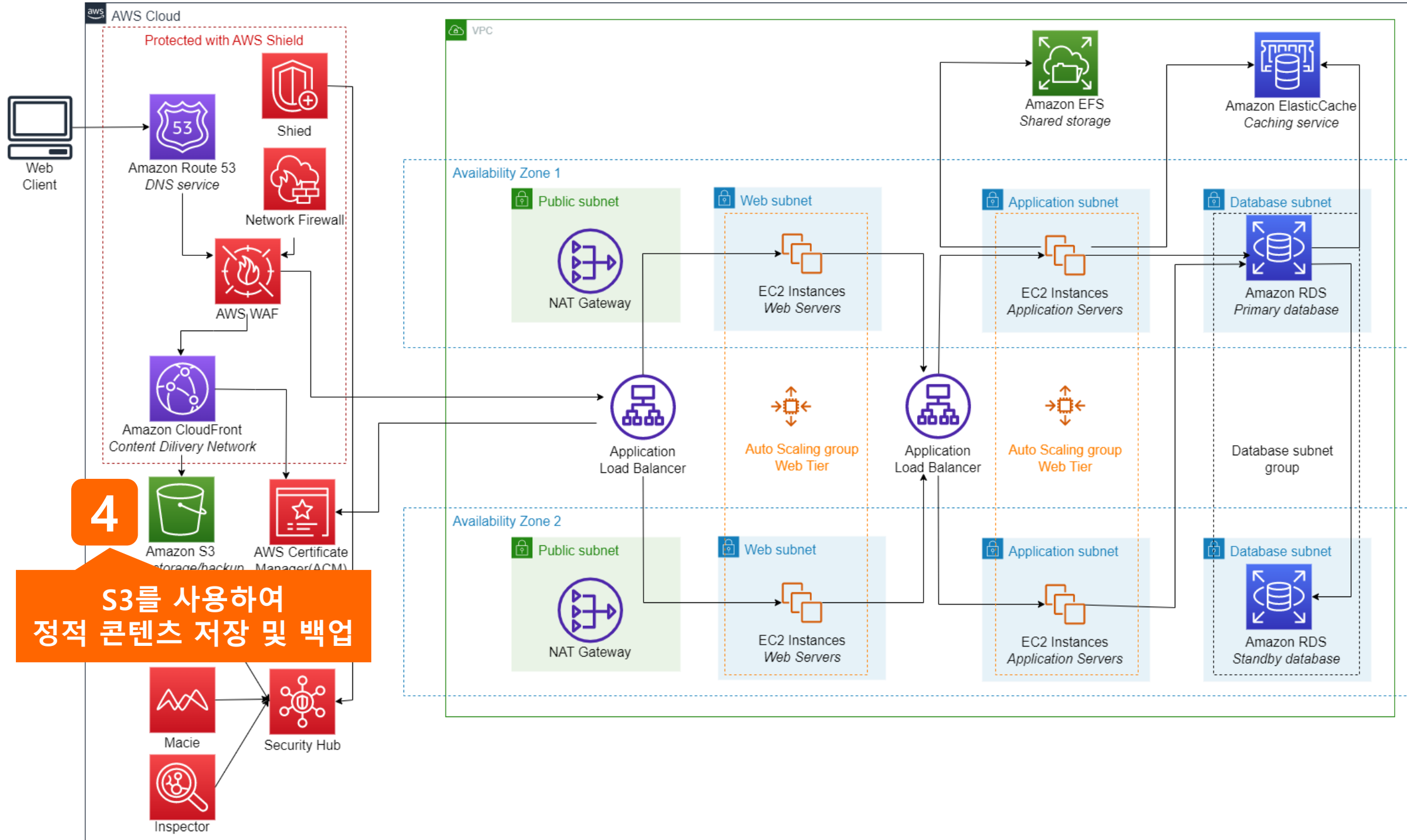


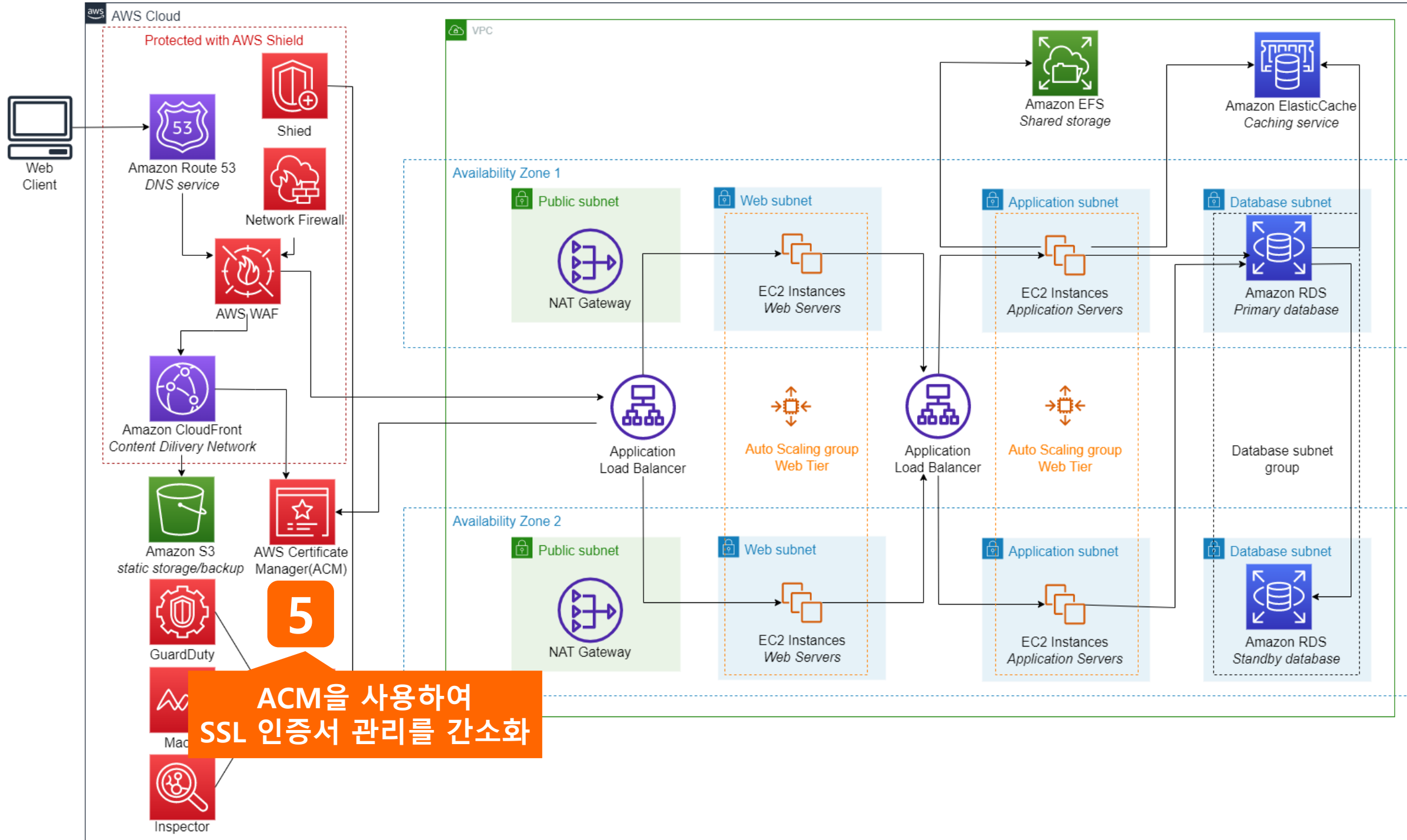


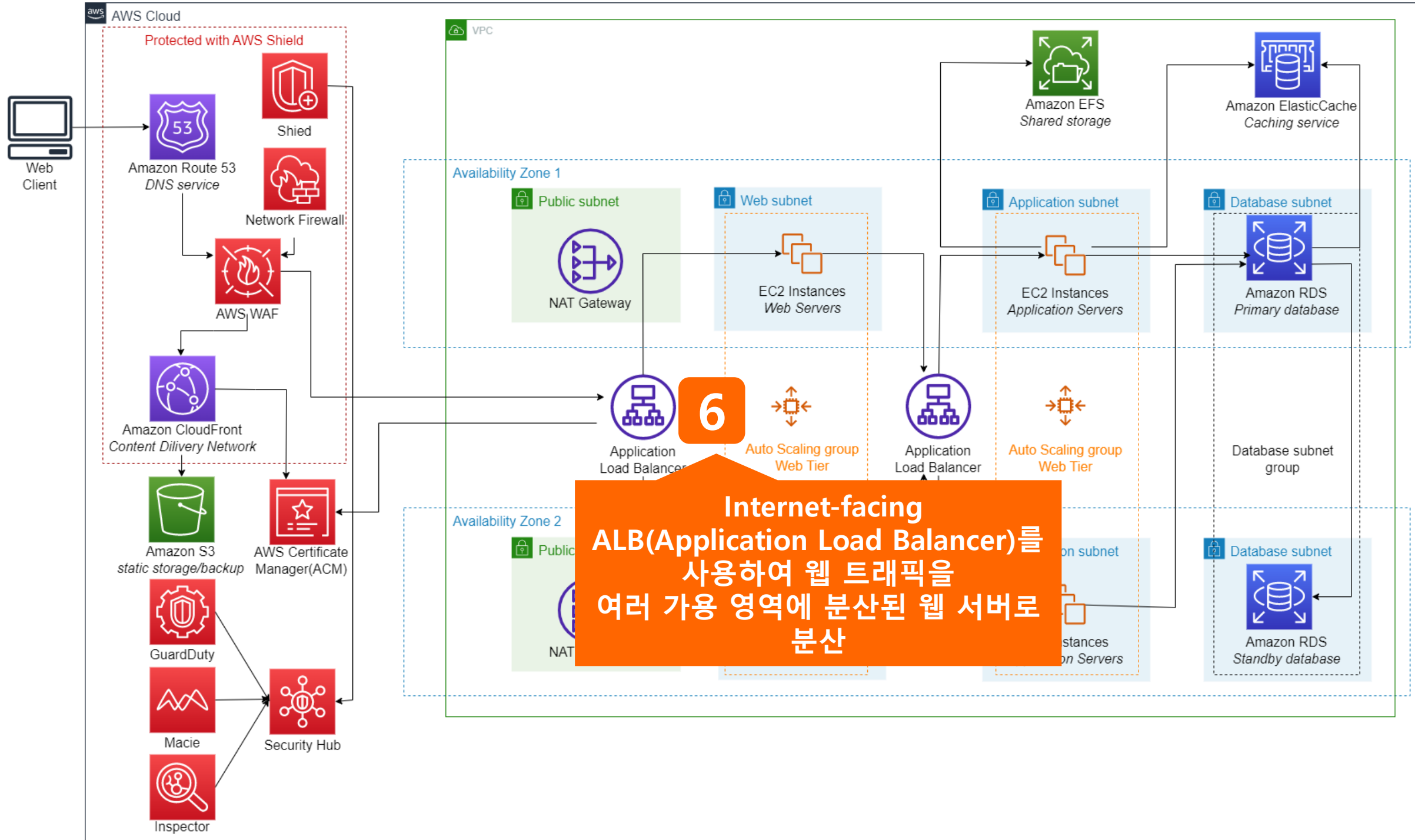
Web Client



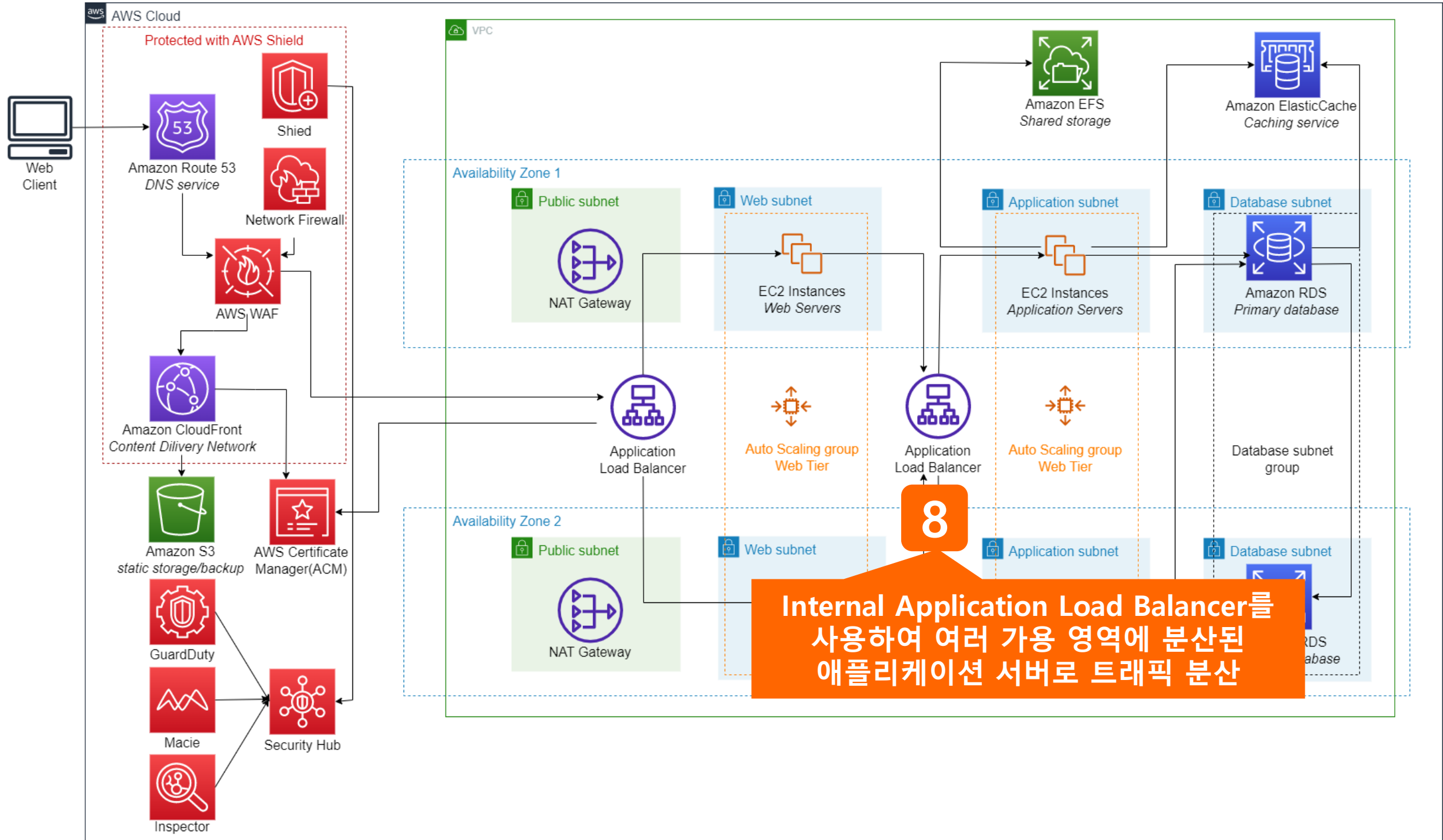
Amazon CloudFront와 같은
CDN(Content Delivery Network)를
사용하여 전송 지연 시간을 줄임

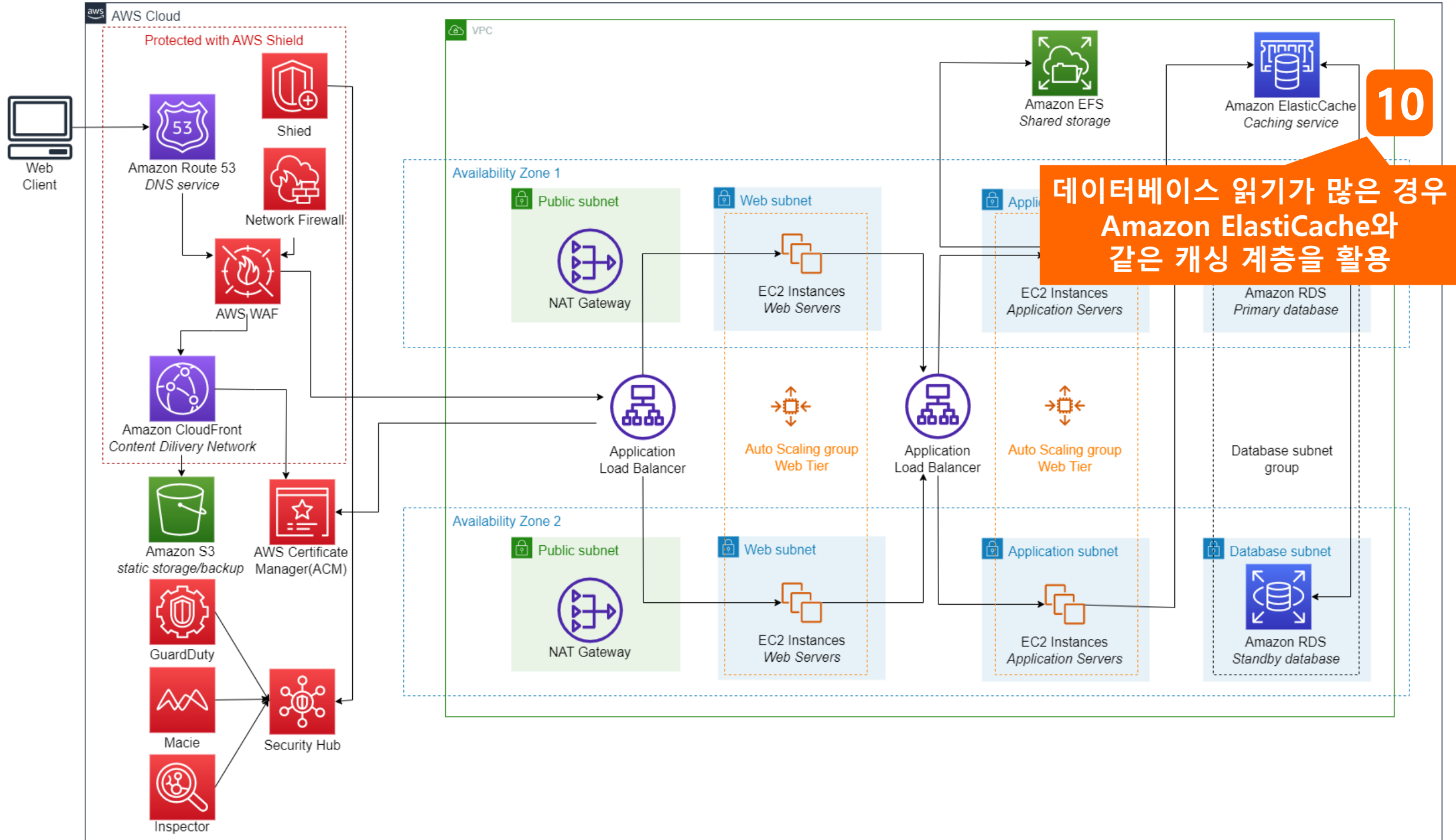


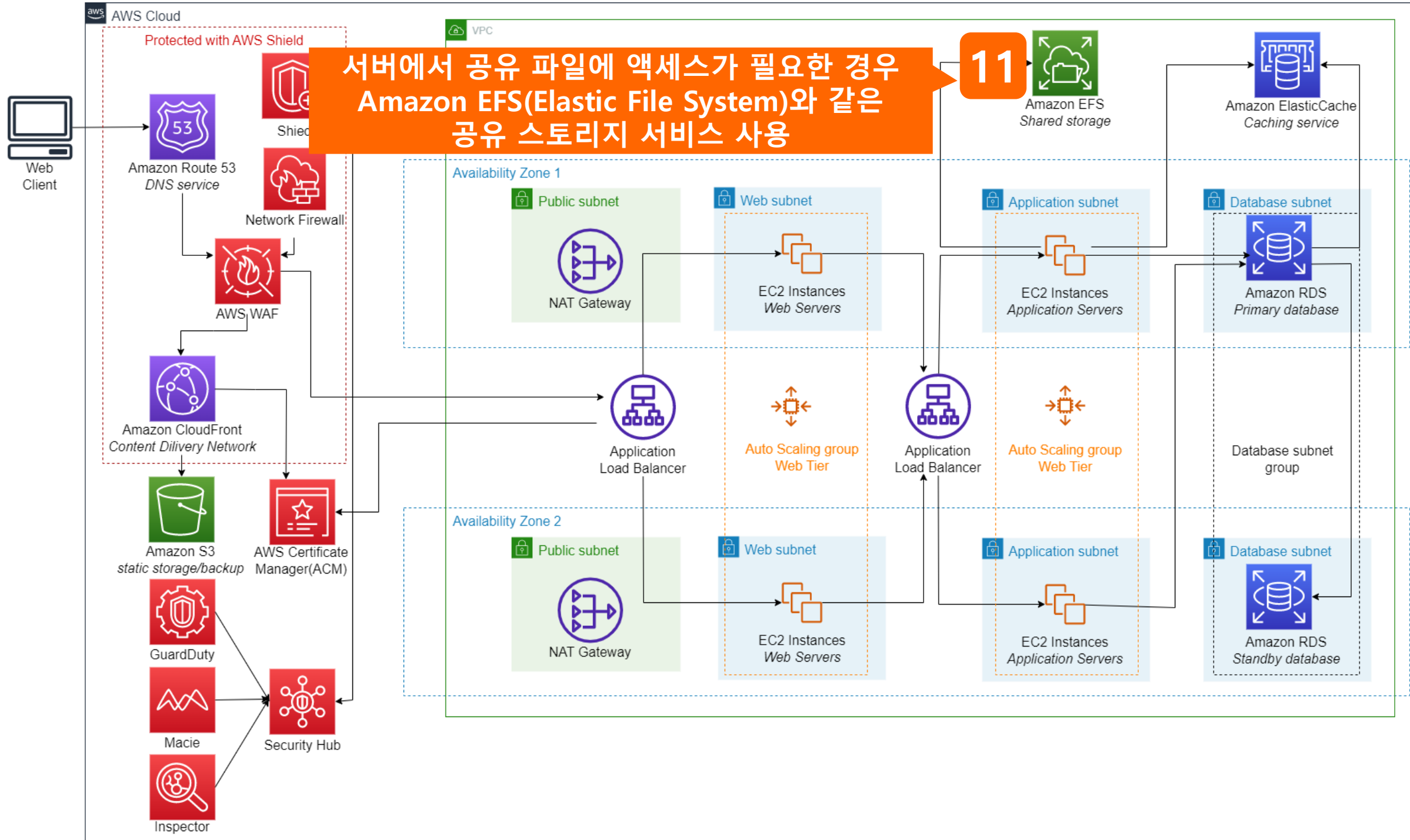












Thank you