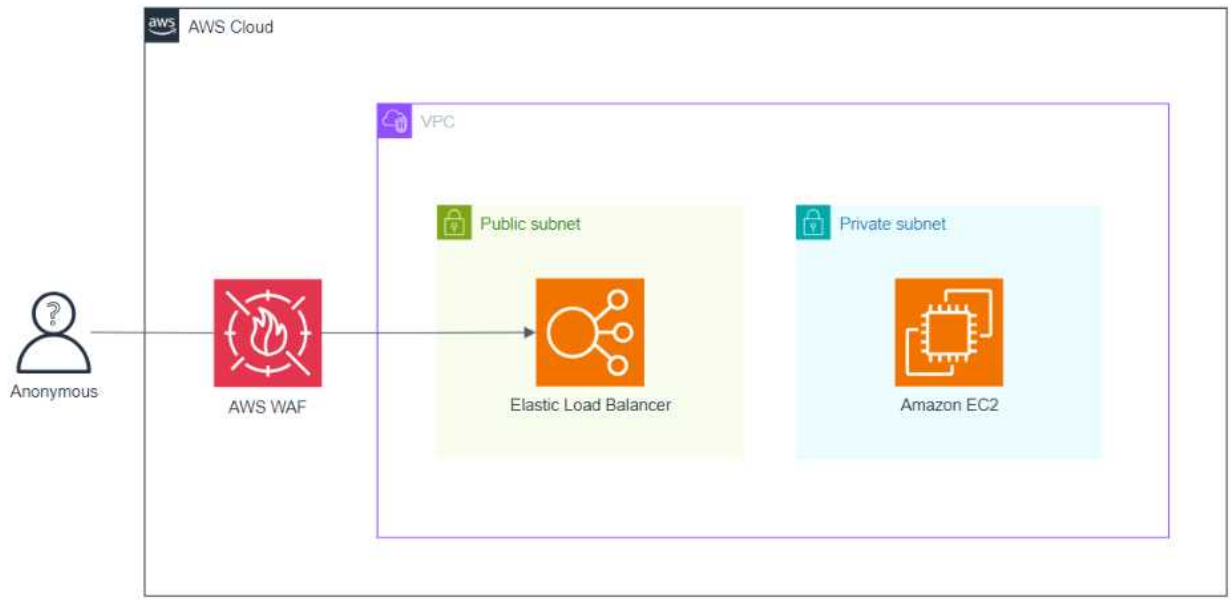
2024년도 전국기능경기대회 과제

직종명	클라우드컴퓨팅	과제명	Web Security	과제번호	제2과제
경기시간	4시간	비번호		심사위원 확 인	(인)

1. 요구사항

많은 웹 애플리케이션에서 사용자 인증을 위해 JWT(JSON Web Token)를 활용하고 있습니다. JWT는 세션 기반 인증의 대안으로 널리 사용되는 토큰 기반 인증 방식이며 사용자의 신원을 확인하고 액세스 권한을 부여하는 데 효과적입니다. 하지만 JWT의 구현 과정에서 발생할 수 있는 여러 취약점들로 인해 보안 위험에 노출될 수 있습니다. 그 중 JWT 토큰의 서명 알고리즘 타입을 "none"으로 변경 후 Signiture를 제거하면 페이로드를 변조하여 유저의 권한을 가로챌 수 있는 공격인 None Algorithm Attack을 AWS WAF 서비스를 통해 JWT 토큰의 유효성을 검사하고, 유효한 토큰만 허용하도록 구성해야 합니다.

다이어그램



Software Stack

AWS	개발언어/프레임워크
- VPC - EC2 - ELB - WAF	- Python / Flask - Docker

2. 선수 유의사항

1) 기계 및 공구 등의 사용 시 안전에 유의하시고, 필요 시 안전장비 및 복장 등을 착용하여 사고를 예방하여 주시기 바랍니다.
2) 작업 중 화상, 감전, 찰과상 등 안전사고 예방에 유의하시고, 공구나 작업도구 사용 시 안전보호구 착용 등 안전수칙을 준수하시기 바랍니다.
3) 작업 중 공구의 사용에 주의하고, 안전수칙을 준수하여 사고를 예방하여 주시기 바랍니다.
4) 경기 시작 전 가벼운 스트레칭 등으로 긴장을 풀어주시고, 작업도구의 사용 시 안전에 주의하십시오.
5) 선수의 계정에는 비용제한이 존재하며, 이보다 더 높게 과금될 시 계정 사용이 불가능할 수 있습니다.
6) 문제에 제시된 괄호박스 는 변수를 뜻함으로 선수가 적절히 변경하여 사용해야 합니다.
7) EC2 인스턴스의 TCP 80/443 outbound는 anyopen하여 사용할 수 있도록 합니다.
8) 과제의 Bastion 서버에서 대부분의 채점이 이루어짐으로 인스턴스를 생성하지 않았거나 종료된 상태면 채점이 불가능하니 각별히 주의하도록 합니다.
9) 과제 종료 시 진행 중인 테스트를 모두 종료하여 서버에 부하가 발생 하지 않도록 합니다.
10) 별도 언급이 없는 경우, ap-northeast-2 리전에 리소스를 생성하도록 합니다.
11) 1페이지의 다이어그램은 구성을 추상적으로 표현한 그림으로, 세부적인 구성은 아래의 요구사항을 만족시킬 수 있도록 합니다. (ex. 서브넷이 2개 이상 존재할 수 있습니다.)
12) 모든 리소스의 이름, 태그, 변수는 대소문자를 구분합니다.
13) 문제에서 주어지지 않는 값들은 AWS Well-Architected Framework 6 pillars를 기준으로 적절한 값을 설정해야 합니다.
14) 불필요한 리소스를 생성한 경우, 감점의 요인이 될 수 있습니다. (e.g. VPC 추가 생성)

3. 네트워크 구성

클라우드 인프라에 대해 네트워크 레벨의 격리 및 분리가 가능하도록 아래 요구사항을 참고하여 VPC를 구성 합니다. 서브넷 이름 뒤의 알파벳은 Availability Zone을 의미합니다.

VPC 정보

- VPC CIDR : 10.0.0.0/16
- VPC Tag : Name=wsi-vpc
- Internet G/W Tag : Name=wsi-igw

Public subnet A 정보

- CIDR : 10.0.0.0/24
- Tag : Name=wsi-public-subnet-a
- 외부 통신 : Internet G/W 를 구성하여 인터넷 접근
- Route table Tag : Name=wsi-public-rtb

Public subnet B 정보

- CIDR : 10.0.1.0/24
- Tag : Name=wsi-public-subnet-b
- 외부 통신 : Internet G/W 를 구성하여 인터넷 접근
- Route table Tag : Name=wsi-public-rtb

Private subnet A 정보

- CIDR :10.0.2.0/24
- Tag : Name=wsi-private-subnet-a
- 외부 통신 : NAT G/W를 구성하여 인터넷 접근이 가능하도록 구성
- Route table Tag : Name=wsi-private-rtb-a
- NAT G/W Tag : Name=wsi-natgw-a

Private subnet B 정보

- CIDR : 10.0.3.0/24
- Tag : Name=wsi-private-subnet-b
- 외부 통신 : NAT G/W를 구성하여 인터넷 접근이 가능하도록 구성
- Route table Tag : Name=wsi-private-rtb-b
- NAT G/W Tag : Name=wsi-natgw-b

4. Bastion 서버

EC2를 활용해 Bastion 서버를 구성합니다. Bastion 서버의 접근을 위해서 SSH 프로토콜을 사용합니다. Bastion 서버는 외부에서 SSH 프로토콜만을 허용하도록 Security Group을 구성하세요. 또한 서버가 재시작 되어도 Public IP가 변경되어서는 안됩니다.

Bastion 서버는 채점을 위해서 사용됩니다. 잘 못 구성하였을 경우 모든 채점 항목에서 불이익을 받을 수 있으니 주의합니다.

- Instance type : t4g.small
- Subnet : Public subnet A
- 설치 패키지 : awscli, jq, curl
- Tag : Name=wsi-bastion
- Security group name : wsi-sg-bastion (명시된 Security Group 하나만을 사용합니다.)
- EC2 IAM Role : PowerUserAccess policy 권한을갖도록 설정해야 합니다. role 이름은 wsi-role-bastion으로 설정합니다.

5. 웹 어플리케이션

해당 과제에서 배포하여 사용할 웹 어플리케이션 입니다. Python Flask를 통해 개발되었습니다.

Path	Method	Description	Request Sample
/v1/token	GET	정상적인 JWT 토큰 생성	{ "token": "eyJ..." }
/v1/token/none	GET	none 취약점을 가진 비정상적 JWT 토크 생성	{ "token": "eyJ..." }
/v1/token/verify	GET	토큰의 Payload에서 isAdmin의 true/false값에 따라 결과 표시	You are not permitted / You are admin!
/healthcheck	GET	서버 동작 확인	ok

6. 로드밸런서

ELB를 이용하여 워크로드를 Private Subnet에 위치한 2대 이상의 EC2로 분산하도록 구성합니다. 웹 어플리케이션은 로드밸런서를 통해서만 HTTP 요청을 받아야 합니다. Bastion 서버와 같이 외부에서 로드밸런서를 거치지 않은 HTTP 요청은 요청 시 차단되어야 합니다.

- Network facing : 인터넷 망에서 로드밸런서로 접근 가능하도록 구성
- Listen : HTTP 80을 통해 접근 가능하도록 구성
- 이름 : wsi-alb
- Tag : Name=wsi-alb

7. 보안 구성

보안성 향상을 위해 AWS WAF를 구성하여 웹 어플리케이션의 GET /v1/token/verify 엔드포인트로 보내는 요청에 포함된 JWT 토큰이 유효하지 않다면 접근을 제한하고자 합니다. JWT 토큰의 서명 알고리즘 타입이 "none"이고, 요청 Path가 /v1/token/verify라면 해당 토큰의 요청을 401(Unauthorized) 응답코드와 함께 Body에는 'Blocked by WAF'를 반환하도록 구성하세요.

- Name : wsi-waf