**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>
* A computer screen shot of a black screen

  AI-generated content may be incorrect.

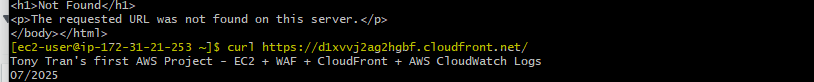
### **2. Install and Configure Web Server**

* 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/index.html
* Verify via EC2 public DNS:
* curl http://<EC2 Public DNS>
* **Output:**  
    
  A screen shot of a computer

  AI-generated content may be incorrect.

### **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/**](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:

A screenshot of a computer program

AI-generated content may be incorrect.

* 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.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

This document can be presented to an interviewer or included in a portfolio to demonstrate hands-on cloud security implementation and web application deployment experience.