AWS 백서

AWS 클라우드에서의 웹 애플리케이션 호스팅

Copyright © Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS 클라우드에서의 웹 애플리케이션 호스팅: AWS 백서

Copyright © Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon의 상표 및 브랜드 디자인은 Amazon 외 제품 또는 서비스와 함께, Amazon 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계 여부에 관계없이 해당 소유자의 자산입니다.

Table of Contents

AWS 클라우드에서의 웹 애플리케이션 호스팅

게시 날짜: 2021년 8월 20일(문서 개정)

요약

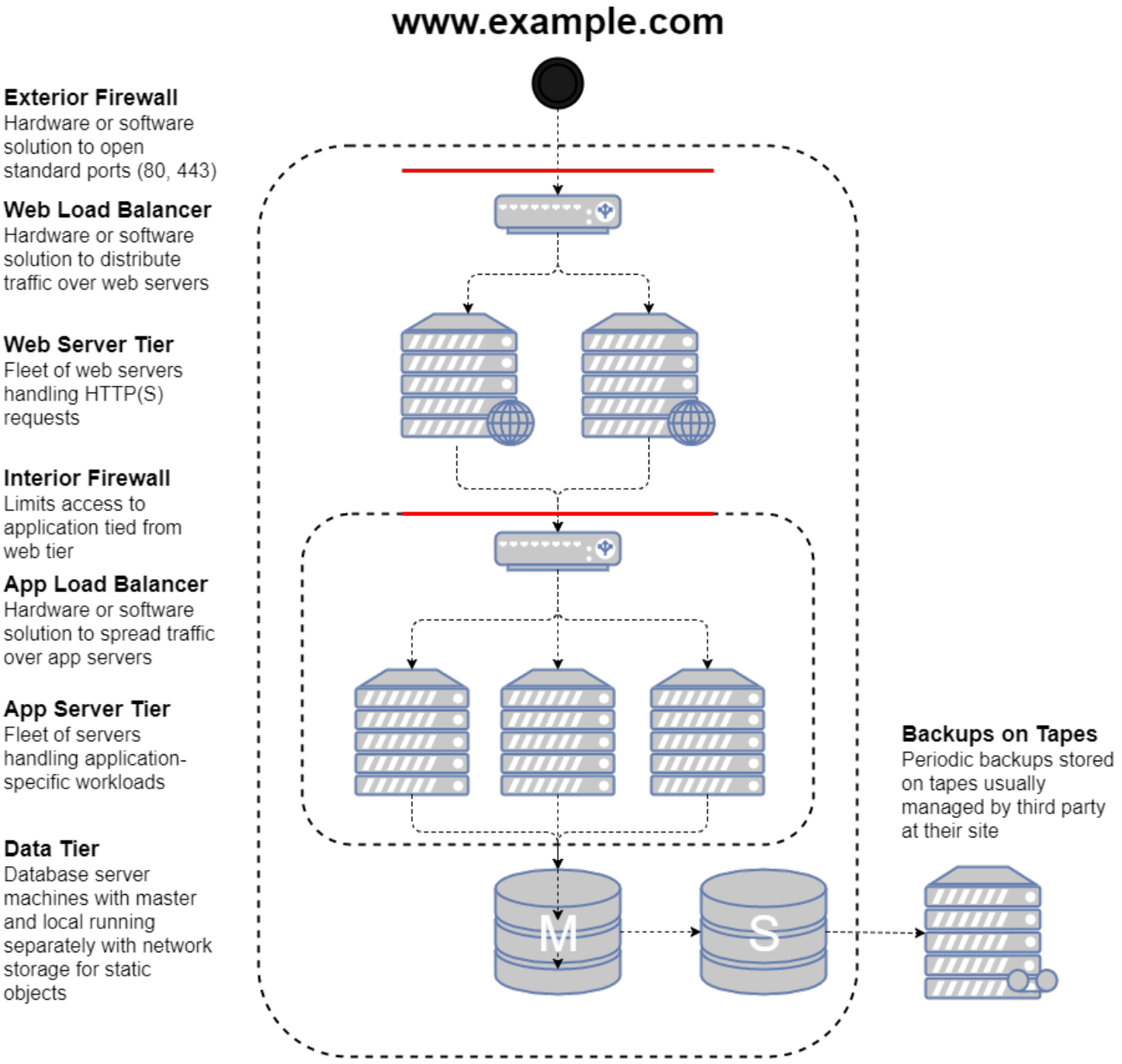
기존 온프레미스 웹 아키텍처는 안정성을 보장하기 위해 복잡한 솔루션과 정확한 예약 용량 예측이 필요합니다. 밀집된 피크 트래픽 기간과 트래픽 패턴의 급격한 변동으로 인해 고가 하드웨어의 사용률이 저조합니다. 이로 인해 유휴 하드웨어를 유지 관리하는 데 많은 운영 비용이 발생하고 사용되지 않는 하드웨어에 투입된 자본의 효율이 크게 저하됩니다.

Amazon Web Services(AWS)에서는 가장 까다로운 웹 애플리케이션을 위한 안정적이고, 확장 가능하며, 안전하고, 성능이 뛰어난 인프라를 제공합니다. 이 인프라는 거의 실시간으로 IT 비용을 고객 트래픽 패턴과 일치시킵니다.

이 백서는 클라우드에서 기존 웹 아키텍처를 실행하여 탄력성, 확장성 및 안정성을 달성하는 방법을 이해하려는 IT 관리자 및 시스템 설계자를 대상으로 합니다.

기존 웹 호스팅 개요

확장 가능한 웹 호스팅은 잘 알려진 문제 공간입니다. 다음 이미지는 일반적인 3티어 웹 애플리케이션 모델을 구현하는 기존 웹 호스팅 아키텍처를 보여줍니다. 이 모델에서 아키텍처는 프레젠테이션, 애플리케이션 및 지속성 계층으로 구분됩니다. 이러한 계층에 호스트를 추가하면 확장성이 제공됩니다. 아키텍처에는 성능, 장애 조치 및 가용성 기능도 내장되어 있습니다. 기존 웹 호스팅 아키텍처는 몇 가지 수정만으로 손쉽게 AWS 클라우드로 이식할 수 있습니다.



기존 웹 호스팅 아키텍처

다음 섹션에서는 이러한 아키텍처를 AWS 클라우드에 배포해야 하는 이유와 방법에 대해 살펴봅니다.

AWS를 사용하여 클라우드에서 웹 애플리케이션 호스팅

가장 먼저 물어야 할 질문은 기존 웹 애플리케이션 호스팅 솔루션을 AWS 클라우드로 이전하는 가치에 관한 것입니다. 클라우드가 적합하다는 판단을 내렸다면 적절한 아키텍처가 필요합니다. 이 섹션은 AWS 클라우드 솔루션을 평가하는 데 도움이 됩니다. 클라우드에서의 웹 애플리케이션 배포를 온프레미스 배포와 비교하고, 애플리케이션 호스팅을 위한 AWS 클라우드 아키텍처를 제시하며, AWS 클라우드 아키텍처 솔루션의 주요 구성 요소에 대해 설명합니다.

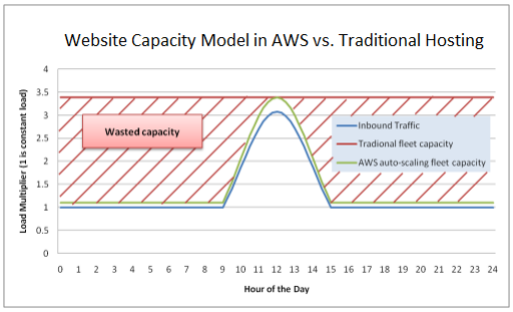
AWS가 일반적인 웹 애플리케이션 호스팅 문제를 해결하는 방법

웹 애플리케이션 실행을 담당하고 있다면 다양한 인프라 및 아키텍처 문제에 직면할 수 있습니다. AWS는 이러한 문제에 대한 원활하고 비용 효율적인 솔루션을 제공할 수 있습니다. 다음은 기존 호스팅 모델 대신 AWS를 사용하는 이점 몇 가지입니다.

과도한 서버 대신 효율적 비용으로 트래픽 최고치를 처리

기존 호스팅 모델에서는 최대 용량을 처리하도록 서버를 프로비저닝해야 합니다. 피크 기간 외에는 사용하지 않은 사이클이 낭비됩니다. AWS에서 호스트하는 웹 애플리케이션은 추가 서버를 온 디맨드로 프로비저닝할 수 있으므로 실제 트래픽 패턴에 맞춰 용량과 비용을 지속적으로 조정할 수 있습니다.

예를 들어 다음 그래프는 사용량이 오전 9시부터 오후 3시까지 최고치를 기록하고 나머지 시간에는 사용량이 감소하는 웹 애플리케이션을 보여줍니다. 필요한 경우에만 리소스를 프로비저닝하는 실제 트래픽 추세에 기반한 자동 크기 조정 방식을 사용하면 낭비되는 용량이 줄어들고 비용이 50% 이상 절감됩니다.



기존 호스팅 모델에서 낭비되는 용량의 사례

예측하지 못한 트래픽 피크를 처리하는 확장 가능한 솔루션

이보다 더 끔찍한 일은 기존 호스팅 모델의 느린 프로비저닝 속도 때문에 예상하지 못한 트래픽 요청 급증에 제때 대응하지 못하는 것입니다. 사이트가 인기 미디어에 언급된 후 예상치 못한 트래픽 급증

으로 인해 웹 애플리케이션이 불통이 된 사례는 많이 있습니다. AWS 클라우드에서는 정기적인 트래픽 급증에 맞춰 웹 애플리케이션을 확장하는 데 도움이 되는 동일한 온디맨드 기능은 예상치 못한 로드도 처리할 수 있습니다. 새 호스트는 시작하고 몇 분 만에 즉시 사용할 수 있으며 트래픽이 정상으로 돌아오면 마찬가지로 빠르게 오프라인 상태로 전환될 수 있습니다.

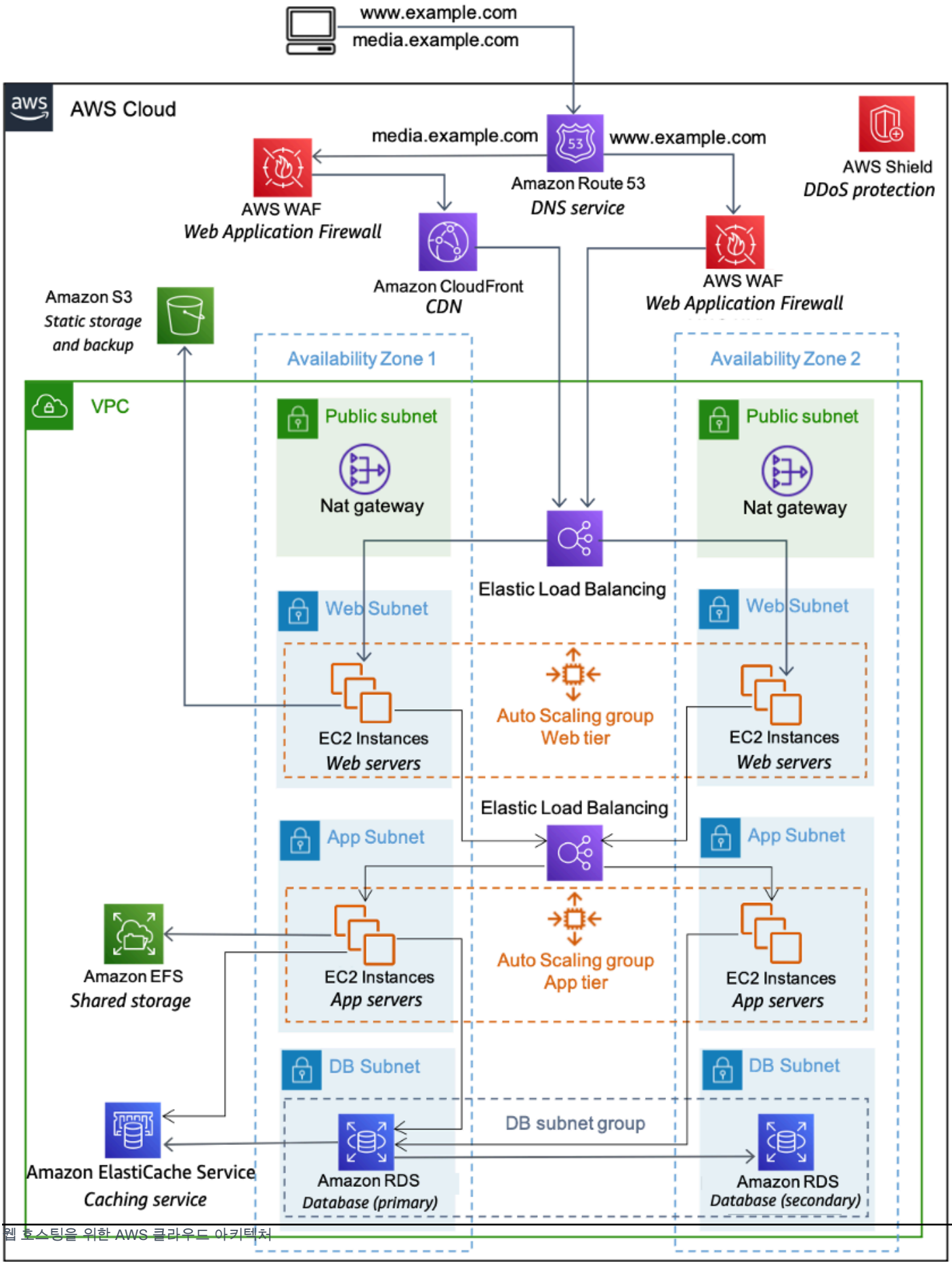
테스트, 로드, 베타 및 사전 프로덕션 환경을 위한 온디맨드 솔루션

프로덕션 웹 애플리케이션을 위한 기존 호스팅 환경을 구성 및 유지 관리하기 위한 하드웨어 비용은 프로덕션 플릿에서 멈추는 것이 아닙니다. 개발 수명 주기의 각 단계에서 웹 애플리케이션의 품질을 보장하기 위해 사전 프로덕션, 베타 및 테스트 플릿을 생성해야 하는 경우가 많습니다. 이 테스트 하드웨어를 최대한 활용할 수 있도록 다양한 최적화를 수행할 수 있지만 이러한 병렬 플릿이 항상 최적으로 사용되는 것은 아니며 많은 고가의 하드웨어가 오랫동안 사용되지 않습니다.

AWS 클라우드에서는 필요할 때 테스트 플릿을 프로비저닝할 수 있습니다. 따라서 실제 사용 며칠 또는 몇 달 전에 리소스를 사전 프로비저닝할 필요가 없어질 뿐만 아니라 인프라 구성 요소가 필요하지 않을 때는 유연하게 해체할 수 있습니다. 또한 로드 테스트 중에 AWS 클라우드에서 사용자 트래픽을 시뮬레이션할 수 있습니다. 이러한 병렬 플릿을 새 프로덕션 릴리스의 스테이징 환경으로 사용할 수도 있습니다. 따라서 서비스 중단이 거의 또는 전혀 없이 현재 운영 환경에서 새 애플리케이션 버전으로 신속하게 전환할 수 있습니다.

웹 호스팅을 위한 AWS 클라우드 아키텍처

다음 그림은 이 기존 웹 애플리케이션 아키텍처가 AWS 클라우드 컴퓨팅 인프라를 어떻게 활용할 수 있는지 다시 보여줍니다.


www.example.com
media.example.com
AWS Cloud
media.example.com
www.example.com
AWS Shield
DDoS protection
AWS WAF
Web Application Firewall
Amazon Route 53
DNS service
Amazon CloudFront
CDN
AWS WAF
Web Application Firewall
Amazon S3
Static storage and backup
Availability Zone 1
Availability Zone 2
VPC
Public subnet
Nat gateway
Public subnet
Nat gateway
Elastic Load Balancing
Web Subnet
Web Subnet
EC2 Instances
Web servers
Auto Scaling group
Web tier
EC2 Instances
Web servers
Elastic Load Balancing
App Subnet
App Subnet
Amazon EFS
Shared storage
EC2 Instances
App servers
Auto Scaling group
App tier
EC2 Instances
App servers
DB Subnet
DB Subnet
DB subnet group
Amazon ElastiCache Service
Caching service
Amazon RDS
Database (primary)
Amazon RDS
Database (secondary)

AWS 기반 웹 호스팅 아키텍처의 사례

1. Amazon Route 53를 사용하는 DNS 서비스 - 도메인 관리를 간소화하는 DNS 서비스를 제공합니다.
2. Amazon CloudFront를 사용한 엣지 캐싱 - 엣지는 대용량 콘텐츠를 캐싱하여 고객의 대기 시간을 줄입니다.
3. AWS WAF를 사용한 Amazon CloudFront용 엣지 보안 - 고객 정의 규칙을 통해 크로스 사이트 스크립팅(XSS) 및 SQL 삽입을 비롯한 악성 트래픽을 필터링합니다.
4. Elastic Load Balancing(ELB)을 사용한 로드 밸런싱 – 서비스의 중복성과 분리를 위해 여러 가용 영역 및 AWS Auto Scaling 그룹에 로드를 분산할 수 있습니다.
5. AWS Shield를 사용한 DDoS 방어 - 가장 일반적인 네트워크 및 전송 계층 DDoS 공격으로부터 자동으로 인프라를 보호합니다.
6. 보안 그룹이 포함된 방화벽 - 보안을 인스턴스로 이동하여 웹 및 애플리케이션 서버 모두에 상태 유지 호스트 수준 방화벽을 제공합니다.
7. Amazon ElastiCache를 사용한 캐싱 - Redis 또는 Memcached와 함께 캐싱 서비스를 제공하여 앱과 데이터베이스에서 로드를 제거하고 빈번한 요청의 대기 시간을 줄입니다.
8. Amazon Relational Database Service(Amazon RDS)를 사용한 관리형 데이터베이스 - 6개 DB 엔진 중에서 사용하여 고가용성 다중 AZ 데이터베이스 아키텍처를 생성합니다.
9. Amazon Simple Storage Service(Amazon S3)를 통한 정적 스토리지 및 백업 - 이미지, 비디오와 같은 정적 자산 및 백업을 위한 간단한 HTTP 기반 객체 스토리지를 구현합니다.

AWS 웹 호스팅 아키텍처의 핵심 구성 요소

다음 섹션에서는 AWS 클라우드에 배포된 웹 호스팅 아키텍처의 주요 구성 요소 중 일부를 간략하게 설명하고 기존 웹 호스팅 아키텍처와 어떻게 다른지 설명합니다.

네트워크 관리

AWS 클라우드에서는 네트워크를 다른 고객의 네트워크와 분할할 수 있으므로 보다 안전하고 확장 가능한 아키텍처를 구현할 수 있습니다. 보안 그룹은 호스트 수준의 보안을 제공하지만(호스트 보안 섹션 참조), Amazon Virtual Private Cloud(Amazon VPC)를 사용하면 사용자가 정의한 논리적으로 격리된 가상 네트워크에서 리소스를 시작할 수 있습니다.

Amazon VPC는 AWS에서 네트워킹 설정 세부 사항을 완벽하게 제어할 수 있는 서비스입니다. 이러한 제어의 예로는 웹 서버용 퍼블릭 서브넷과 데이터베이스에 대한 인터넷 액세스가 없는 프라이빗 서브넷을 만드는 것이 있습니다. 또한 Amazon VPC를 사용하면 하드웨어 가상 사설 네트워크(VPN)를 사

용하여 하이브리드 아키텍처를 생성하고 AWS 클라우드를 자체 데이터 센터의 확장으로 사용할 수 있습니다.

또한 Amazon VPC에는 네트워크에 대한 IPv4 지원 외에도 IPv6 지원이 포함되어 있습니다.

콘텐츠 전송

웹 트래픽이 지리적으로 분산되어 있는 경우, 전 세계에 걸쳐 전체 인프라를 복제하는 방식이 현실적으로 불가능할 수도 있고 비용상 효율적이지도 않습니다. 콘텐츠 전송 네트워크(CDN)는 글로벌 엣지 로케이션 네트워크를 활용하여 동영상, 웹 페이지, 이미지 등 웹 콘텐츠의 캐싱된 복사본을 고객에게 제공하는 기능을 제공합니다. 응답 시간을 줄이기 위해 CDN은 고객과 가장 가까운 엣지 로케이션 또는 원래 요청 위치를 활용함으로써 응답 시간을 단축합니다. 웹 자산이 캐시에서 제공되므로 처리량이 대폭 늘어납니다. 동적 데이터의 경우 많은 수의 CDN을 구성하여 오리진 서버에서 데이터를 검색할 수 있습니다.

CloudFront를 사용하여 글로벌 네트워크의 엣지 로케이션을 통해 동적, 정적 및 스트리밍 콘텐츠를 포함하는 웹 사이트를 전송할 수 있습니다. CloudFront는 자동으로 콘텐츠에 대한 요청을 가장 가까운 엣지 로케이션으로 라우팅하므로 콘텐츠가 가능한 최상의 성능으로 전송됩니다. CloudFront는 Amazon S3 및 Amazon Elastic Compute Cloud(Amazon EC2) 같은 다른 AWS 서비스와 연동하도록 최적화되어 있습니다. CloudFront는 AWS 오리진 서버는 아니지만 파일의 원본 버전을 저장하는 다른 모든 오리진 서버에서도 원활하게 작동합니다.

다른 AWS 서비스와 마찬가지로 CloudFront 사용에 대한 계약 또는 월간 약정은 없습니다. 서비스를 통해 실제로 전송하는 콘텐츠 만큼만 비용을 지불하면 됩니다.

또한 웹 애플리케이션 인프라의 엣지 캐싱을 위한 기존 솔루션이 AWS 클라우드에서 제대로 작동해야 합니다.

퍼블릭 DNS 관리

웹 애플리케이션을 AWS 클라우드로 이동하려면 도메인 이름 시스템(DNS)을 일부 변경해야 합니다. DNS 라우팅을 관리할 수 있도록 AWS는 가용성 및 확장성이 뛰어난 클라우드 DNS 웹 서비스인 Amazon Route 53를 제공합니다. Route 53는 최종 사용자를 인터넷 애플리케이션으로 라우팅할 수 있는 매우 안정적이고 비용 효율적인 방법을 개발자와 기업에 제공하기 위해 설계되었습니다. 이 서비스에서는 'www.example.com'과 같은 이름을 192.0.2.1과 같은 컴퓨터 간 연결에 사용되는 숫자 IP 주소로 변환합니다. 또한 Route 53는 IPv6와도 완벽히 호환됩니다.

호스트 보안

AWS는 엣지에서의 인바운드 네트워크 트래픽 필터링 외에도 웹 애플리케이션이 호스트 수준에서 네트워크 트래픽 필터링을 적용할 것을 권장합니다. Amazon EC2는 보안 그룹이라는 기능을 제공합니다. 보안 그룹은 인바운드 네트워크 방화벽과 유사하며, 이를 위해 EC2 인스턴스에 도달할 수 있는 프로토콜, 포트 및 소스 IP 범위를 지정할 수 있습니다.

각 EC2 인스턴스에 하나 이상의 보안 그룹을 할당할 수 있습니다. 각 보안 그룹은 각 인스턴스에 적절한 트래픽을 허용합니다. 특정 서브넷, IP 주소 및 리소스만 EC2 인스턴스에 액세스할 수 있도록 보안 그룹을 구성할 수 있습니다. 또는 다른 보안 그룹을 참조하여 특정 그룹에 있는 EC2 인스턴스에 대한 액세스를 제한할 수 있습니다.

그림 3의 AWS 웹 호스팅 아키텍처에서 웹 서버 클러스터의 보안 그룹은 웹 계층 로드 밸런스에서만 액세스를 허용하고 포트 80 및 443(HTTP 및 HTTPS)에서 TCP를 통해서만 액세스를 허용할 수 있습니다. 반면 애플리케이션 서버 보안 그룹은 애플리케이션 계층 로드 밸런서에서만 액세스를 허용할 수 있습니다. 이 모델에서 지원 엔지니어도 EC2 인스턴스에 액세스해야 하는데, 이는 AWS Systems Manager 세션 관리자를 통해 달성할 수 있습니다. 보안에 대한 자세한 내용은 AWS의 보안 기능을 설명하는 보안 게시판, 인증 정보 및 보안 백서가 포함된 AWS 클라우드 보안을 참조하십시오.

클러스터 간 로드 밸런싱

하드웨어 로드 밸런서는 기존 웹 애플리케이션 아키텍처에서 사용되는 일반적인 네트워크 어플라이언스입니다. AWS는 Elastic Load Balancing(ELB) 서비스를 통해 이 기능을 제공합니다. ELB는 EC2 인스턴스, 컨테이너, IP 주소, AWS Lambda 함수, 가상 어플라이언스와 같은 여러 대상에 걸쳐 수신 애플리케이션 트래픽을 자동으로 분산합니다. Elastic Load Balancing은 단일 가용 영역 또는 여러 가용 영역에서 다양한 애플리케이션 부하를 처리할 수 있습니다. Elastic Load Balancing이 제공하는 네 가지 로드 밸런서는 모두 애플리케이션의 내결함성에 필요한 고가용성, 자동 크기 조정, 강력한 보안을 갖추고 있습니다.

다른 호스팅 및 서비스 검색

기존 웹 호스팅 아키텍처에서는 대부분의 호스트에 고정 IP 주소가 있습니다. AWS 클라우드에서는 대부분의 호스트에 동적 IP 주소가 있습니다. 모든 EC2 인스턴스는 퍼블릭 및 프라이빗 DNS 항목을 둘 다 가질 수 있고 인터넷 상에서 주소를 지정할 수 있지만, 인스턴스를 시작할 때 DNS 항목과 IP 주소가 동적으로 할당됩니다. 수동으로 할당할 수는 없습니다. 고정 IP 주소(AWS 용어에서는 탄력적 IP 주소)는 인스턴스가 시작된 후 실행 중인 인스턴스에 할당할 수 있습니다. 프라이머리 데이터베이스, 중앙 파일 서버, EC2 호스팅 로드 밸런서와 같이 일관된 엔드포인트가 필요한 인스턴스 및 서비스에는 탄력적 IP 주소를 사용해야 합니다.

웹 애플리케이션 내에서의 캐싱

인 메모리 애플리케이션 캐시는 자주 사용하는 정보를 캐시하여 서비스에 대한 부하를 줄이고 데이터베이스 계층의 성능과 확장성을 향상시킬 수 있습니다. Amazon ElastiCache는 클라우드에서 인 메모리 캐시를 손쉽게 배포, 운영 및 조정할 수 있게 해주는 웹 서비스입니다. 로드에 따라 자동으로 크기가 조정되고 장애가 발생한 노드를 자동으로 교체하도록 생성한 인 메모리 캐시를 구성할 수 있습니다. ElastiCache는 Memcached 및 Redis와 프로토콜이 호환되므로 현재 온프레미스 솔루션에서 간단하게 마이그레이션할 수 있습니다.

데이터베이스 구성, 백업 및 장애 조치

많은 웹 애플리케이션에는 일반적으로 관계형 또는 비관계형 데이터베이스의 형태로 일정한 형태의 지속성이 포함되어 있습니다. AWS는 관계형 및 비관계형 데이터베이스 서비스를 모두 제공합니다. 또는 EC2 인스턴스에 자체 데이터베이스 소프트웨어를 배포할 수 있습니다. 다음 표에는 이러한 옵션이 요약되어 있으며 이 섹션에서 자세히 설명합니다.

표 1 - 관계형 및 비관계형 데이터베이스 솔루션

	관계형 데이터베이스 솔루션	NoSQL 솔루션
관리형 데이터베이스 서비스	Amazon RDS for MySQL, Oracle, SQL Server, MariaDB, PostgreSQL, Amazon Aurora	Amazon DynamoDB, Amazon Keyspaces, Amazon Neptune, Amazon QLDB, Amazon Timestream
자체 관리형	Amazon EC2 인스턴스에서 관계형 데이터베이스 관리 시스템(DBMS) 호스팅	EC2 인스턴스에서 비관계형 데이터베이스 솔루션 호스팅

Amazon RDS

Amazon Relational Database Service(Amazon RDS)를 사용하면 친숙한 MySQL, PostgreSQL, Oracle 및 Microsoft SQL Server 데이터베이스 엔진의 기능에 액세스할 수 있습니다. 이미 사용 중인 코드, 애플리케이션 및 도구를 Amazon RDS와 함께 사용할 수 있습니다. Amazon RDS는 자동으로 데이터베이스 소프트웨어를 패치하고 데이터베이스를 백업하며, 사용자가 정의한 보존 기간 동안 백업을 저장합니다. 또한 특정 시점으로 복구도 지원합니다. 단일 API 호출로 관계형 데이터베이스 인스턴스와 연결된 컴퓨팅 리소스 또는 스토리지 용량을 유연하게 확장할 수 있습니다.

Amazon RDS 다중 AZ 배포는 데이터베이스 가용성을 높이고 예기치 않은 중단으로부터 데이터베이스를 보호합니다. Amazon RDS 읽기 전용 복제본은 데이터베이스의 읽기 전용 복제본을 제공하므로 단일 데이터베이스 배포의 용량 이상으로 확장하여 읽기 중심의 데이터베이스 워크로드를 처리할 수 있습니다. 모든 AWS 서비스와 마찬가지로 사전 투자가 필요하지 않으며 사용한 리소스에 대해서만 비용을 지불하면 됩니다.

Amazon EC2 인스턴스에서 관계형 데이터베이스 관리 시스템(RDBMS) 호스팅

관리형 Amazon RDS 서비스 외에도 EC2 인스턴스에 원하는 RDBMS(예: MySQL, Oracle, SQL Server 또는 DB2)를 설치하고 직접 관리할 수 있습니다. Amazon EC2에서 데이터베이스를 호스트하는 AWS 고객은 읽기 전용 복사본에 대한 미러링 및 상시 준비 패시브 슬레이브에 대한 로그 전달을 포함하여 다양한 프라이머리/스탠바이 및 복제 모델을 성공적으로 사용합니다.

Amazon EC2에서 직접 데이터베이스 소프트웨어를 관리하는 경우 내결함성 영구 스토리지의 가용성도 고려해야 합니다. 이를 위해 Amazon EC2에서 실행되는 데이터베이스는 네트워크 연결 스토리지와 유사한 Amazon Elastic Block Store(Amazon EBS) 볼륨을 사용하는 것이 좋습니다.

데이터베이스를 실행하는 EC2 인스턴스의 경우 모든 데이터베이스 데이터와 로그를 EBS 볼륨에 배치해야 합니다. 이들은 데이터베이스 호스트에 장애가 발생하더라도 계속 사용할 수 있습니다. 이 구성을 사용하면 호스트에 장애가 발생할 경우 새 EC2 인스턴스를 시작하고 기존 EBS 볼륨을 새 인스턴스에 연결할 수 있는 간단한 장애 조치 시나리오가 가능합니다. 그러면 데이터베이스가 중단된 부분부터 다시 시작할 수 있습니다.

EBS 볼륨은 가용 영역 내에서 자동으로 중복성을 제공합니다. 단일 EBS 볼륨의 성능이 데이터베이스 요구 사항에 충분하지 않은 경우 볼륨을 스트라이프하여 데이터베이스의 IOPS(초당 입출력 작업 수) 성능을 높일 수 있습니다.

까다로운 워크로드의 경우 사용자가 필요한 IOPS를 지정하는 EBS 프로비저닝된 IOPS를 사용할 수도 있습니다. Amazon RDS를 사용하는 경우 서비스에서 자체 스토리지를 관리하므로 사용자는 데이터 관리에 집중할 수 있습니다.

비관계형 데이터베이스

AWS는 관계형 데이터베이스에 대한 지원 외에도 다음과 같은 여러 관리형 비관계형 데이터베이스도 제공합니다.

- Amazon DynamoDB는 완전관리형 NoSQL 데이터베이스 서비스로서 원활한 확장성과 함께 빠르고 예측 가능한 성능을 제공합니다. AWS Management Console 또는 DynamoDB API를 사용하면 가동 중지 시간이나 성능 저하 없이 용량을 확장 또는 축소할 수 있습니다. DynamoDB는 분산 데이터베이

스를 운영하고 조정하는 데 따른 관리 부담을 AWS로 전가하므로 하드웨어 할당, 설정 및 구성, 복제, 소프트웨어 패치 또는 클러스터 조정에 대해 걱정할 필요가 없습니다.

- Amazon DocumentDB(MongoDB 호환)는 대규모 JSON 데이터 관리를 위해 특별히 설계된 데이터베이스 서비스로, 완전관리형이고 AWS에서 실행되며 탁월한 내구성을 바탕으로 엔터프라이즈급 성능을 제공합니다.
- Amazon Keyspaceshttps://aws.amazon.com/keyspaces/(Apache Cassandra용)는 확장성과 가용성이 뛰어나며 관리형 Apache Cassandra와 호환되는 데이터베이스 서비스입니다. Amazon Keyspaces를 사용하면 현재 사용 중인 것과 동일한 Cassandra 애플리케이션 코드 및 개발자 도구를 사용하여 AWS에서 Cassandra 워크로드를 실행할 수 있습니다.
- Amazon Neptune은 빠르고 안정적인 완전관리형 그래프 데이터베이스 서비스로, 상호연결성이 높은 데이터 집합을 활용하는 애플리케이션을 손쉽게 구축 및 운영할 수 있습니다. Amazon Neptune은 한마디로 수십억 개의 관계를 저장하고 불과 몇 밀리초의 지연 시간으로 그래프를 쿼리하는 데 최적화된, 특수 목적의 고성능 그래프 데이터베이스 엔진입니다.
- Amazon Quantum Ledger Database(QLDB)는 완전관리형 원장 데이터베이스로, 신뢰할 수 있는 중앙 기관에서 소유하는 투명하고 변경 불가능하며 암호화 방식으로 검증 가능한 트랜잭션 로그를 제공합니다. QLDB는 모든 애플리케이션 데이터 변경 내용을 추적하며 완전하고 검증 가능한 시간대별 변경 기록을 유지합니다.
- Amazon Timestream는 IoT 및 운영 애플리케이션으로 제공되는 확장이 용이한 고속 서버리스 시계열 데이터베이스 서비스로, 관계형 데이터베이스에 비해 최대 10배나 저렴한 비용으로 1,000배 더 빠르게 매일 수조 건의 이벤트를 쉽게 저장하고 분석할 수 있습니다.

또한 Amazon EC2를 사용하여 다른 비관계형 데이터베이스 기술을 호스트할 수도 있습니다.

데이터 및 자산에 대한 스토리지 및 백업

AWS 클라우드에는 웹 애플리케이션 데이터 및 자산을 저장, 액세스 및 백업하는 다양한 방법들이 있습니다. Amazon S3는 고가용성 중복 객체 스토어를 제공합니다. Amazon S3는 이미지, 동영상 및 기타 정적 미디어와 같이 다소 정적이거나 느리게 변화하는 객체를 위한 훌륭한 스토리지 솔루션입니다. 또한 Amazon S3는 CloudFront와 상호 작용하여 이러한 자산의 엣지 캐싱 및 스트리밍을 지원합니다.

연결된 파일 시스템과 유사한 스토리지를 위해 EC2 인스턴스에 EBS 볼륨을 연결할 수 있습니다. 이는 EC2 인스턴스를 실행하기 위한 탑재 가능한 디스크처럼 작동합니다. Amazon EBS는 블록 스토리지로 액세스해야 하고 데이터베이스 파티션 및 애플리케이션 로그와 같이 실행 중인 인스턴스의 수명 이상으로 지속성이 필요한 데이터에 적합합니다.

EC2 인스턴스와 독립적인 수명 주기를 갖는 외에도 EBS 볼륨의 스냅샷을 생성하여 Amazon S3에 저장할 수 있습니다. EBS 스냅샷은 이전 스냅샷 이후의 변경 사항만 백업하므로 스냅샷 빈도가 높을수록

스냅샷 시간을 줄일 수 있습니다. 또한 데이터를 여러 EBS 볼륨에 복제하고 해당 볼륨을 실행 중인 다른 인스턴스에 연결하기 위한 기준으로 EBS 스냅샷을 사용할 수도 있습니다.

EBS 볼륨은 최대 16TB까지 확장할 수 있으며, 더 큰 볼륨 또는 향상된 I/O(입출력) 성능을 위해 여러 EBS 볼륨을 스트라이프할 수 있습니다. I/O 집약적 애플리케이션의 성능을 극대화하기 위해 프로비저닝된 IOPS 볼륨을 사용할 수 있습니다. 프로비저닝된 IOPS 볼륨은 성능에 민감하고 임의 액세스 I/O 처리량이 일정한 I/O 집약적 워크로드, 특히 데이터베이스 워크로드 요구 사항을 충족하도록 설계되었습니다.

볼륨을 생성할 때 IOPS 속도를 지정하면 Amazon EBS가 볼륨의 수명 주기 동안 해당 속도를 프로비저닝합니다. Amazon EBS는 현재 최대 16,000(모든 인스턴스 유형)에서 최대 64,000(Nitro 시스템에 구축된 인스턴스)에 이르는 볼륨당 IOPS를 지원합니다. 여러 볼륨을 함께 스트라이프하여 애플리케이션에 인스턴스당 수천 IOPS를 제공할 수 있습니다. 이 외에도 더 높은 처리량과 1밀리초 미만의 대기 시간이 필요한 미션 크리티컬 워크로드의 경우 최대 64TB의 스토리지 용량으로 최대 256,000 IOPS를 지원할 수 있는 io2 block express 볼륨 유형을 사용할 수 있습니다.

플릿 자동 크기 조정

AWS 클라우드 아키텍처와 기존 호스팅 모델의 주요 차이점 중 하나는 AWS가 트래픽 변화를 처리하기 위해 필요에 따라 웹 애플리케이션 플릿의 크기를 자동으로 조정할 수 있다는 것입니다. 일반적으로 기존 호스팅 모델에서는 예상되는 트래픽의 양에 맞춰 호스트의 양을 미리 프로비저닝하는 트래픽 예측 모델을 사용합니다. AWS에서는 플릿을 확장 및 축소하기 위한 일련이 트리거에 따라 즉시 인스턴스를 프로비저닝할 수 있습니다.

Auto Scaling 서비스는 필요에 따라 확장 또는 축소할 수 있는 서버의 용량 그룹을 생성할 수 있습니다. 또한 Auto Scaling은 지표 데이터를 위해 CloudWatch와 직접 연동되며 Elastic Load Balancing과 연동되어 로드 분산을 위한 호스트를 추가 및 제거합니다. 예를 들어 웹 서버가 일정 기간 동안 80% 이상의 CPU 사용률을 보고하는 경우 추가 웹 서버를 신속하게 배포한 다음 로드 밸런서에 자동으로 추가하여 로드 밸런싱 회전에 즉시 포함할 수 있습니다.

AWS 웹 호스팅 아키텍처 모델에서 볼 수 있듯이 아키텍처의 여러 계층에서 여러 Auto Scaling 그룹을 생성하여 각 계층을 독립적으로 확장할 수 있습니다. 예를 들어 웹 서버 Auto Scaling 그룹은 네트워크 I/O의 변경에 따라 확장 및 축소가 트리거되는 반면 애플리케이션 서버 Auto Scaling 그룹은 CPU 사용률에 따라 확장 및 축소할 수 있습니다. 최소값 및 최대값을 설정하여 24/7 가용성을 보장하고 그룹 내 사용량을 제한할 수 있습니다.

Auto Scaling 트리거는 리소스 사용률을 실제 수요에 맞추기 위해 지정된 계층에서 전체 플릿을 확장하고 축소하도록 설정할 수 있습니다. Auto Scaling 서비스 외에도 인스턴스를 시작, 종료 및 검사할 수 있는 Amazon EC2 API를 통해 직접 Amazon EC2 플릿의 크기를 조정할 수 있습니다.

추가 보안 기능

분산 서비스 거부(DDoS) 공격이 횟수와 정교함이 증가하고 있습니다. 기본적으로 이러한 공격은 방어하기가 어렵습니다. 공격 중 웹 사이트 방문 손실로 인한 기회 비용뿐만 아니라 완화 시간과 전력 소비 모두에서 비용이 많이 드는 경우가 많습니다. 이러한 공격으로부터 방어하는 데 도움이 될 수 있는 여러 AWS 요인 및 서비스가 있습니다. 그 가운데 하나가 AWS 네트워크의 규모입니다. AWS 인프라는 상당한 규모이므로 AWS의 규모를 활용하여 방어를 최적화할 수 있습니다. Elastic Load Balancing, Amazon CloudFront, Amazon Route 53 등 여러 서비스는 트래픽의 대폭 증가에 대응하여 웹 애플리케이션을 확장하는 데 효과적입니다.

인프라 보호 서비스는 특히 방어 전략에 도움이 됩니다.

- AWS Shield는 다양한 형태의 DDoS 공격 벡터로부터 보호하는 데 도움이 되는 관리형 DDoS 보호 서비스입니다. AWS Shield의 표준 서비스는 무료이며 계정 전체에서 자동으로 활성화됩니다. 이 표준 서비스는 가장 일반적인 네트워크 및 전송 계층 공격으로부터 방어하는 데 도움이 됩니다. 이 수준 외에도 고급 서비스는 진행 중인 공격에 대한 거의 실시간 가시성을 제공하고 앞서 언급한 서비스와 더 상위 수준에서 통합함으로써 웹 애플리케이션에 대한 더 높은 수준의 보호를 제공합니다. 또한 AWS DDoS 대응 팀(DRT)에 액세스하여 리소스에 대한 대규모의 정교한 공격을 완화할 수 있습니다.
- AWS WAF(웹 애플리케이션 방화벽)는 가용성 또는 보안을 손상시키거나 과도한 리소스를 소비할 수 있는 공격으로부터 웹 애플리케이션을 보호하도록 설계되었습니다. AWS WAF는 CloudFront 또는 Application Load Balancer와 연동하고 사용자 지정 규칙과 함께 크로스 사이트 스크립팅, SQL 삽입, DDoS와 같은 공격으로부터 방어합니다. 대부분의 AWS 서비스와 마찬가지로, AWS WAF에는 보안 요구 사항이 변경됨에 따라 AWS WAF 인스턴스에 대한 규칙 생성 및 편집을 자동화할 수 있는 모든 기능을 갖춘 API가 함께 제공됩니다.
- AWS Firewall Manager는 AWS Organizations의 계정 및 애플리케이션 전반에 걸쳐 방화벽 규칙을 중앙에서 구성 및 관리할 수 있는 보안 관리 서비스입니다. 새 애플리케이션이 생성되면 AWS Firewall Manager는 공통 보안 규칙 세트를 적용하여 새로운 애플리케이션 및 리소스에 손쉽게 규정 준수를 적용할 수 있습니다.

AWS를 사용한 장애 조치

기존 웹 호스팅에 비해 AWS의 또 다른 주요 이점은 중복 배포 위치에 쉽게 액세스할 수 있는 가용 영역입니다. 각 가용 영역은 다른 가용 영역에서 발생한 장애로부터 격리되도록 설계된 물리적으로 분리된 위치입니다. 가용 영역은 같은 AWS 리전에 있는 다른 가용 영역에 대해 저렴하고 대기 시간이 짧은 네트워크 연결을 제공합니다. AWS 웹 호스팅 아키텍처 다이어그램에서 볼 수 있듯이 EC2 호스트를 여러 가용 영역에 배포하여 웹 애플리케이션의 내결함성을 높이는 것이 좋습니다.

장애 발생 시 가용 영역 전체에 단일 액세스 지점을 마이그레이션하기 위한 프로비저닝이 있는 것이 중요합니다. 예를 들어, 예기치 않은 오류 시나리오에서도 데이터 지속성이 일관되고 고가용성을 유지할 수 있도록 두 번째 가용 영역에 스탠바이 데이터베이스를 설정해야 합니다. Amazon EC2 또는 Amazon RDS에서 버튼 클릭 한 번으로 이 작업을 수행할 수 있습니다.

기존 웹 애플리케이션을 AWS 클라우드로 이전할 때 일부 아키텍처 변경이 필요한 경우가 많지만 확장성, 안정성 및 비용 효율성이 크게 향상되므로 AWS 클라우드를 사용하기 위해 들이는 노력은 충분한 가치가 있습니다. 다음 섹션에서는 이러한 개선 사항에 대해 설명합니다.

웹 호스팅에 AWS 사용 시 주요 고려 사항

AWS 클라우드와 기존 웹 애플리케이션 호스팅 모델 간에는 몇 가지 주요 차이점이 있습니다. 이전 섹션에서는 클라우드에 웹 애플리케이션을 배포할 때 고려해야 할 여러 주요 영역을 살펴봤습니다. 이 섹션에서는 애플리케이션을 클라우드로 가져올 때 고려해야 할 몇 가지 주요 아키텍처 변화에 대해 설명합니다.

물리적 네트워크 어플라이언스의 제거

AWS에서는 물리적 네트워크 어플라이언스를 배포할 수 없습니다. 예를 들어 AWS 애플리케이션의 방화벽, 라우터, 로드 밸런서는 더 이상 물리적 디바이스에 상주할 수 없고 소프트웨어 솔루션으로 대체해야 합니다. 로드 밸런싱 또는 VPN 연결 설정 등 다양한 엔터프라이즈급 소프트웨어 솔루션이 있습니다. 이는 AWS 클라우드에서 실행할 수 있는 기능에 대한 제한은 아니지만, 현재 이러한 디바이스를 사용하는 경우 애플리케이션 아키텍처를 변경해야 합니다.

어디에나 있는 방화벽

과거에는 단순한 완충 영역(DMZ)을 구축해 호스트 간에 커뮤니케이션을 개통했지만, AWS는 모든 호스트가 차단되는 보다 안전한 모델을 적용합니다. AWS 배포를 계획하는 단계 중 하나는 호스트 간 트래픽 분석입니다. 이 분석은 정확히 어떤 포트를 열어야 하는지에 대한 결정을 안내합니다. 아키텍처의 각 호스트 유형에 대한 보안 그룹을 생성할 수 있습니다. 또한 단순한 계층형 보안 모델을 매우 다양하게 생성하여 아키텍처 내 호스트 간의 액세스를 최소화할 수 있습니다. Amazon VPC 내에서 네트워크 액세스 제어 목록을 사용하면 서브넷 수준에서 네트워크를 잠글 수 있습니다.

여러 데이터 센터의 가용성 고려

AWS 리전 내의 가용 영역을 여러 데이터 센터로 생각하십시오. 서로 다른 가용 영역에 있는 EC2 인스턴스는 논리적으로 또한 물리적으로 분리되어 있으며, 고가용성 및 안정성을 확보하기 위해 여러 데이터 센터에 애플리케이션을 배포할 수 있는 사용하기 쉬운 모델을 제공합니다. 리전별 서비스로서의 Amazon VPC를 사용하면 가용 영역을 활용하는 동시에 모든 리소스를 동일한 논리적 네트워크에 유지할 수 있습니다.

한시적인 자동증감 호스트

AWS 애플리케이션을 설계하는 방식에서 가장 중요한 변화는 Amazon EC2 호스트가 자동으로 증감될 가능성이 있기 때문에 그 수명이 한시적이라는 사실입니다. AWS 클라우드용으로 구축된 애플리케

이션은 호스트를 항상 사용할 수 있다고 가정해서는 안 되며 EC2 인스턴스에 장애가 발생할 경우 EC2 인스턴트 스토어의 데이터가 손실된다는 사실을 인식하고 설계해야 합니다.

새 호스트를 불러올 때 호스트의 가용 영역 내의 IP 주소 또는 위치를 가정해서는 안 됩니다. 구성 모델은 유연해야 하며 호스트 부트스트래핑에 대한 접근 방식에서는 클라우드의 동적 특성을 고려해야 합니다. 이러한 기술은 확장성 및 내결함성이 뛰어난 애플리케이션을 구축하고 실행하는 데 중요합니다.

컨테이너 및 서버리스 고려

이 백서는 주로 보다 전통적인 웹 아키텍처에 초점을 맞추고 있습니다. 그러나 컨테이너 및 서버리스 기술로 전환하고 AWS Fargate 및 AWS Lambda 같은 서비스를 활용하여 웹 애플리케이션을 현대화하는 것을 고려해 보십시오. 그러면 컴퓨팅 작업을 수행하기 위한 가상 머신 사용을 추상화할 수 있습니다. 서버리스 컴퓨팅을 사용하면 AWS가 용량 프로비저닝 및 패치 적용과 같은 인프라 관리 작업을 처리하므로 보다 민첩한 애플리케이션을 구축하여 더 신속하게 혁신에 대응할 수 있습니다.

자동화된 배포 고려

- Amazon Lightsail은 애플리케이션 또는 웹 사이트를 구축하는 데 필요한 모든 것을 제공하는 사용하기 쉬운 가상 프라이빗 서버(VPS)로, 매월 경제적인 요금 플랜도 지원합니다. Lightsail은 단순한 워크로드 및 빠른 배포는 물론, AWS에서 시작할 때 매우 적합합니다. 처음에 작은 규모로 시작했다가 비즈니스 성장에 따라 확장할 수 있도록 설계되었습니다.
- AWS Elastic Beanstalk는 Java, .NET, PHP, Node.js, Python, Ruby, Go 및 Docker를 사용하여 개발된 웹 애플리케이션 및 서비스를 Apache, NGINX, Passenger, IIS와 같은 친숙한 서버에 간편하게 배포하고 확장할 수 있는 서비스입니다. 코드를 업로드하기만 하면 Elastic Beanstalk가 배포, 용량 프로비저닝, 로드 밸런싱, 자동 크기 조정, 애플리케이션 상태 모니터링을 자동으로 처리합니다. 이뿐만 아니라 애플리케이션을 실행하는 데 필요한 AWS 리소스를 완벽하게 제어할 수 있으며 언제든지 기본 리소스에 액세스할 수 있습니다.
- AWS App Runner는 개발자가 사전 인프라 경험 없이도 컨테이너화된 웹 애플리케이션 및 API를 대규모로 빠르게 배포할 수 있도록 지원하는 완전관리형 서비스입니다. 소스 코드 또는 컨테이너 이미지로 시작해보세요. App Runner는 웹 애플리케이션을 자동으로 구축 및 배포하고 암호화를 통해 트래픽의 로드 밸런싱을 수행합니다. 또한, App Runner는 트래픽 요구 사항을 충족하도록 자동으로 확장 또는 축소됩니다.
- AWS Amplify는 모바일 및 프런트 엔드 웹 개발자가 AWS에서 제공하는 확장 가능한 풀 스택 애플리케이션을 구축하도록 지원하는 함께 혹은 단독으로 사용 가능한 도구 및 서비스 집합입니다. Amplify를 사용하면 몇 분 만에 앱 백엔드를 구성하고 앱을 연결하며, 클릭 몇 번 만으로 정적 웹 앱을 배포하고, AWS Management Console 외부 앱 콘텐츠를 쉽게 관리할 수 있습니다.

결론 및 기여자

결론

웹 애플리케이션을 AWS 클라우드로 마이그레이션하기 위해 숙고하는 과정에서 많은 아키텍처 및 개념적 고려 사항이 있습니다. 비즈니스와 함께 성장하는 비용 효율적이고 확장성이 뛰어나며 내결함성을 갖춘 인프라를 보유하여 얻을 수 있는 이점은 AWS 클라우드로 마이그레이션하는 노력을 훨씬 능가합니다.

기여자

다음은 본 문서를 작성하는 데 도움을 준 개인 및 조직입니다.

- Amir Khairalomoum, 선임 솔루션스 아키텍트, AWS
- Dinesh Subramani, 선임 솔루션스 아키텍트, AWS
- Jack Hemion, 선임 솔루션스 아키텍트, AWS
- Jatin Joshi, 클라우드 지원 엔지니어, AWS
- Jorge Fonseca, 선임 솔루션스 아키텍트, AWS
- Shinduri K S, 솔루션스 아키텍트, AWS

추가 자료

- Amazon Lightsail에 Django 기반 애플리케이션 배포
- Elastic Beanstalk에 고가용성 Drupal 웹 사이트 배포
- Elastic Beanstalk에 고가용성 PHP 애플리케이션 배포
- Elastic Beanstalk에 DynamoDB를 사용하는 Node.js 애플리케이션 배포
- AWS 클라우드에서 Linux 웹 애플리케이션 시작하기
- 정적 웹 사이트 호스팅
- Amazon S3를 사용하여 정적 웹 사이트 호스팅
- 자습서: Elastic Beanstalk를 사용하여 ASP.NET Core 애플리케이션 배포
- 자습서: Elastic Beanstalk를 사용하여 .NET 샘플 애플리케이션을 배포하는 방법

문서 개정

이 백서의 업데이트에 대한 알림을 받으려면 RSS 피드를 구독하십시오.

업데이트 기록-변경	update-history-description	update-history-date
백서 업데이트됨	여러 섹션 및 다이어그램이 새로운 서비스, 기능 및 업데이트된 서비스 한도로 업데이트되었습니다.	2021년 8월 20일
백서 업데이트됨	그림 3에서 ‘ElastiCache를 사용한 캐싱’에 대한 아이콘 레이블이 업데이트되었습니다.	2019년 9월 29일
백서 업데이트됨	새로운 서비스 때문에 여러 섹션이 추가되고 업데이트되었습니다. 명확성을 개선하고 서비스를 추가하기 위해 다이어그램이 업데이트되었습니다. ‘네트워크 관리’에서 AWS의 표준 네트워킹 방법으로 VPC가 추가되었습니다. ‘추가 보안 기능’에 DDoS 보호 및 완화에 대한 섹션이 추가되었습니다. 웹 호스팅을 위한 서버리스 아키텍처에 대한 짧은 섹션이 추가되었습니다.	2017년 7월 1일
백서 업데이트됨	명확성을 개선하기 위해 여러 섹션이 업데이트되었습니다. AWS 아이콘을 사용하도록 다이어그램이 업데이트되었습니다. Amazon Route 53를 자세히 소개하는 ’퍼블릭 DNS 관리’ 섹션이 추가되었습니다. 명확성을 개선하기 위해 ‘다른 호스	2012년 9월 1일

	트 및 서비스 검색' 섹션이 업데이트되었습니다. 명확성을 개선하고 DynamoDB를 추가하기 위해 '데이터베이스 구성, 백업 및 장애 조치' 섹션이 업데이트되었습니다. '데이터 및 자산에 대한 스토리지 및 백업' 섹션이 EBS 프로비저닝된 IOPS 볼륨을 포함하도록 확장되었습니다.	
최초 게시	백서가 게시되었습니다.	2010년 5월 1일

고지 사항

이 문서는 정보 제공 목적으로만 제공됩니다. 본 문서의 발행일 당시 AWS의 현재 제품 및 실행 방법을 설명하며, 예고 없이 변경될 수 있습니다. 고객은 본 문서에 포함된 정보나 AWS 제품 또는 서비스의 사용을 독립적으로 평가할 책임이 있으며, 각 정보 및 제품은 명시적이든 묵시적이든 어떠한 종류의 보증 없이 "있는 그대로" 제공됩니다. 본 문서는 AWS, 그 계열사, 공급업체 또는 라이선스 제공자로부터 어떠한 보증, 표현, 계약 약속, 조건 또는 보증을 구성하지 않습니다. 고객에 대한 AWS의 책임 및 채무는 AWS 계약에 준거합니다. 본 문서는 AWS와 고객 간의 어떠한 계약도 구성하지 않으며 이를 변경하지도 않습니다.

© 2019 Amazon Web Services, Inc. or its affiliates. All rights reserved.