AWS 3-Tier 웹 서비스 인프라 구축 가이드

안전하고 확장 가능한 클라우드 아키텍처 설계

SK 쉴더스 루키즈 AWS 클라우드 교육 프로젝트

2025년 12월

1. 프로젝트 개요

1.1 배경 및 목적

본 프로젝트는 SK 쉴더스 루키즈 AWS 클라우드 교육을 통해 학습한 내용을 실무 시나리오에 적용하여, 안전하고 확장 가능한 3-Tier 웹 애플리케이션 인프라를 설계하고 구축 방안을 문서화합니다. 현대 웹 서비스에 요구되는 높은 가용성, 확장성, 보안성을 동시에 달성할 수 있는 클라우드 아키텍처를 제시합니다.

1.2 시나리오 설정

비즈니스 시나리오:

중소 규모의 전자상거래 웹사이트를 AWS에 구축합니다. 평시 1,000명/일, 프로모션 시 10,000명/일의 트래픽을 처리하며, 고객 개인정보와 결제 정보 보호를 위한 강력한 보안이 필요합니다. 서비스 중단은 직접적인 매출 손실로 이어지므로 99.9% 이상의 가용성을 목표로 합니다.

핵심 요구사항:

- 가용성: 99.9% 이상 (월간 최대 43 분 다운타임)
- 확장성: 10 배 트래픽 증가 시 자동 대응
- 보안: PCI-DSS 기본 준수, 데이터 암호화
- 성능: 웹 페이지 로딩 3 초 이내
- 비용: 사용한 만큼만 지불하는 종량제

2. 요구사항 분석

2.1 비기능적 요구사항

항목	목표	달성 방법
성능	응답시간 < 3초	CloudFront CDN, ElastiCache 캐싱
확장성	10배 트래픽 대응	Auto Scaling Group, ELB
보안	제로 데이터 유출	IAM, Security Groups, WAF, 암호화
가용성	99.9% 이상	Multi-AZ, Auto Scaling, Health Check

3. 아키텍처 설계

3.1 전체 아키텍처 개요

본 프로젝트는 3-Tier 아키텍처를 기반으로 설계되었습니다. Presentation(프레젠테이션), Application(애플리케이션), Data(데이터) 계층으로 구분되어, 각 계층이 독립적으로 동작하며 확장 및 유지보수가 용이합니다.

계층별 역할:

- **Presentation Tier:** 사용자 접점, 콘텐츠 전송 (CloudFront, Route53, WAF)
- **Application Tier:** 비즈니스 로직 처리 (VPC, EC2, Auto Scaling, ALB)
- **Data Tier:** 데이터 저장 및 관리 (RDS, S3, ElastiCache)

3.2 AWS 서비스 선정

계층	AWS 서비스	선정 이유
네트워크	VPC	격리된 가상 네트워크, 세밀한 제어
컴퓨팅	EC2 + Auto Scaling	완전한 제어, 자동 확장
로드밸런서	ALB	HTTP/HTTPS 분산, 경로 기반 라우팅
데이터베이스	RDS Multi-AZ	관리형 서비스, 자동 백업, 고가용성
스토리지	S3	높은 내구성(11 nines), 무제한 용량
CDN	CloudFront	글로벌 콘텐츠 전송, DDoS 보호
보안	IAM + WAF	세밀한 권한 관리, 웹 공격 차단
모니터링	CloudWatch	통합 모니터링, 알람, 로그 수집

4. 네트워크 설계 (VPC)

4.1 VPC 및 Subnet 구성

CIDR 블록 설계:

- **VPC:** 10.0.0.0/16 (65,536 개 IP)
- **Public Subnet (AZ-1/2):** 10.0.1.0/24, 10.0.2.0/24
- **Private Subnet (AZ-1/2):** 10.0.11.0/24, 10.0.12.0/24
- **Database Subnet (AZ-1/2):** 10.0.21.0/24, 10.0.22.0/24

Public Subnet:

- Internet Gateway 를 통해 인터넷 직접 통신
- ALB, NAT Gateway, Bastion Host 배치

Private Subnet:

- NAT Gateway 를 통해서만 아웃바운드 접속
- EC2 애플리케이션 서버, ElastiCache 배치

Database Subnet:

- 완전 격리, 애플리케이션 서버만 접근 가능
- RDS 데이터베이스 배치

4.2 Security Groups 설계

리소스	인바운드 규칙	아웃바운드 규칙
ALB	HTTP(80), HTTPS(443) from 0.0.0.0/0	All traffic
EC2 (Web)	HTTP(80) from ALB SG	All traffic
RDS	MySQL(3306) from EC2 SG	None

5. 컴퓨팅 계층 구성

5.1 EC2 및 Auto Scaling

인스턴스 구성:

- **인스턴스 타입:** t3.medium (2 vCPU, 4GB RAM)
- **OS:** Amazon Linux 2
- **스토리지:** EBS gp3 50GB

Auto Scaling 설정:

- **Desired:** 2 개 (정상 운영)
- **Minimum:** 2 개 (최소 가용성)
- **Maximum:** 10 개 (피크 시간)
- **정책:** Target Tracking - CPU 70% 유지

5.2 Application Load Balancer

ALB 주요 기능:

- HTTP/HTTPS 트래픽 분산
- 경로 기반 라우팅 (/api, /static)
- Health Check 로 비정상 인스턴스 제외
- SSL/TLS 종료 (ACM 인증서)

Health Check 설정:

- **프로토콜:** HTTP
- **경로:** /health
- **간격:** 30 초
- **타임아웃:** 5 초
- **임계값:** 2 회 연속 성공/실패

6. 데이터 계층 구성

6.1 Amazon RDS (데이터베이스)

RDS 구성:

- **엔진:** MySQL 8.0
- **인스턴스:** db.t3.medium
- **스토리지:** 100GB gp3 (IOPS 3000)
- **Multi-AZ:** 활성화 (고가용성)
- **백업:** 자동 백업 7 일 보관

보안 설정:

- 암호화: 저장 시 AES-256 암호화
- 접근: EC2 Security Group 만 허용
- 엔드포인트: Private Subnet 내부만 접근

6.2 Amazon S3 (스토리지)

S3 버킷 구성:

- **용도:** 정적 파일 (이미지, CSS, JS)
- **버전 관리:** 활성화 (실수로 삭제 방지)
- **암호화:** SSE-S3 (서버 측 암호화)
- **Lifecycle:** 90 일 후 Glacier 로 이동

6.3 Amazon ElastiCache

캐시 구성:

- **엔진:** Redis 7.x
- **용도:** 세션 데이터, DB 쿼리 캐싱
- **노드:** cache.t3.micro x 2 (Multi-AZ)

7. 보안 구성

7.1 IAM (Identity and Access Management)

IAM 구성 원칙:

- **최소 권한 원칙:** 필요한 권한만 부여
- **역할 기반:** EC2 에 Role 할당, 하드코딩 금지
- **MFA:** 관리자 계정에 다중 인증 활성화
- **정기 검토:** 사용하지 않는 권한 제거

7.2 AWS WAF (Web Application Firewall)

WAF 규칙:

- SQL Injection 차단
- XSS (Cross-Site Scripting) 차단
- Rate Limiting (IP 당 초당 100 요청)
- Known Bad IP 차단 (AWS Managed Rules)

7.3 데이터 암호화

전송 중 암호화:

- HTTPS (TLS 1.2 이상) 강제
- ACM (AWS Certificate Manager)으로 인증서 관리

저장 시 암호화:

- **RDS:** AES-256 암호화
- **S3:** SSE-S3 서버 측 암호화
- **EBS:** 볼륨 암호화 활성화

8. 모니터링 및 로깅

8.1 Amazon CloudWatch

모니터링 메트릭:

- **EC2:** CPU, 메모리, 디스크, 네트워크
- **ALB:** 요청 수, 응답 시간, 오류율
- **RDS:** CPU, 연결 수, 쿼리 실행 시간
- **S3:** 요청 수, 전송량

알람 설정:

- CPU 사용률 80% 초과 시 알람
- ALB 응답 시간 3 초 초과 시 알람
- RDS 연결 수 200 개 초과 시 알람

8.2 AWS CloudTrail

CloudTrail은 모든 AWS API 호출을 기록하여 감사 및 컴플라이언스를 지원합니다.

- 모든 API 호출 로그 S3 에 저장
- 보안 그룹 변경 시 즉시 알림
- IAM 권한 변경 추적

9. 비용 분석

9.1 월간 예상 비용

서비스	사양	사용량	월 비용
EC2	t3.medium	2개 x 730시간	$60
RDS	db.t3.medium Multi-AZ	730시간	$120
ALB	Application Load Balancer	730시간	$25
S3	Standard Storage	100GB	$3
CloudFront	CDN	1TB 전송	$85
ElastiCache	cache.t3.micro x 2	730시간	$25
합계			**$318**

※ 위 비용은 서울 리전(ap-northeast-2) 기준 평시 운영 비용입니다. 프로모션 기간 등 트래픽 증가 시 Auto Scaling에 의해 추가 비용이 발생할 수 있습니다.

9.2 비용 최적화 전략

- **예약 인스턴스:** 1 년 약정 시 최대 40% 할인
- **Savings Plans:** 유연한 컴퓨팅 사용량 약정
- **Auto Scaling:** 사용하지 않는 시간에 인스턴스 축소
- **S3 Lifecycle:** 오래된 데이터 Glacier 로 이동
- **CloudWatch 알람:** 비정상 비용 증가 즉시 탐지

10. 결론 및 향후 계획

10.1 프로젝트 성과

본 프로젝트를 통해 AWS 클라우드에서 안전하고 확장 가능한 3-Tier 웹 애플리케이션 인프라를 설계했습니다. VPC를 통한 네트워크 격리, Auto Scaling을 통한 자동 확장, Multi-AZ 배포를 통한 고가용성, IAM과 Security Groups를 통한 보안 강화 등 AWS의 핵심 서비스를 실무 시나리오에 적용하는 방법을 체계적으로 학습했습니다.

핵심 학습 내용:

- 클라우드 아키텍처 설계 원칙 이해
- AWS 주요 서비스의 역할과 통합 방법
- 보안과 가용성을 동시에 고려한 설계
- 비용 효율적인 리소스 구성
- 모니터링과 로깅을 통한 운영 관리

10.2 향후 개선 방향

단기 개선 사항:

- **CI/CD 파이프라인:** CodePipeline 으로 자동 배포
- **컨테이너화:** ECS/EKS 로 마이크로서비스 전환
- **서버리스 통합:** Lambda 로 이벤트 기반 처리

장기 개선 사항:

- **글로벌 확장:** 다중 리전 배포
- **AI/ML 통합:** SageMaker 로 추천 시스템
- **빅데이터 분석:** Redshift 로 데이터 웨어하우스

10.3 마무리

이번 프로젝트는 단순히 AWS 서비스를 나열하는 것이 아니라, 실제 비즈니스 요구사항을 분석하고 이를 충족하는 아키텍처를 설계하는 전 과정을 다루었습니다. 이전 프로젝트에서 학습한 Linux 시스템 관리와 네트워크 프로토콜 지식은 클라우드 환경에서도 그대로 적용되며, 오히려 더 중요한 기반 지식임을 확인했습니다. AWS는 단순히 서버를 빌려주는 것이 아니라, 복잡한 인프라를 코드처럼 관리하고 자동화할 수 있는 플랫폼을 제공합니다. 앞으로도 지속적으로 학습하고 실습하며 클라우드 전문가로

성장해 나가겠습니다.

참고문헌

- AWS Well-Architected Framework (https://aws.amazon.com/architecture/well-architected/)
- AWS Architecture Center (https://aws.amazon.com/architecture/)
- AWS 공식 문서 (https://docs.aws.amazon.com/)
- SK 쉴더스 루키즈 AWS 클라우드 교육 자료
- AWS Pricing Calculator (https://calculator.aws/)