9 Kubernetes 트러블 슈팅 방법

05 보안관련 로깅 및 이슈 사례 조치방법

실습 내용

순서

1. 보안 로깅 방법 소개
2. 보안 이슈 사례 소개
3. 보안 로깅 방법 실습
4. 보안 이슈 사례 발생시 확인 방법 실습

1. 보안 로깅 방법 소개

구분	상세 설명
Access Log	• API 서버에 request 및 처리 현황 로깅 • AWS IAM 자격 증명을 사용한 RBAC 인증 내역 로깅
Audit Log	• EKS 클러스터에 접속해 API, 명령어를 사용하는 사용자의 감시 로깅 • EKS 클러스터 내부 및 외부 시스템에서 API를 사용해 처리된 부분 감시 로깅
CloudTrail Log	• AWS Resource 사용을 위한 API 호출 이벤트 로깅 • AWS Management Console 내에서 사용한 계정 이벤트 로깅
WAF Log	• 인터넷에서 사용자의 웹서비스에 접속하려는 모든 행위에 대한 로깅 • 해킹, 침입 탐지, Bot, DDoS 공격 등의 취약점에 대한 보안 로깅

2. 보안 이슈 사례 소개

침입 탐지

보안 취약점 발견

특이 API호출 발견

공격 시도 탐지

3. 보안 로깅 방법 실습 #1

(1) Access Log를 통한 접속 및 처리내역 확인

- AWS CloudWatch > 로그 > 로그 그룹 > /aws/eks/<EKS명>/cluster/authenticator

(2) Audit Log를 통한 보안 상태 및 보안 취약점 확인

- AWS CloudWatch > 로그 > 로그 그룹 > /aws/eks/<EKS명>/cluster/kube-apiserver-audit

3. 보안 로깅 방법 실습 #2

(3) CloudTrail Log를 통한 AWS API Audit 로그 확인

- AWS CloudWatch > 로그 > 로그 그룹 > aws-cloudtrail-logs

(4) WAF Log를 통한 웹 보안 침입 탐지 로그 확인

- AWS CloudWatch > 로그 > 로그 그룹 > aws-waf-logs

4. 보안 이슈 사례 발생시 확인 방법 실습

(1) 침입 탐지/보안 취약점 발견 (WAF)

- AWS WAF > Web ACLs > (생성된 Web ACLs 선택) > **Overview**

(2) 특이 API호출 발견/공격 시도 탐지 (CloudTrail)

- AWS CloudTrail > 이벤트 기록 > **전체 이벤트 기록 보기**

실습 아키텍처 구성사항 #1

1. AWS Network 구성

- VPC1개, Public Subnet 2개
- Internet Gateway 1개

2. AWS EKS 구성

- EKS Cluster 1개, EKS NodeGroup1개(2개 Worker Node 생성)

실습 아키텍처 구성사항 #2

3.1 Terraform으로 Backend생성

- IaC 경로 : Ch09_05-security-troubleshooting > terraform-backend

3.2 terraform 명령어 실행

$ terraform init

$ terraform plan

$ terraform apply

실습 아키텍처 구성사항 #3

4.1 Terraform으로 AWS 클라우드 아키텍처 구성 경로

- IaC 경로 : Ch09_05-security-troubleshooting > terraform-codes

4.2 terraform 명령어 실행하여 전체 아키텍처 구성

$ terraform init

$ terraform plan

$ terraform apply

실습 아키텍처 구성사항 #4

5.1 Ingress 구성을 위한 IAM Policy 적용을 위한 예제코드 경로로 이동

- 경로 : Ch09_05-security-troubleshooting > iam-policy

5.2 Ingress 구성을 위한 IAM Policy 적용

$ aws iam create-policy --policy-name AWSLoadBalancerControllerIAMPolicy --policy-document file://iam_policy.json

실습 아키텍처 구성사항 #5

5.3 Ingress 구성을 위한 eksctl을 통한 Service Account 및 IAM Role 설정

```
$ eksctl create iamserviceaccount \
  --cluster=<EKS Cluster명> \
  --namespace=kube-system \
  --name=aws-load-balancer-controller \
  --role-name "AmazonEKSLoadBalancerControllerRole" \
  --attach-policy-arn=arn:aws:iam::<AWS 계정 ID>:policy/AWSLoadBalancerControllerIAMPolicy \
  --approve
```

실습 아키텍처 구성사항 #6

5.4 Ingress 구성을 위한 Helm Chart Repository 등록

```
$ helm repo add eks https://aws.github.io/eks-charts
```

5.5 Ingress 구성을 위한 AWS Load Balancer Controller를 배포

```
$ helm install aws-load-balancer-controller eks/aws-load-balancer-controller \
  -n kube-system \
  --set clusterName=<EKS 클러스터명> \
  --set serviceAccount.create=false \
  --set serviceAccount.name=aws-load-balancer-controller
```

실습 아키텍처 구성사항 #7

6.1 예제를 위한 마이크로서비스 배포 경로

- 경로 : Ch09_05-security-troubleshooting > k8s-manifests

6.2 마이크로서비스의 Deployment(POD), Service, Ingress 배포

$ kubectl apply -f ./