

모듈 4: AWS 아키텍처 설계

주제

- Well-Architected 프레임워크 소개
- 참조 아키텍처 – 내결함성 및 고가용성
- 참조 아키텍처: 웹 호스팅

Well-Architected 프레임워크 소개

소개

- 아키텍처를 평가하고 개선
- 설계 의사 결정이 비즈니스에 미치는 영향을 이해
- 5가지 원칙과 설계 원칙을 학습

5가지 원칙

- 보안
- 안정성
- 성능 효율성
- 비용 최적화
- 운영 우수성

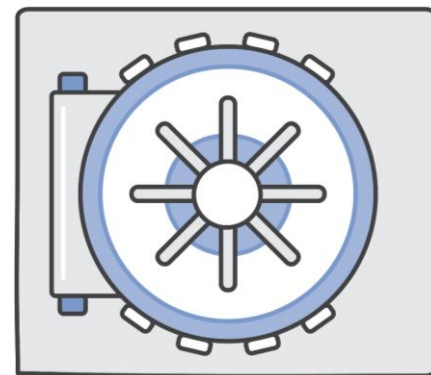
보안 원칙

- Identity and Access Management (IAM)
- 탐지 제어
- 인프라 보호
- 데이터 보호
- 인시던트 대응



보안 원칙 설계 원칙

- 모든 계층에 보안 구현
- 추적 가능성 활성화
- 최소 권한의 원칙 적용
- 시스템 보호에 집중
- 자동화



안정성 원칙

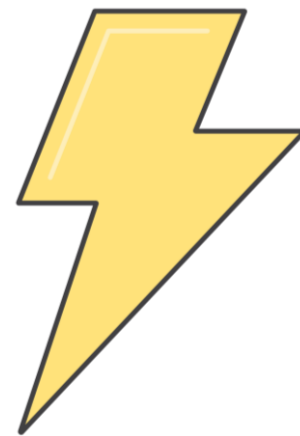
- 문제/장애로부터 복구
- 다음에 모범 사례를 적용:
 - 기반
 - 변경 관리
 - 장애 관리
- 장애를 예측, 대응 및 방지

안정성 원칙 설계 원칙

- 복구 절차 테스트
- 자동 복구
- 수평 확장
- 용량 추정 불필요
- 자동화 변경 사항 관리

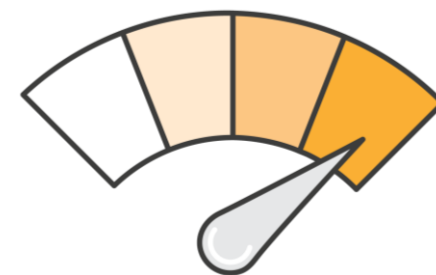
성능 효율성 원칙

- 사용자 지정 가능한 솔루션 선택
- 지속적 혁신을 위한 검토
- AWS 서비스 모니터링
- 절충을 고려



성능 효율성 원칙 : 설계 원칙

- 고급 기술을 대중화
- 몇 분 만에 전 세계에 배포
- 서버리스 아키텍처 사용
- 실험 빈도 증가
- 기술적 동조 적용



비용 최적화 원칙

- 비용 효율적인 리소스 사용
- 수요에 맞춰 공급
- 지출에 대한 인식 고취
- 시간을 두고 최적화

비용 최적화 원칙 : 설계 원칙

- 소비 모델 채택
- 전반적인 효율성 측정
- 데이터 센터 운영에 지출하는 비용 절감
- 지출 속성 파악 및 분석
- 관리형 서비스 사용



운영 우수성 원칙

- 변경 관리 및 자동화
- 이벤트에 응답
- 표준을 정의

요약

- 5가지 원칙 및 관련 설계 원칙
 - 보안
 - 안정성
 - 성능 효율성
 - 비용 최적화
 - 운영 우수성

참조 아키텍처 – 내결함성 및 고가용성

내결함성

- 시스템이 작동 가능 상태를 유지할 수 있는 능력
- 애플리케이션 구성요소의 내장된 중복성

고가용성

고가용성은 다음을 위해 설계됩니다.

- 시스템이 일반적으로 작동하고 액세스 가능한 상태 유지
- 가동 중단을 최소화
- 필요한 인적 개입을 최소화
- 사전 비용 투자를 최소화



고가용성 온프레미스와 AWS 비교

- 기존(온프레미스)
 - 높은 비용
 - 미션 크리티컬 애플리케이션만 해당
- AWS
 - 여러 대의 서버
 - 가용 영역
 - 리전
 - 내결함성 서비스



고가용성 AWS 서비스

AWS 서비스 및 고가용성

- ✓ Amazon S3 및 Amazon Glacier
- ✓ DynamoDB
- ✓ Amazon CloudFront
- ✓ Amazon SWF
- ✓ Amazon SQS
- ✓ Amazon SNS
- ✓ Amazon SES
- ✓ Amazon Route 53
- ✓ Elastic Load Balancing
- ✓ IAM
- ✓ Amazon CloudWatch
- ✓ Amazon CloudSearch
- ✓ AWS Data Pipeline
- ✓ Amazon Kinesis
- ✓ Auto Scaling
- ✓ Amazon Elastic File System
- ✓ AWS CloudFormation
- ✓ Amazon WorkMail
- ✓ AWS Directory Service
- ✓ AWS Lambda
- ✓ Amazon EBS
- ✓ Amazon RDS

기본적으로 HA 지원 서비스

- ▶ Amazon EC2
- ▶ Amazon VPC
- ▶ Amazon Redshift
- ▶ Amazon ElastiCache
- ▶ AWS Direct Connect

* 여기에 나열된 서비스가 전부는 아닙니다.

적절한 아키텍처를 통해 HA 지원

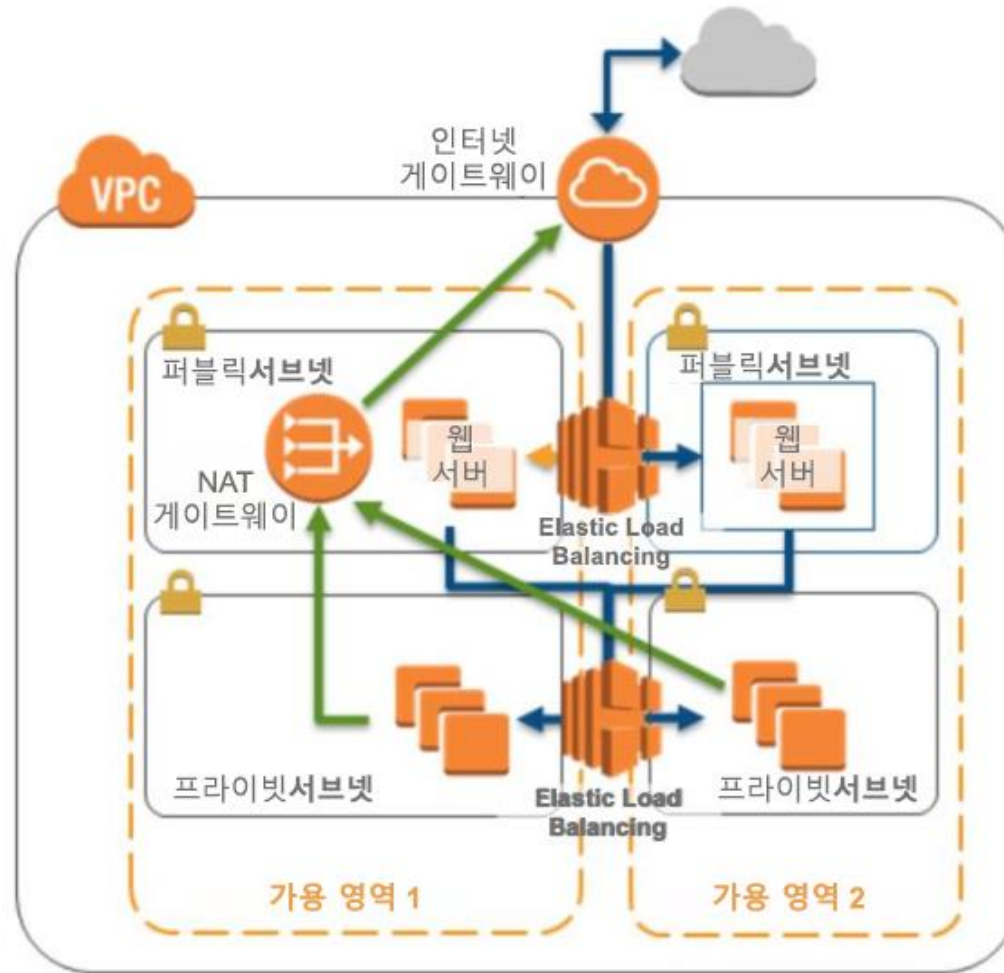
고가용성 서비스 도구

- 탄력적 로드 밸런서
- 탄력적 IP 주소
- Amazon Route 53
- Auto Scaling
- Amazon CloudWatch

탄력적 로드 밸런서

- 수신 트래픽(로드) 분산
- Amazon CloudWatch로 지표 전송
- 트리거/알림
 - 긴 지연 시간
 - 과도한 사용

탄력적 로드 밸런서



탄력적 IP 주소

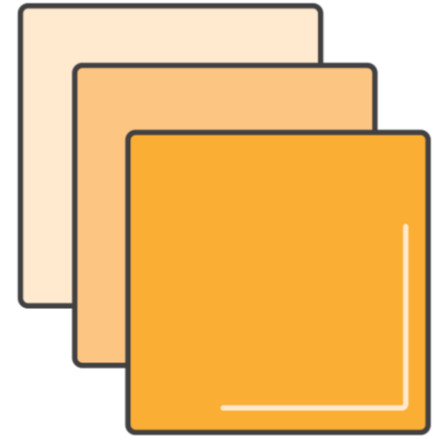
- 고정 IP 주소입니다.
- 장애를 마스킹합니다(장애가 발생하는 경우).
- 인스턴스에 장애가 발생해도 애플리케이션에 계속 액세스합니다.

Amazon Route 53

- 신뢰할 수 있는 DNS
 - 도메인 이름을 IP 주소로 변환
- 지원:
 - 단순 라우팅
 - 지연 시간 기반 라우팅
 - 상태 확인
 - DNS 장애 조치
 - 지리 위치 라우팅

Auto Scaling

- 인스턴스를 종료 및 시작
- 용량 조정 또는 변경을 지원
- 온디맨드로 새 리소스를 생성



Amazon CloudWatch

- 경보 예제:
 - CPU 사용률이 5분 동안 60%를 초과하면...
 - 동시 연결 수가 1분 동안 10개를 초과하면...
 - 정상 호스트 수가 10분 동안 5개 미만이 되면...

내결함성 도구

- Amazon Simple Queue Service
- Amazon Simple Storage Service
- Amazon SimpleDB
- Amazon Relational Database Service

요약

- 내결함성 및 고가용성을 갖춘 아키텍처
- 아키텍처를 지원하는 서비스

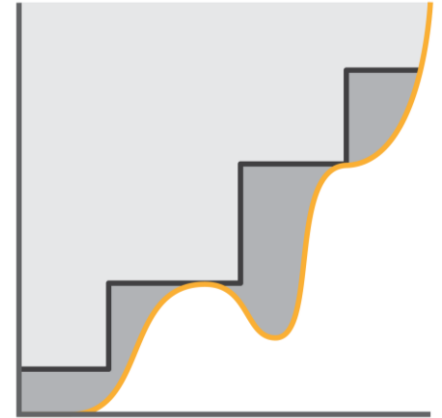
참조 아키텍처 – 웹 호스팅

웹 호스팅

- AWS에서 웹 호스팅
 - 빠름
 - 간단함
 - 저렴한 비용
- 일반적인 웹 애플리케이션:
 - 기업 웹 사이트
 - 콘텐츠 관리 시스템
 - 소셜 미디어 애플리케이션 개발
 - 사내 SharePoint 사이트

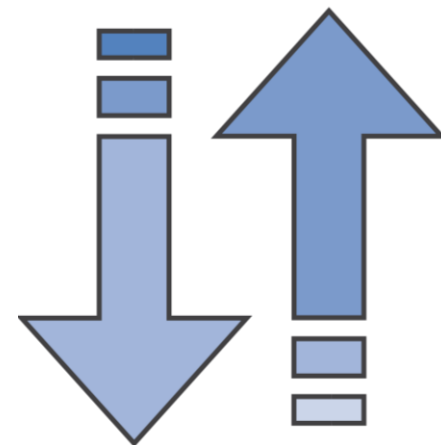
비용 효율적 대안

- 온디맨드 프로비저닝 활용
- 낭비되는 용량 제거
- 실제 트래픽 패턴에 맞춰 계속해서 조정



확장성

- 예상치 못한 트래픽 피크 또는 예상치 못한로드를 처리
- 몇 분 만에 새로운 호스트를 시작
- 호스트를 확장 또는 축소



다양한 환경을 위한 온디맨드 솔루션

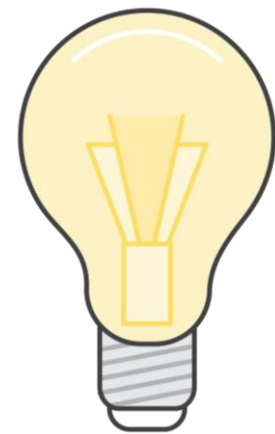
- 테스트 플릿을 프로비저닝
- 몇 분 만에 스테이징을 개발
- 사용자 트래픽을 시뮬레이션

AWS로 마이그레이션: AWS 웹 호스팅 서비스

- 이전에 도움이 되는 제품
 - Amazon Virtual Private Cloud
 - Amazon Route 53
 - Amazon CloudFront
 - Elastic load balancing
 - 방화벽/AWS Shield
 - Auto Scaling
 - 웹 서버/EC2 인스턴스
 - Amazon ElastiCache
 - Amazon RDS/Amazon DynamoDB

주요 아키텍처 관련 고려 사항

- 물리적 네트워크 어플라이언스를 소프트웨어 솔루션으로 대체
- 모든 곳에 방화벽 배포
- 여러 데이터 센터를 사용
- 임시적이고 동적인 아키텍처를 구축



요약

- AWS 및 웹 호스팅
- AWS 웹 호스팅 서비스
- 웹 호스팅 아키텍처의 주요 고려 사항

지식 확인

- 다음 중 성능 효율성 원칙의 4가지 영역 중 하나에 속하지 않는 것은 무엇입니까?
 - 절충
 - 선택
 - 모니터링
 - 추적 기능

지식 확인

- 새 리소스를 온디맨드 또는 예약된 일정에 따라 생성할 수 있는 능력의 한계를 벗어나는 데 도움이 되는 도구는 다음 중 무엇입니까?
 - Route 53
 - 탄력적 로드 밸런서
 - Auto Scaling
 - CloudWatch

지식 확인

- 물리적 데이터 센터에서는 일반적으로 어떤 영역에서 보안을 고려하게 됩니까?
 - 경계에서만
 - 엣지 로케이션에서
 - 가장 가까운 리전에서
 - 가장 가까운 가용 영역에서

지식 확인

- 시스템의 일부 구성 요소에 장애가 발생해도 시스템이 계속 작동할 수 있는 기능으로 정의되는 것은 무엇입니까?
 - DNS 장애 조치
 - 높은 내구성
 - 고가용성
 - 내결함성

지식 확인

- 다음 중 Amazon Route 53의고가용성 특성에 해당하는 것은 무엇입니까? (2가지 선택)
 - 지연 시간 기반 라우팅
 - 지리 위치 라우팅
 - 긴 지연 시간 지표를 수집 및 추적
 - 인스턴스/소프트웨어의 장애를 마스킹
 - 지정된 조건에 따라 인스턴스를 종료

지식 확인

- 성능 효율성을 고려할 때 권장되는 설계 원칙은 다음 중 무엇입니까?
(2가지 선택)
 - 추적 기능 활성화
 - 고급 기술을 대중화
 - 지출 인식
 - 수요에 맞춰 공급
 - 서버리스 아키텍처

지식 확인

- 다음 중 인증되고 권한이 있는 사용자만 리소스에 액세스할 수 있도록 설계된 클라우드 보안 제어 항목은 무엇입니까?
 - 탐지 제어
 - 자격 증명 및 액세스 관리
 - 인프라 보호
 - 인시던트 대응

지식 확인

- 비용 최적화를 고려할 때 실제로 사용하는 컴퓨팅 리소스에 대해서만 비용을 지불할 수 있는 모델은 다음 중 무엇입니까?
 - 소비 모델
 - 범위의 경제 모델
 - 규모의 경제 모델
 - 지출 모델

지식 확인

- 다음 중 탄력적 로드 밸런서(ELB)에 대한 설명은 무엇입니까?
 - 지정된 조건에 따라 인스턴스를 시작 또는 종료
 - 온디맨드로 새 리소스를 생성
 - 여러 인스턴스에 걸쳐 수신 트래픽을 분산
 - 도메인 이름을 IP 주소로 변환

지식 확인

- 다음 중 내결함성 도구로 간주되지 않는 것은 무엇입니까?
 - S3
 - WAF
 - SQS
 - RDS