AWS Web Application Deployment with CloudFront and WAF - Walkthrough Summary

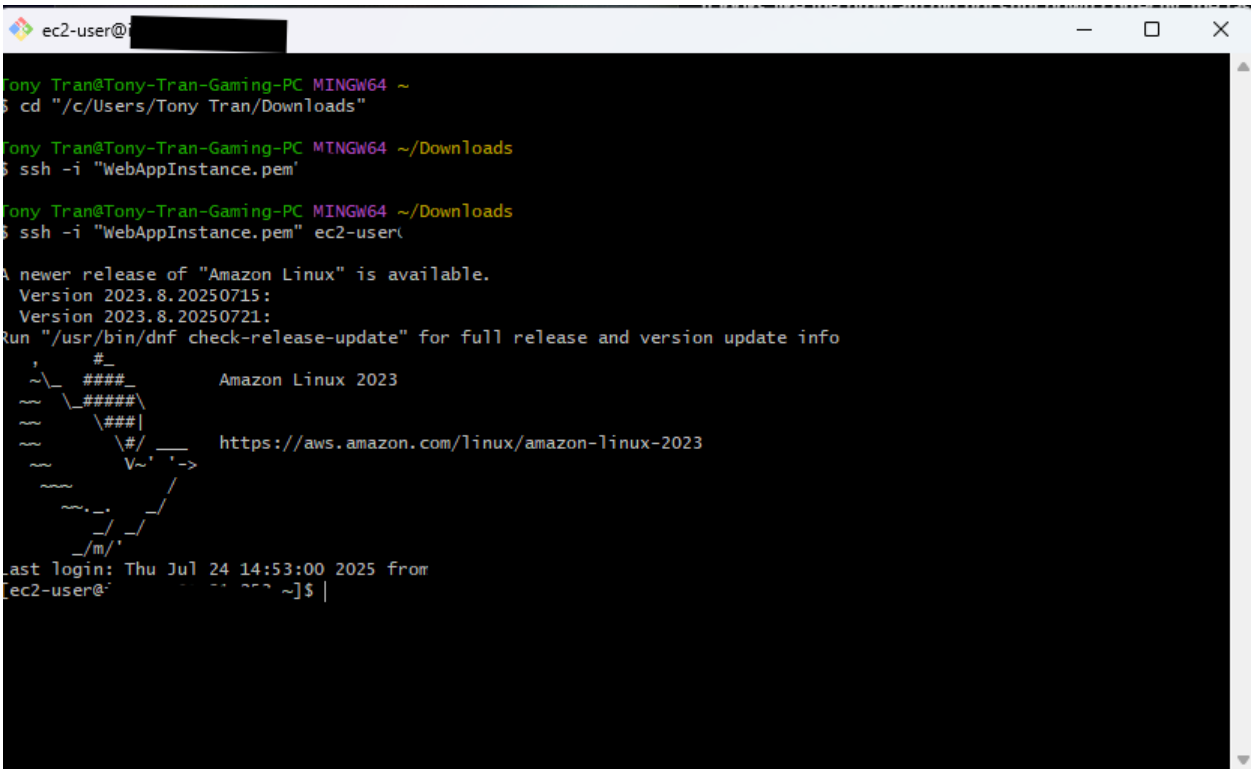
Author: Tony Tran
Project Goal: Deploy a secure web application using EC2, CloudFront, and AWS WAF with OWASP protection.

---

1. EC2 Setup

- **Launch EC2 instance** using Amazon Linux 2023.
- Open ports 22 (SSH) and 80 (HTTP) in the **Security Group**.
- Connect to the EC2 instance using `.pem` key with:

```
ssh -i "WebAppInstance.pem" ec2-user@<EC2 Public DNS>
```



---

2. Install and Configure Web Server

- Install Apache web server:

```
sudo yum update -y
sudo yum install httpd -y
```

```
sudo systemctl start httpd
sudo systemctl enable httpd
```

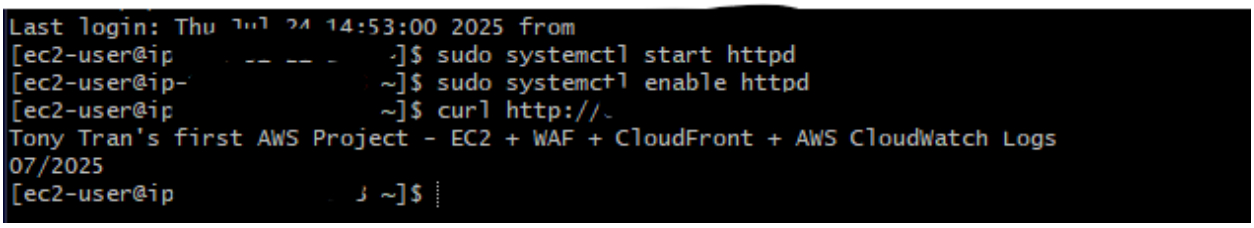
- Modify `index.html`:

```
echo "Hello from Tony's secure AWS web app!" | sudo tee /var/www/html/i
ndex.html
```

- Verify via EC2 public DNS:

```
curl http://<EC2 Public DNS>
```

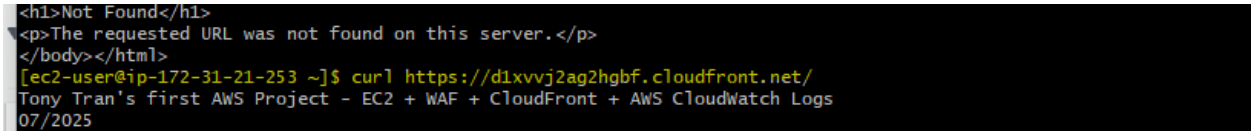
Output:



---

3. CloudFront Configuration

- Create a **CloudFront distribution**:
  - **Origin:** EC2 Public DNS (e.g., `ec2-34-228-71-196.compute-1.amazonaws.com`)
  - **Origin Protocol Policy:** HTTP only
  - **Viewer Protocol Policy:** Redirect HTTP to HTTPS
  - Enable **WAF security protections**
- Wait until **Status = Deployed**
- **CloudFront URL: https://d1xvvj2ag2hgbf.cloudfront.net/**



---

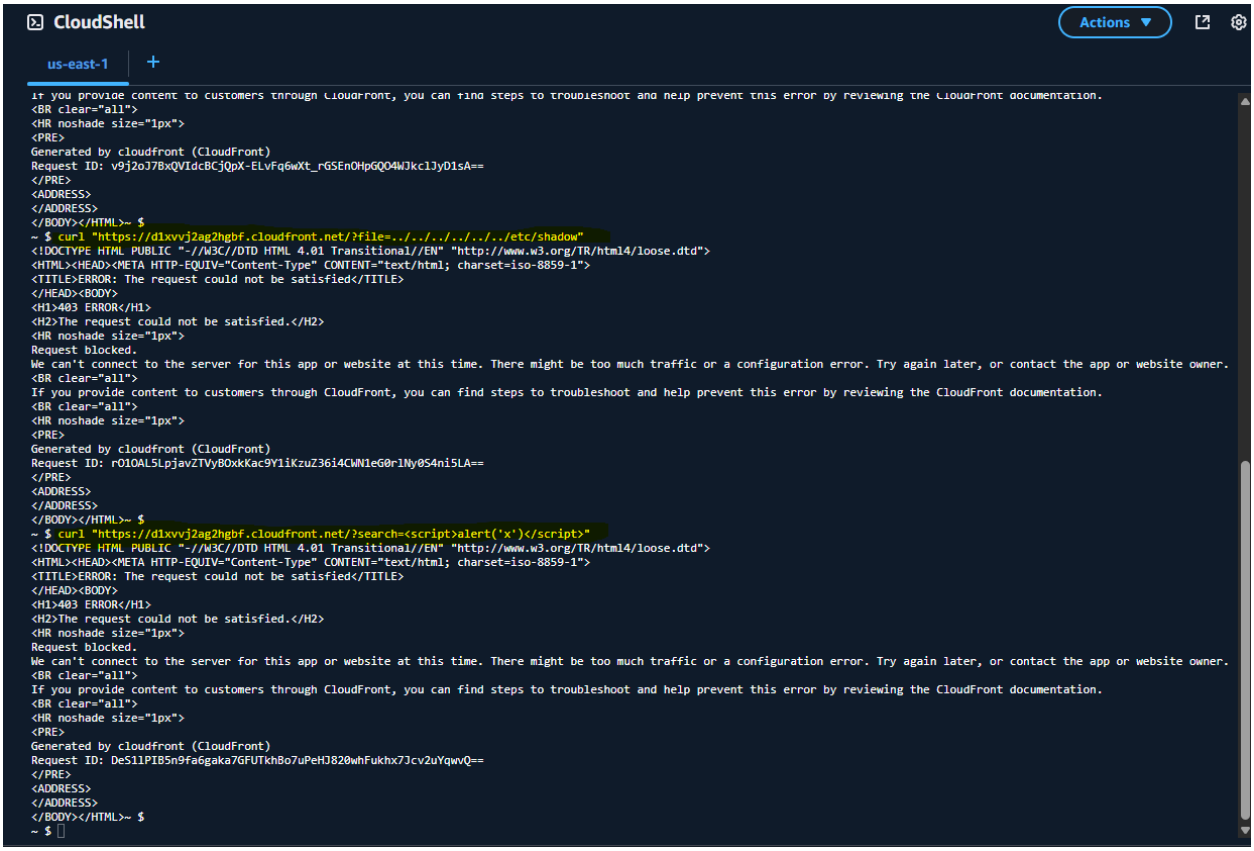
4. AWS WAF Setup

- Create a **Web ACL** in AWS WAF:
  - Scope: CloudFront
  - Associate with the CloudFront distribution
  - Add managed rule group: **AWSManagedRulesCommonRuleSet** (OWASP protection)

---

5. Test WAF Protection

- Test payloads via CloudFront URL:
- `# XSS Payload`
- `curl "https://d1xvvj2ag2hgbf.cloudfront.net/?search=<script>alert('x')</script>"`
- `# SQL Injection Payload`
- `curl "https://d1xvvj2ag2hgbf.cloudfront.net/?id=1' OR '1'='1"`
- `# Path Traversal Payload`
- `curl "https://d1xvvj2ag2hgbf.cloudfront.net/?search=../../etc/passwd"`
- `# Admin Enumeration Payload`
- `curl "https://d1xvvj2ag2hgbf.cloudfront.net/?admin=true"`
- `# Command Injection (for completeness)`
- `curl "https://d1xvvj2ag2hgbf.cloudfront.net/?cmd=ls%20-la"`
- `# Local File Inclusion (LFI) variant`
- `curl https://d1xvvj2ag2hgbf.cloudfront.net/?file=../../../../../../etc/shadow`
- All return:



`403 ERROR: The request could not be satisfied. Request blocked.`

---

6. Troubleshooting Notes

- If CloudFront returns `504 Gateway Timeout`:
  - Confirm EC2 is running
  - Ensure EC2 allows traffic on port 80
  - Check CloudFront origin settings and wait for deployment

---

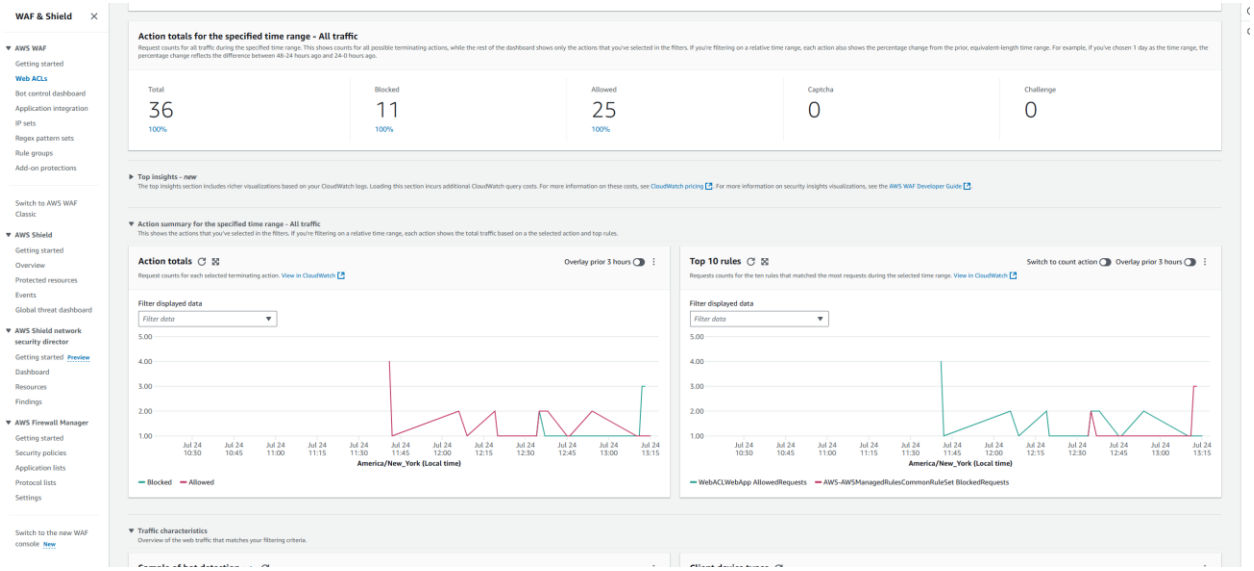
7. Key Services Used

- **EC2:** Hosts the web application
- **CloudFront:** CDN layer with performance and caching
- **AWS WAF:** Blocks malicious input using managed rules
- **Security Groups:** Controls inbound/outbound EC2 traffic

---

8. Outcome

- Successfully deployed a public web app with **CloudFront CDN** and **WAF protection**.
- Verified defense against **OWASP Top 10** payloads.



Traffic characteristics

Overview of the web traffic that matches your filtering criteria.

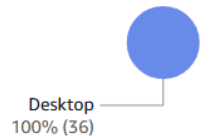
Sample of bot detection Info

Counts for non-bot activity, and for verified and unverified bot activity. This includes only bots labeled by Bot Control for common bots View in CloudWatch

Bots:Unverified Non-bots

Client device types

The device types of the clients that sent the requests, obtained from the web request's user-agent header. View in CloudWatch



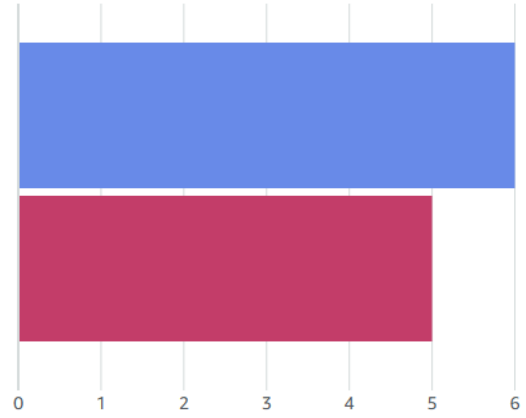
Desktop

Attack types Info

The types of attacks identified in the requests. View in CloudWatch

Filter displayed data

Filter data



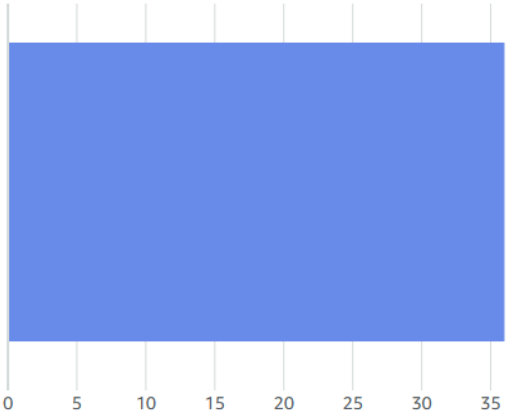
XSS GenericLFI

Top 10 countries Info

The ten countries that sent the most requests. View in CloudWatch

Filter displayed data

Filter data



United States

Managed rule groups