AWS 백서

ID 액세스 관리(IAM)

IAM 단순화:

IAM은 AWS 내에서 중앙 집중식 제어 허브를 제공하고 다른 모든 AWS 서비스와 통합됩니다. IAM은 다양한 수준의 권한에서 액세스를 공유하는 기능과 함께 제공되며 임시 또는 제한된 액세스를 위해 ID 연합(Facebook 또는 Google과 같은 신뢰할 수 있는 외부 당사자에게 인증을 위임하는 프로세스)을 사용하는 기능을 지원합니다. IAM은 MFA 지원과 함께 제공되며 전체 조직에서 사용자 지정 암호 교체 정책을 설정할 수 있습니다. 또한 PCI DSS와 호환됩니다. 즉, 지불 카드 산업 데이터 보안 표준입니다. (정부에서 지정한 신용 카드 보안 규정 통과).

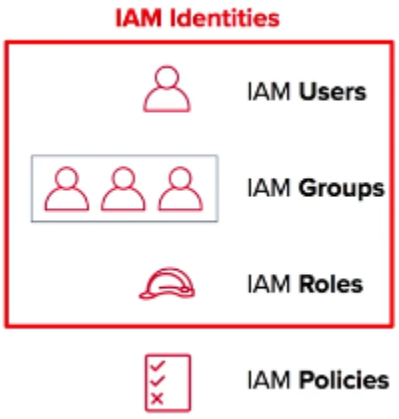
IAM 엔터티:

사용자 - 직원, 시스템 설계자, CTO 등과 같은 모든 개별 최종 사용자

그룹 - 시스템 관리자, HR 직원, 재무 팀 등과 같은 공유 권한을 가진 유사한 사람들의 모음입니다. 지정된 그룹 내의 각 사용자는 그룹에 대해 설정된 권한을 상속합니다.

역할 - S3에 대한 쓰기 권한이 필요한 AWS Lambda 또는 RDS MySQL 데이터베이스의 읽기 권한이 필요한 EC2 인스턴스 집합과 같이 작업을 수행하기 위해 권한을 부여받아야 하는 모든 소프트웨어 서비스.

정책 - 액세스 권한을 부여하거나 제한하기 위해 적용되는 문서화된 규칙 집합입니다. 사용자, 그룹 또는 역할이 권한을 적절하게 설정하기 위해 정책을 사용합니다. 정책은 JSON으로 작성되며 특정 요구 사항에 대한 사용자 지정 정책을 사용하거나 AWS에서 설정한 기본 정책을 사용할 수 있습니다.



IAM 정책은 IAM 자격 증명이 아니기 때문에 위의 다른 엔터티와 분리됩니다. 대신 해당 IAM 자격 증명이 필요한 기능을 수행할 수 있도록 IAM 자격 증명에 연결됩니다.

IAM 키 세부 정보:

IAM은 지역 제한이 없는 글로벌 AWS 서비스입니다. 모든 사용자, 그룹, 역할 또는 정책은 전역적으로 액세스할 수 있습니다.

완전한 관리자 액세스 권한이 있는 루트 계정은 AWS에 가입하는 데 사용되는 계정입니다. 따라서 사용할 AWS 계정을 생성하는 데 사용된 이메일 주소는 아마도 공식 회사 이메일 주소여야 합니다.

새 사용자는 계정이 처음 생성될 때 권한이 없습니다. 이는 의도적으로 권한을 부여해야 하므로 액세스를 안전하게 위임하는 방법입니다.

AWS 에코시스템에 처음 가입할 때 프로그래밍 방식의 액세스 권한을 부여하면 새 사용자에게 액세스 키 ID와 보안 액세스 키 ID가 제공됩니다. 이는 새 사용자가 참여할 수 있도록 특별히 한 번만 생성되므로 분실한 경우 새 액세스 키 ID와 새 보안 액세스 키 ID를 생성하기만 하면 됩니다. 액세스 키는 AWS CLI 및 SDK에만 사용되므로 콘솔에 액세스하는 데 사용할 수 없습니다.

AWS 계정을 생성할 때 회사 내부에 싱글 사인온(SSO)을 제공하는 기존 자격 증명 공급자가 있을 수 있습니다. 이 경우 AWS에서 기존 자격 증명을 재사용하는 것이 유용하고 효율적이며 완전히 가능합니다. 이렇게 하려면 Active Directory 중 하나가 IAM 역할을 맡도록 합니다. IAM ID 연동 기능을 통해 외부 서비스가 IAM 역할을 맡을 수 있기 때문입니다.

IAM 역할은 처음 사용/생성하기 전이나 사용/생성된 후에 EC2 인스턴스와 같은 서비스에 할당할 수 있습니다. 권한은 필요한 만큼 변경할 수 있습니다. 이 모든 작업은 AWS 콘솔과 AWS 명령줄 도구를 모두 사용하여 수행할 수 있습니다.

IAM 그룹을 중첩할 수 없습니다. 개별 IAM 사용자는 여러 그룹에 속할 수 있지만 하나의 IAM 그룹이 다른 IAM 그룹 내부에 포함되도록 하위 그룹을 생성하는 것은 불가능합니다.

IAM 정책을 사용하면 누가 액세스할 수 있는 리소스를 정의하는 데 도움이 되는 태그를 쉽게 추가할 수 있습니다. 그런 다음 이러한 태그는 특정 IAM 정책을 통해 액세스를 제어하는 데 사용됩

니다. 예를 들어 프로덕션 및 개발 EC2 인스턴스에 태그가 지정될 수 있습니다. 이렇게 하면 개발 인스턴스에만 액세스할 수 있어야 하는 사람들이 프로덕션 인스턴스에 액세스할 수 없습니다.

IAM의 우선 순위 수준:

명시적 거부 : 특정 리소스에 대한 액세스를 거부하며 이 판결을 번복할 수 없습니다.

명시적 허용 : 연결된 명시적 거부가 없는 한 특정 리소스에 대한 액세스를 허용합니다.

기본 거부(또는 암시적 거부) : IAM 자격 증명은 리소스 액세스 없이 시작됩니다. 대신 액세스 권한이 부여되어야 합니다.

Simple Storage Service(S3)

S3 단순화:

S3는 개발자와 IT 팀에 안전하고 내구성이 높으며 확장성이 뛰어난 개체 스토리지를 제공합니다. 블록 스토리지와 반대되는 오브젝트 스토리지는 다음 세 가지로 구성된 데이터를 나타내는 일반적인 용어입니다.

1.) 저장하려는 데이터

2.) 확장 가능한 메타데이터 양

3.) 데이터를 검색할 수 있는 고유 식별자

이것은 파일이나 디렉토리를 호스팅하기에 완벽한 후보이고 데이터베이스나 운영 체제를 호스팅하기에는 부적합한 후보가 됩니다. 다음 표는 객체와 블록 스토리지 간의 주요 차이점을 강조합니다.

	OBJECT STORAGE	BLOCK STORAGE
PERFORMANCE	Performs best for big content and high stream throughput	Strong performance with database and transactional data
GEOGRAPHY	Data can be stored across multiple regions	The greater the distance between storage and application, the higher the latency
SCALABILITY	Can scale infinitely to petabytes and beyond	Addressing requirements limit scalability
ANALYTICS	Customizable metadata allows data to be easily organized and retrieved	No metadata

S3에 업로드된 데이터는 여러 파일과 시설에 분산되어 있습니다. S3에 업로드되는 파일의 상한선은 파일당 5TB이며 업로드할 수 있는 파일의 수는 거의 무제한입니다. 모든 파일을 포함하는 S3 버킷은 유니버셜 네임스페이스로 이름이 지정되므로 고유성이 필요합니다. 모든 성공적인 업로드는 HTTP 200 응답을 반환합니다.

S3 주요 세부 정보:

객체(일반 파일 또는 디렉터리)는 키, 값, 버전 ID 및 메타데이터와 함께 S3에 저장됩니다. 또한 기본적으로 개체 자체에 대한 권한인 액세스 제어 목록에 대한 급류 및 하위 리소스를 포함할 수 있습니다.

S3용 데이터 일관성 모델은 초기 PUT 요청 후 새 객체에 대한 즉각적인 읽기 액세스를 보장합니다. 이러한 새 객체는 AWS에 처음 도입 되었으므로 어디에서나 업데이트할 필요가 없으므로 즉시 사용할 수 있습니다.

S3용 데이터 일관성 모델은 2020년 12월부터 이미 존재하는 객체의 PUTS 및 DELETES에 대한 즉각적인 읽기 액세스도 보장합니다 .

Amazon은 Reduced Redundancy Storage 클래스를 제외한 모든 S3 스토리지 클래스에 대해 99.999999999%(또는 11 9s)의 내구성을 보장합니다.

S3에는 다음과 같은 주요 기능이 있습니다.

1.) 계층형 스토리지 및 가격 변동성

2.) 오래된 콘텐츠를 만료시키는 수명 주기 관리

3.) 버전 관리를 위한 버전 관리

4.) 개인 정보 보호를 위한 암호화

5.) MFA는 콘텐츠의 우발적 또는 악의적 제거를 방지하기 위해 삭제합니다.

6.) 데이터 보호를 위한 액세스 제어 목록 및 버킷 정책

S3 요금:

1.) 저장 크기

2.) 요청 수

3.) 스토리지 관리 가격(계층이라고 함)

4.) 데이터 전송 요금(인터넷을 통해 AWS에 들어오거나 나가는 객체)

5.) 전송 가속(Cloudfront를 통해 움직이는 개체에 대한 선택적 속도 증가)

6.) 교차 지역 복제(기본적으로 제공되는 것보다 더 많은 HA)

버킷 정책은 버킷 수준에서 데이터를 보호하는 반면 액세스 제어는 보다 세분화된 객체 수준에서 보안 데이터를 나열합니다.

기본적으로 새로 생성된 모든 버킷은 비공개입니다.

S3는 액세스 로그를 생성하도록 구성할 수 있으며, 이 로그는 현재 계정의 다른 버킷이나 별도의 계정으로 모두 함께 배송될 수 있습니다. 이를 통해 누가 S3 내부에서 무엇에 액세스하는지 쉽게 모니터링할 수 있습니다.

AWS 계정 간에 S3 버킷을 공유하는 3가지 방법이 있습니다.

1.) 프로그래밍 방식 액세스의 경우에만 IAM 및 버킷 정책을 사용하여 전체 버킷 공유

2.) 프로그래밍 방식 액세스의 경우에만 ACL 및 버킷 정책을 사용하여 객체 공유

3.) 콘솔 및 터미널을 통한 액세스의 경우 교차 계정 IAM 역할을 사용합니다.

S3는 정적 웹사이트 호스팅을 위한 훌륭한 후보입니다. S3용 정적 웹 사이트 호스팅을 활성화하면 index.html 파일과 error.html 파일이 모두 필요합니다. 정적 웹 사이트 호스팅은 인터넷을 통해 액세스할 수 있는 웹 사이트 끝점을 만듭니다.

새 파일을 업로드하고 버전 관리를 활성화하면 이전 버전의 속성을 상속하지 않습니다.

S3 스토리지 클래스:

S3 Standard - 99.99% 가용성 및 99.99999%의 내구성. 이 클래스의 데이터는 여러 시설의 여러 장치에 중복 저장되며 2개의 동시 데이터 센터 오류를 견딜 수 있도록 설계되었습니다.

S3 Infrequently Accessed (IA) - 덜 자주 필요하지만 필요할 때 데이터를 신속하게 사용할 수 있는 데이터의 경우. 보관 수수료는 저렴하지만 검색 비용이 청구됩니다.

S3 One Zone Infrequently Accessed(레거시 RRS 개선 / 중복 스토리지 감소) - IA 비용을 낮추고 싶지만 고가용성은 필요하지 않은 경우에 적합합니다. 이것은 HA가 없기 때문에 더욱 저렴합니다.

S3 Intelligent Tiering(지능형 계층화) - 기본 제공 ML/AI를 사용하여 가장 비용 효율적인 스토리지 클래스를 결정한 다음 자동으로 데이터를 적절한 계층으로 이동합니다. 운영 오버헤드나 성능 영향 없이 이 작업을 수행합니다.

S3 Glacier - 데이터 보관을 위한 저렴한 스토리지 클래스입니다. 이 클래스는 검색이 자주 필요하지 않은 순수한 저장 목적을 위한 것입니다. 검색 시간은 몇 분에서 몇 시간까지 다양합니다. 기본 검색 시간이 허용되는 정도에 따라 검색 방법이 다릅니다. 자주 액세스 하지 않는 기밀파일

Expedited: 1 - 5 minutes, but this option is the most expensive.

Standard: 3 - 5 hours to restore.

Bulk: 5 - 12 hours. This option has the lowest cost and is good for a large set of data.

위에 나열된 긴급 기간은 AWS 전체에서 수요가 비정상적으로 높은 드문 상황에서 더 길어질 수 있습니다. 모든 상황에서 Glacier 데이터에 빠르게 액세스하는 것이 절대적으로 중요한 경우 프로비저닝된 용량을 구입해야 합니다. 프로비저닝된 용량은 긴급 검색이 항상 1~5분의 시간 제약 내에서 작동함을 보장합니다.

S3 Deep Glacier - 검색에 12시간이 소요될 수 있는 가장 저렴한 S3 스토리지입니다.

Storage Class	Designed for	Durability (designed for)	Availability (designed for)	Availability Zones	Min storage duration	Min billable object size	Other Considerations
STANDARD	Frequently accessed data	99.999999999%	99.99%	>= 3	None	None	None
STANDARD_IA	Long-lived, infrequently accessed data	99.999999999%	99.9%	>= 3	30 days	128 KB	Per GB retrieval fees apply.
INTELLIGENT_TIERING	Long-lived data with changing or unknown access patterns	99.999999999%	99.9%	>= 3	30 days	None	Monitoring and automation fees per object apply. No retrieval fees.
ONEZONE_IA	Long-lived, infrequently accessed, non-critical data	99.999999999%	99.5%	1	30 days	128 KB	Per GB retrieval fees apply. Not resilient to the loss of the Availability Zone.
GLACIER	Long-term data archiving with retrieval times ranging from minutes to hours	99.999999999%	99.99% (after you restore objects)	>= 3	90 days	40 KB	Per GB retrieval fees apply. You must first restore archived objects before you can access them. For more information, see Restoring Archived Objects.
DEEP_ARCHIVE	Archiving rarely accessed data with a default retrieval time of 12 hours	99.999999999%	99.99% (after you restore objects)	>= 3	180 days	40 KB	Per GB retrieval fees apply. You must first restore archived objects before you can access them. For more information, see Restoring Archived Objects.
RRS (Not recommended)	Frequently accessed, non-critical data	99.99%	99.99%	>= 3	None	None	None

S3 암호화:

S3 데이터는 전송 및 저장 모두에서 암호화될 수 있습니다.

전송 중 암호화 : 한 끝점에서 다른 끝점으로 전달되는 트래픽을 해독할 수 없는 경우입니다. 서버 A와 서버 B 사이를 도청하는 사람은 지나가는 정보를 이해할 수 없습니다. S3의 전송 중 암호화는 항상 SSL/TLS를 통해 이루어집니다.

Encryption At Rest : S3 내부에 있는 고정 데이터를 암호화할 때. 누군가가 서버에 침입하더라도 여전히 해당 서버 내의 암호화된 정보에 액세스할 수 없습니다. 유휴 시 암호화는 서버 측 또는 클라이언트 측에서 수행할 수 있습니다. 서버 측에서는 S3가 데이터를 디스크에 쓸 때 암호화하고 액세스할 때 암호를 해독합니다. 클라이언트 측에서는 자체적으로 개체를 개인적으로 암호화한 다음 나중에 S3에 업로드할 때입니다.

다음과 같은 방법으로 AWS 지원 서버 측에서 암호화할 수 있습니다.

- S3 관리형 키 / SSE - S3(서버 측 암호화 S3) - Amazon이 자동으로 암호화 및 복호화 키를 관리하는 경우. 이 시나리오에서는 사용 편의성을 대가로 Amazon에 약간의 제어 권한을 부여합니다.
- AWS Key Management Service / SSE - KMS - Amazon과 귀하가 암호화 및 복호화 키를 함께 관리하는 경우. 사용된 때만 사용권한을 요청하는 감사목적기능제공
- 서버 측 암호화 고객 제공 키 / SSE - C - 내가 관리하는 자체 키를 Amazon에 제공할 때. 이 시나리오에서는 더 많은 제어에 대한 대가로 사용 용이성을 인정합니다.

S3 버전 관리:

버전 관리가 활성화되면 S3는 모든 쓰기 및 삭제를 포함하여 객체의 모든 버전을 저장합니다.

콘텐츠를 암시적으로 백업하고 사람의 실수로 인해 쉽게 롤백할 수 있는 훌륭한 기능입니다.

Git과 유사하다고 생각할 수 있습니다.

버킷에서 버전 관리가 활성화되면 비활성화할 수 없으며 일시 중단만 됩니다.

버전 관리는 수명 주기 규칙을 통합하므로 해당 버전에 따라 데이터를 만료하거나 마이그레이션하는 규칙을 설정할 수 있습니다.

버전 관리에는 추가 보안 계층을 제공하는 MFA 삭제 기능도 있습니다.

S3 수명 주기 관리:

서로 다른 스토리지 계층 간의 개체 이동을 자동화합니다.

버전 관리와 함께 사용할 수 있습니다.

수명 주기 규칙은 객체의 현재 버전과 이전 버전 모두에 적용할 수 있습니다.

S3 교차 리전 복제:

교차 지역 복제는 버전 관리가 활성화된 경우에만 작동합니다.

교차 지역 복제가 활성화되면 기존 데이터가 전송되지 않습니다. 원래 버킷으로의 새 업로드만 복제됩니다. 모든 후속 업데이트가 복제됩니다.

한 버킷의 콘텐츠를 다른 버킷으로 복제할 때 원하는 경우 실제로 콘텐츠의 소유권을 변경할 수 있습니다. 복제된 콘텐츠로 새 버킷의 스토리지 계층을 변경할 수도 있습니다.

파일이 원본 버킷에서 삭제되면(버전 관리로 인해 실제 삭제가 방지되므로 삭제 마커를 통해) 해당 삭제가 복제되지 않습니다.

교차 지역 복제 개요

암호화된 개체, 삭제, Glacier의 항목 등과 같이 복제되는 것과 복제되지 않는 것

S3 전송 가속:

Transfer Acceleration은 오리진에서 업로드 또는 다운로드 속도가 느려지는 대신 CDN 접속 지점(에지 로케이션이라고 함)에서 데이터를 보내거나 수신하여 CloudFront 네트워크를 사용합니다.

이는 버킷 자체가 아닌 엣지 로케이션에 대한 고유한 URL에 업로드하여 수행됩니다. 그런 다음 훨씬 빠른 속도로 AWS 네트워크 백본을 통해 전송됩니다.

일반 업로드와 비교하여 직접 전송 가속 속도를 테스트할 수 있습니다.

S3 이벤트 알림:

Amazon S3 알림 기능을 사용하면 버킷에서 특정 이벤트가 발생할 때 알림을 받고 보낼 수 있습니다. 알림을 활성화하려면 먼저 Amazon S3에서 게시할 이벤트(새 객체 추가, 이전 객체 삭제 등)와 Amazon S3에서 이벤트 알림을 보낼 대상을 구성해야 합니다. Amazon S3는 이벤트를 게시할 수 있는 다음 대상을 지원합니다.

Amazon Simple Notification Service(Amazon SNS) – 구독 엔드포인트 또는 클라이언트에 대한 메시지 전달 또는 전송을 조정하고 관리하는 웹 서비스입니다.

Amazon Simple Queue Service(Amazon SQS) – SQS는 컴퓨터 간에 이동할 때 메시지를 저장할 수 있는 안정적이고 확장 가능한 호스팅 대기열을 제공합니다.

AWS Lambda – AWS Lambda는 사용자가 코드를 업로드할 수 있고 서비스가 AWS 인프라를 사용하여 사용자를 대신하여 코드를 실행할 수 있는 컴퓨팅 서비스입니다. Lambda 함수를 생성할 때 사용자 지정 코드를 패키징하고 AWS Lambda에 업로드합니다. Lambda 함수를 트리거하는 S3 이벤트는 코드의 입력으로도 사용할 수 있습니다.

S3 및 ElasticSearch:

S3를 사용하여 로그 파일을 저장하는 경우 ElasticSearch는 로그에 대한 전체 검색 기능을 제공하며 S3 버킷에 저장된 데이터를 검색하는 데 사용할 수 있습니다.

ElasticSearch 도메인을 S3 및 Lambda와 통합할 수 있습니다. 이 설정에서 S3가 수신한 모든 새 로그는 Lambda에 대한 이벤트 알림을 트리거하고, 그러면 Lambda가 새 로그 데이터에서 애플리케이션 코드를 실행합니다. 코드가 처리를 마치면 데이터가 ElasticSearch 도메인으로 스트리밍되고 관찰에 사용할 수 있습니다.

S3 읽기/쓰기 성능 최대화:

S3에 대한 객체 읽기 및 쓰기 요청 비율이 매우 높은 경우 접두사에 대한 순차적 날짜 기반 이름 지정을 사용하여 성능을 향상시킬 수 있습니다. 이전 버전의 AWS Docs에서는 해시 키 또는 임의의 문자열을 사용하여 객체 이름 접두사를 사용할 것을 제안했습니다. 이러한 경우 개체를 저장하는 데 사용되는 파티션이 더 잘 분산되어 개체에 대한 읽기/쓰기 성능이 향상됩니다.

S3 데이터가 사용자로부터 많은 수의 GET 요청을 수신하는 경우 성능 최적화를 위해 Amazon CloudFront 사용을 고려해야 합니다. CloudFront를 S3와 통합하면 CloudFront의 캐시를 통해 사용자에게 콘텐츠를 배포하여 지연 시간을 줄이고 데이터 전송 속도를 높일 수 있습니다. 여기에는 S3에 더 적은 직접 요청을 보내는 추가 보너스가 있어 비용이 절감됩니다. 예를 들어 매우 인기 있는 몇 가지 개체가 있다고 가정합니다. CloudFront는 S3에서 해당 객체를 가져와 캐시합니다. 그런 다음 CloudFront는 캐시에서 객체에 대한 향후 요청을 처리하여 Amazon S3로 보내는 총 GET 요청 수를 줄일 수 있습니다.

S3에서 고성능을 보장하는 방법에 대한 추가 정보

S3 서버 액세스 로깅:

서버 액세스 로깅은 버킷에 대한 요청에 대한 자세한 기록을 제공합니다. 서버 액세스 로그는 많은 애플리케이션에 유용합니다. 예를 들어, 액세스 로그 정보는 보안 및 액세스 감사에 유용할 수 있습니다. 또한 고객 기반에 대해 배우고 Amazon S3 청구서를 더 잘 이해하는 데 도움이 될 수 있습니다.

기본적으로 로깅은 비활성화되어 있습니다. 로깅이 활성화되면 로그는 원본 버킷과 동일한 AWS 리전의 버킷에 저장됩니다.

각 액세스 로그 레코드는 요청자, 버킷 이름, 요청 시간, 요청 작업, 응답 상태 및 오류 코드(해당되는 경우)와 같은 단일 액세스 요청에 대한 세부 정보를 제공합니다.

다음과 같은 방식으로 작동합니다.

- 모니터링하려는 버킷의 액세스 로그 기록을 주기적으로 수집하는 S3
- 그런 다음 S3는 해당 레코드를 로그 파일로 통합합니다.
- S3는 마침내 로그 파일을 로그 객체로 보조 모니터링 버킷에 업로드합니다.

S3 멀티파트 업로드:

멀티파트 업로드를 사용하면 단일 개체를 부분 집합으로 업로드할 수 있습니다. 각 부분은 개체 데이터의 연속 부분입니다. 이러한 개체 부분을 독립적으로 어떤 순서로든 업로드할 수 있습니다.

멀티파트 업로드는 100MB를 초과하는 파일에 권장되며 5GB를 초과하는 파일을 업로드 하는 유일한 방법 입니다. 효율성을 높이기 위해 데이터를 병렬로 업로드하여 기능을 달성합니다.

어떤 부분의 전송이 실패하면 다른 부분에 영향을 주지 않고 해당 부분을 재전송할 수 있습니다. 객체의 모든 부분이 업로드되면 Amazon S3는 이러한 부분을 조합하고 객체를 생성합니다.

멀티파트 업로드를 사용하려는 가능한 이유:

- 멀티파트 업로드는 최종 개체 크기를 알기 전에 업로드를 시작할 수 있는 기능을 제공합니다.
- 멀티파트 업로드는 향상된 처리량을 제공합니다.
- 멀티파트 업로드는 개체 업로드를 일시 중지하고 다시 시작하는 기능을 제공합니다.
- 멀티파트 업로드는 네트워크 문제에서 빠른 복구를 제공합니다.

AWS SDK를 사용하여 객체를 부분적으로 업로드할 수 있습니다. 또는 AWS CLI를 통해 동일한 작업을 수행할 수 있습니다.

바이트 범위 가져오기 를 사용하여 S3에서 다운로드를 병렬화할 수도 있습니다 . 다운로드하는 동안 오류가 발생하면 전체 개체가 아닌 특정 바이트 범위에만 오류가 지역화됩니다.

S3 사전 서명된 URL:

모든 S3 객체는 기본적으로 프라이빗이지만 프라이빗 객체가 있는 프라이빗 버킷의 객체 소유자는 버킷의 권한을 퍼블릭으로 변경하지 않고도 해당 객체를 선택적으로 공유할 수 있습니다.

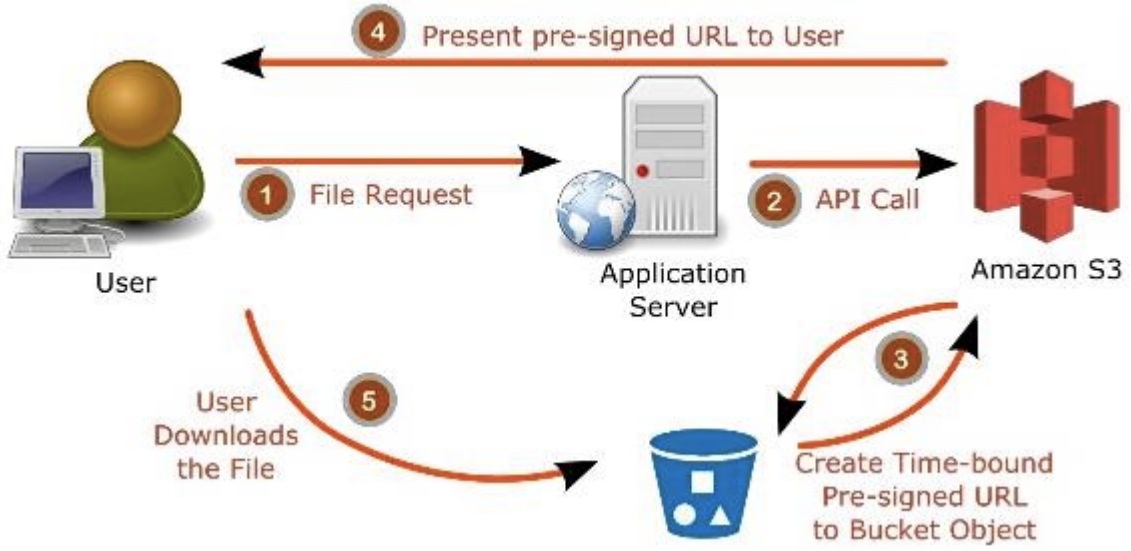
이것은 미리 서명된 URL을 생성하여 수행됩니다. 자체 보안 자격 증명을 사용하여 비공개 S3 객체를 다운로드하거나 볼 수 있는 제한된 권한을 부여할 수 있습니다.

S3 객체에 대해 미리 서명된 URL을 생성할 때 다음을 수행해야 합니다.

- 보안 자격 증명을 제공합니다.
- 버킷을 지정합니다.
- 개체 키를 지정합니다.
- HTTP 메서드를 지정합니다(객체를 다운로드하려면 GET).
- 만료 날짜와 시간을 지정합니다.

미리 서명된 URL은 지정된 기간 동안만 유효하며 해당 기간 내에 미리 서명된 URL을 받는 사람은 누구나 객체에 액세스할 수 있습니다.

다음 다이어그램은 사전 서명된 URL의 작동 방식을 강조합니다.



S3 선택:

S3 Select는 객체에서 필요한 데이터만 가져오도록 설계된 Amazon S3 기능으로, 성능을 크게 향상시키고 S3의 데이터에 액세스해야 하는 애플리케이션의 비용을 절감할 수 있습니다.

대부분의 응용 프로그램은 전체 개체를 검색한 다음 추가 분석을 위해 필요한 데이터만 필터링해야 합니다. S3 Select를 사용하면 애플리케이션에서 Amazon S3 서비스로 객체 내부의 데이터에 액세스하고 필터링하는 작업을 오프로드할 수 있습니다.

예를 들어 대형 소매업체의 개발자가 단일 매장의 주간 판매 데이터를 분석해야 하지만 200개 매장 모두에 대한 데이터가 매일 새로운 GZIP 형식 CSV로 저장된다고 가정해 보겠습니다.

- S3 Select가 없으면 필요한 데이터를 얻으려면 전체 CSV를 다운로드, 압축 해제 및 처리해야 합니다.
- S3 Select를 사용하면 전체 객체를 검색하는 대신 간단한 SQL 표현식을 사용하여 관심 있는 저장소의 데이터만 반환할 수 있습니다.

애플리케이션에서 로드하고 처리해야 하는 데이터의 양을 줄임으로써 S3 Select는 훨씬 적은 양의 데이터를 처리하기 때문에 S3의 데이터에 자주 액세스하는 대부분의 애플리케이션의 성능을 최대 400%까지 향상시킬 수 있습니다.

Glacier용 S3 Select를 사용할 수도 있습니다.

CloudFront

간소화된 CloudFront:

AWS CDN 서비스를 CloudFront라고 합니다. 애플리케이션의 향상된 글로벌 성능을 위해 캐시된 콘텐츠와 자산을 제공합니다. CloudFront의 주요 구성 요소는 엣지 로케이션(캐시 엔드포인트), 오리진(EC2 인스턴스, S3 버킷, Elastic Load Balancer 또는 Route 53 구성과 같이 캐싱할 원본 소스) 및 배포(원점 또는 기본적으로 네트워크 자체에서 에지 위치의 배열). CloudFront의 기능에 대한 추가 정보

CloudFront 키 세부 정보:

콘텐츠가 캐시되면 TTL(Time To Live)이라고 하는 특정 시간 제한(항상 초 단위) 동안 수행됩니다.

필요한 경우 CloudFront는 동적, 정적, 스트리밍 및 대화형 콘텐츠를 포함한 전체 웹 사이트를 제공할 수 있습니다.

요청은 항상 사용자에게 가장 가까운 에지 위치에서 라우팅 및 캐시되므로 CDN 노드를 전파하고 향후 요청에 대해 최상의 성능을 보장합니다.

두 가지 다른 유형의 배포가 있습니다.

- 웹 배포 : 웹사이트, 일반 캐시 아이템 등
- RTMP : 스트리밍 콘텐츠, 어도비 등

Edge locations 읽기 전용이 아닙니다. 그런 다음 쓰기 값을 원점으로 되돌려 보낼 수 있습니다.

캐시된 콘텐츠는 TTL을 초과하여 수동으로 무효화하거나 지울 수 있지만 비용이 발생합니다.

콘텐츠가 매번 원본에서 직접 로드되도록 특정 개체 또는 전체 디렉터리의 배포를 무효화할 수 있습니다. 콘텐츠를 무효화하는 것은 오리진에서 가져온 콘텐츠가 올바른 것처럼 보이지만 에지 위치에서 동일한 콘텐츠를 가져오는 것이 잘못된 것처럼 보이는 경우 디버깅할 때도 유용합니다.

내부에 두 개의 오리진이 있는 오리진 그룹을 생성하여 오리진에 대한 장애 조치를 설정할 수 있습니다. 하나의 오리진은 기본 역할을 하고 다른 하나는 보조 역할을 합니다. CloudFront는 기본 오리진에 장애가 발생하면 자동으로 둘 사이를 전환합니다.

Amazon CloudFront는 각 엣지 로케이션에서 콘텐츠를 제공하고 전용 IP 사용자 지정 SSL 기능을 제공합니다. SNI Custom SSL은 대부분의 최신 브라우저에서 작동합니다.

PCI 또는 HIPAA 호환 워크로드를 실행하고 사용 데이터를 기록해야 하는 경우 다음을 수행할 수 있습니다.

- CloudFront 액세스 로그를 활성화합니다.
- CloudFront API로 전송되는 요청을 캡처합니다.

OAI(Origin Access Identity)는 CloudFront를 통해 비공개 콘텐츠를 공유하는 데 사용됩니다. OAI는 오리진(예: S3 버킷)에서 프라이빗 객체를 가져올 수 있는 권한을 CloudFront 배포에 부여하는 데 사용되는 가상 사용자입니다.

CloudFront 서명된 URL 및 서명된 쿠키:

CloudFront 서명된 URL과 서명된 쿠키는 동일한 기본 기능을 제공합니다. 이를 통해 콘텐츠에 액세스할 수 있는 사람을 제어할 수 있습니다. 이러한 기능은 인터넷을 통해 콘텐츠를 배포하는 많은 회사에서 문서, 비즈니스 데이터, 미디어 스트림 또는 선택한 사용자를 대상으로 하는 콘텐츠에 대한 액세스를 제한하기를 원하기 때문에 존재합니다. 예를 들어 요금을 지불한 사용자는 프리 티어 사용자가 액세스할 수 없는 비공개 콘텐츠에 액세스할 수 있어야 합니다.

CloudFront를 통해 프라이빗 콘텐츠를 제공하고 서명된 URL을 사용할지 또는 서명된 쿠키를 사용할지 결정하려는 경우 다음을 고려하십시오.

다음 경우에 서명된 URL을 사용하십시오.

- RTMP 배포를 사용하려고 합니다. RTMP 배포에는 서명된 쿠키가 지원되지 않습니다.
- 개별 파일(예: 응용 프로그램 설치 다운로드)에 대한 액세스를 제한하려고 합니다.
- 사용자가 쿠키를 지원하지 않는 클라이언트(예: 사용자 지정 HTTP 클라이언트)를 사용하고 있습니다.

다음 경우에 서명된 쿠키를 사용하십시오.

- 여러 제한된 파일에 대한 액세스를 제공하려고 합니다. 예를 들어, HLS 형식의 비디오에 대한 모든 파일 또는 웹사이트의 유료 사용자 영역에 있는 모든 파일.
- 현재 URL을 변경하고 싶지 않습니다.

Snowball

Snowball 단순화:

Snowball은 대량의 데이터를 AWS로 마이그레이션하는 데 사용되는 거대한 물리적 디스크입니다. 페타바이트 규모의 데이터 전송 솔루션입니다. Snowball과 같은 대용량 디스크를 사용하면 높은 네트워크 비용, 긴 전송 시간, 보안 문제와 같은 일반적인 대규모 데이터 전송 문제를 피할 수 있습니다. Snowball은 설계상 매우 안전하며 데이터 전송이 완료되면 Snowball은 데이터에서 지워집니다.

50TB (42TB 사용가능) - 이전
80TB (72TB 사용가능)

Snowball → Glacier
↳ S3 → Glacier

Snowball 키 세부 정보:

Snowball은 AWS로 테라바이트에서 수 페타바이트에 이르는 안전하고 빠른 데이터 전송이 필요한 경우 데이터 전송 작업을 위한 강력한 선택입니다.

Snowball은 또한 기존 네트워크 인프라에 대한 값비싼 업그레이드를 원하지 않는 경우, 대량의 데이터 백로그가 자주 발생하는 경우, 물리적으로 격리된 환경에 있는 경우 또는 고속 인터넷 연결이 불가능하거나 비용이 많이 드는 지역.

일반적으로 기존 인터넷 연결의 여유 용량을 사용하여 데이터를 AWS에 업로드하는 데 1주일 이상 걸린다면 Snowball 사용을 고려해야 합니다.

예를 들어 데이터 전송 전용으로 사용할 수 있는 100Mb 연결이 있고 총 100TB의 데이터를 전송해야 하는 경우 해당 연결을 통해 전송이 완료되는 데 100일 이상이 걸립니다. Snowball을 여러 개 사용하면 약 일주일 만에 동일한 전송을 수행할 수 있습니다.

다음은 인터넷 연결을 통해 동일한 전송을 수행하는 데 걸리는 일 수를 기준으로 Snowball을 고려해야 하는 경우에 대한 참조입니다.

Available Internet Connection	Theoretical Min. Number of Days to Transfer 100TB at 80% Network Utilization	When to Consider AWS Import/Export Snowball?
T3 (44.736Mbps)	269 days	2TB or more
100Mbps	120 days	5TB or more
1000Mbps	12 days	60TB or more

Snowball Edge 및 Snowmobile:

인터넷보다 빠른 속도로 데이터

Snowball Edge는 AWS Lambda 및 특정 EC2 인스턴스 유형을 통한 컴퓨팅 및 스토리지 기능과 함께 제공되는 특정 유형의 Snowball입니다. 즉, 데이터가 Amazon 데이터 센터로 전송되는 동안 Snowball 내에서 코드를 실행할 수 있습니다. 이를 통해 원격 또는 오프라인 위치에서 로컬 워크로드를 지원할 수 있으므로 Snowball Edge는 데이터 전송 서비스로 제한될 필요가 없습니다. 흥미로운 사용 사례는 여객기입니다. 비행기는 때때로 Snowball Edge를 탑재한 상태로 비행하므로 많은 양의 비행 데이터를 저장하고 비행기 자체 시스템에 필요한 기능을 계산할 수 있습니다. Snowball Edge는 더 나은 성능을 위해 로컬로 클러스터링할 수도 있습니다.

Snowmobile은 엑사바이트 규모의 데이터 전송 솔루션입니다. 100페타바이트의 데이터를 위한 데이터 전송 솔루션이며 세미 트럭으로 운반되는 45피트 컨테이너 안에 들어 있습니다. 이러한 대규모 전송은 수년간의 데이터가 포함된 전체 데이터 센터를 클라우드로 이동하려는 경우에 적합합니다.

100TB (83TB 사용가능)

Storage Gateway

간소화된 Storage Gateway:

Storage Gateway는 온프레미스 애플리케이션을 클라우드 스토리지 백엔드와 원활하고 안전하게 통합하기 위해 온프레미스 환경과 클라우드 기반 스토리지를 연결하는 서비스입니다. 클라우드에 가상 하드 디스크 드라이브를 저장하는 방법으로 Volume Gateway.

Storage Gateway 주요 세부 정보:

Storage Gateway 서비스는 물리적 장치이거나 온프레미스 데이터 센터의 호스트에 다운로드된 VM 이미지일 수 있습니다. AWS에서 데이터를 보내거나 받는 브리지 역할을 합니다.

Storage Gateway는 Linux 머신용 VMWare의 ESXi 하이퍼바이저와 Windows 머신용 Microsoft Hyper-V 하이퍼바이저 위에 위치할 수 있습니다.

세 가지 유형의 Storage Gateway는 다음과 같습니다.

- 파일 게이트웨이 – NFS 또는 SMB를 통해 작동하며 제공된 가상 머신의 네트워크 파일 시스템 마운트 지점을 통해 S3에 파일을 저장하는 데 사용됩니다. 간단히 말해서 파일 게이트웨이는 S3의 파일 시스템 마운트로 생각할 수 있습니다.
- 볼륨 게이트웨이 – iSCSI를 통해 작동하며 S3에 하드 디스크 드라이브 또는 가상 하드 디스크 드라이브의 복사본을 저장하는 데 사용됩니다. 이는 Stored Volumes 또는 Cached Volumes 를 통해 달성할 수 있습니다. 간단히 말해서 볼륨 게이트웨이는 클라우드에 가상 하드 디스크 드라이브를 저장하는 방법으로 생각할 수 있습니다.
- 테이프 게이트웨이 – 가상 테이프 라이브러리로 작동

파일 소유권, 권한, 타임스탬프 등과 같이 Storage Gateway를 통과하는 관련 파일 정보는 그들이 속한 객체에 대한 메타데이터로 저장됩니다. 이러한 파일 세부 정보가 S3에 저장되면 기본적으로 관리할 수 있습니다. 즉, 버전 관리, 수명 주기 관리, 버킷 정책, 교차 지역 복제 등과 같은 모든 S3 기능을 Storage Gateway의 일부로 적용할 수 있습니다.

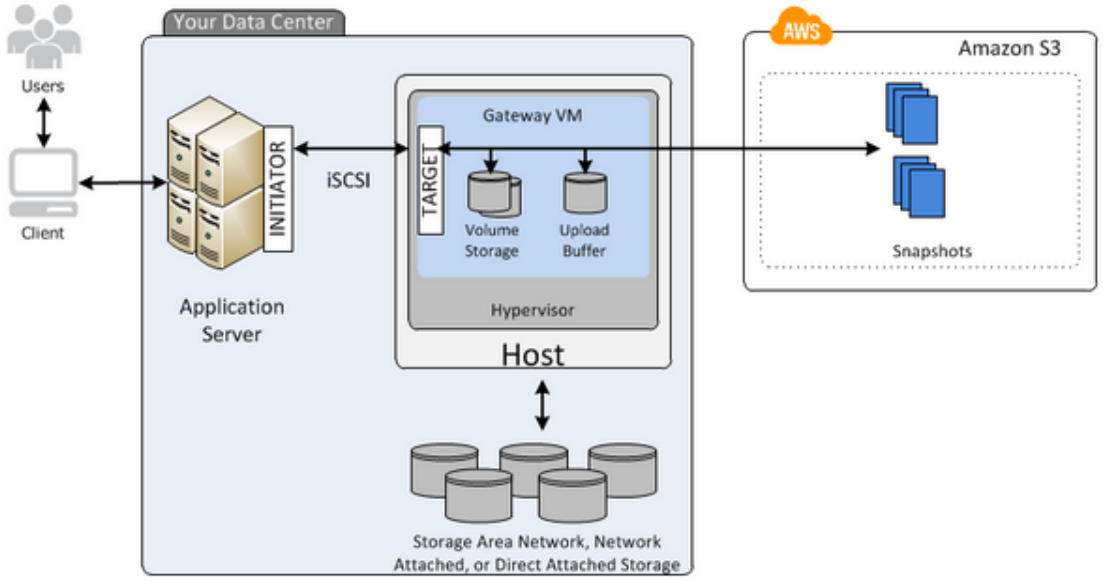
볼륨 게이트웨이를 통해 AWS와 인터페이스하는 애플리케이션은 iSCSI 블록 프로토콜을 통해 수행됩니다. 이러한 볼륨에 기록된 데이터는 볼륨 콘텐츠의 특정 시점 스냅샷으로 AWS Elastic Block Store(EBS)에 비동기식으로 백업할 수 있습니다. 이러한 종류의 스냅샷은 Git의 pull 요청과 유사한 변경된 상태만 캡처하는 증분 백업 역할을 합니다. 또한 모든 스냅샷은 저장 비용을 줄이기 위해 압축됩니다.

테이프 게이트웨이는 테이프(구식 데이터 스토리지)를 제거하면서 데이터를 S3에 보관 및 복제하는 내구성 있고 비용 효율적인 방법을 제공합니다. VTL(가상 테이프 라이브러리)은 기존 테이프 기반 백업 인프라를 활용하여 테이프 게이트웨이에서 생성한 가상 테이프 카트리지에 데이터를 저장합니다. 백업을 현대화하고 클라우드로 이동하는 좋은 방법입니다.

Stored Volumes VS Cached Volumes:

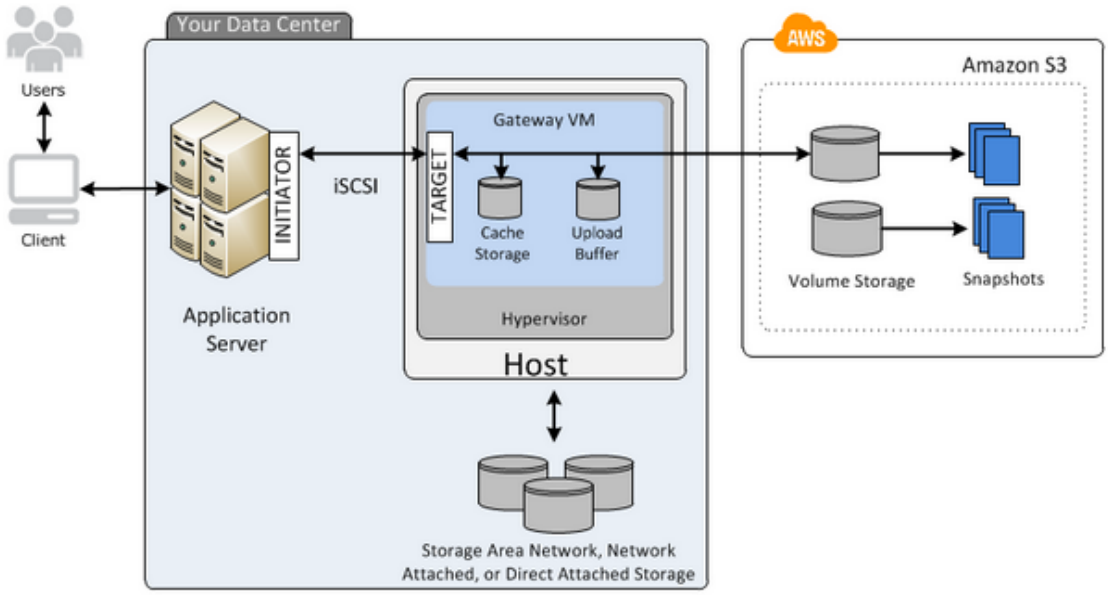
Volume Gateway의 Stored Volumes 를 사용하면 데이터를 온프레미스에 로컬로 저장하고 데이터를 보조 데이터 원본으로 AWS에 백업할 수 있습니다. Stored Volumes는 전체 데이터 세트에 대한 짧은 대기 시간 액세스를 허용하는 동시에 하이브리드 클라우드 솔루션에 대한 고가용성을 제공합니다. 또한 애플리케이션 인프라에 저장된 볼륨을 iSCSI 드라이브로 탑재할 수 있으므로 이러한 볼륨에 데이터가 기록될 때 데이터가 온프레미스 하드웨어에 기록되고 AWS EBS 또는 S3에서 스냅샷으로 비동기식으로 백업됩니다.

- Stored Volume 아키텍처의 다음 다이어그램에서 데이터는 데이터 센터 내의 Storage Area Network, Network Attached 또는 Direct Attached Storage에서 사용자에게 제공됩니다. S3는 안전하고 안정적인 백업으로 존재합니다.



Volume Gateway의 Cached Volumes 는 Stored Volumes처럼 전체 데이터 세트를 로컬에 저장하지 않는다는 점에서 다릅니다. 대신 AWS가 기본 데이터 소스로 사용되고 로컬 하드웨어가 캐싱 계층으로 사용됩니다. 가장 자주 사용되는 구성 요소만 온프레미스 인프라에 유지되고 나머지 데이터는 AWS에서 제공됩니다. 이렇게 하면 가장 많이 참조되는 데이터에 대한 낮은 대기 시간 액세스를 유지하면서 온프레미스 인프라를 확장해야 할 필요성이 최소화됩니다.

- 캐시 볼륨 아키텍처의 다음 다이어그램에서 가장 자주 액세스하는 데이터는 데이터 센터 내의 Storage Area Network, Network Attached 또는 DAS에서 사용자에게 제공됩니다. S3는 AWS의 나머지 데이터를 제공합니다.



Elastic Compute Cloud(EC2)

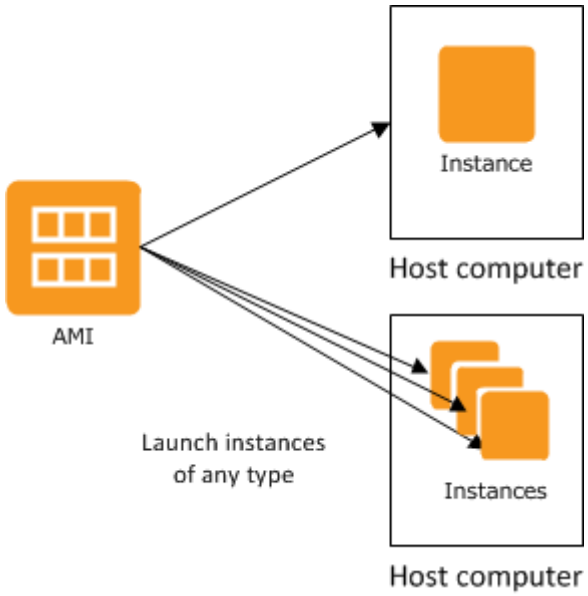
EC2 단순화:

EC2는 빠르게 확장 및 축소할 수 있는 크기 조정 가능한 서버 인스턴스를 가동합니다. 인스턴스는 클라우드의 가상 서버입니다. Amazon EC2를 사용하면 인스턴스에서 실행되는 운영 체제와 애플리케이션을 설정하고 구성할 수 있습니다. 시작 시 구성 은 인스턴스를 시작할 때 지정한 Amazon 머신 이미지(AMI) 의 라이브 복사본입니다. EC2는 새 인스턴스를 프로비저닝하고 부팅하는 데 소요되는 시간을 극도로 단축했으며 EC2는 사용한 만큼 지불하고 사용한 만큼 지불하고 사용한 만큼 적게 지불하고 용량을 예약할 때 더 적은 비용을 지불하도록 보장합니다. EC2 인스턴스가 실행 중일 때 CPU, 메모리, 스토리지 및 네트워킹에 대한 요금이 부과됩니다. 중지되면 EBS 스토리지에 대해서만 요금이 청구됩니다.

EC2 키 세부 정보:

단일 AMI에서 다양한 유형의 인스턴스를 시작할 수 있습니다. 인스턴스 유형은 기본적으로 인스턴스에 사용되는 호스트 컴퓨터의 하드웨어를 결정합니다. 각 인스턴스 유형은 서로 다른 컴퓨팅 및 메모리 기능을 제공합니다. 인스턴스 위에서 실행하려는 애플리케이션이나 소프트웨어에 필요한 메모리 및 컴퓨팅 성능에 따라 인스턴스 유형을 선택해야 합니다.

다음 그림과 같이 AMI의 여러 인스턴스를 시작할 수 있습니다.



인스턴스에 전용 테넌시를 사용할 수 있는 옵션이 있습니다. 즉, AWS 데이터 센터 내에서 물리적 하드웨어에 독점적으로 액세스할 수 있습니다. 당연히 이 옵션은 비용이 많이 들지만 엄격한 라이선스 정책이 있는 기술로 작업하는 경우에는 합리적입니다.

EC2 VM Import를 사용하면 해당 호스트가 VMware ESX, VMware Workstation, Microsoft Hyper-V 또는 Citrix Xen 가상화 형식을 사용하는 한 기존 VM을 AWS로 가져올 수 있습니다.

새 EC2 인스턴스를 시작하면 EC2는 모든 VM이 서로 다른 하드웨어에 분산되어 장애를 단일 위치로 제한하는 방식으로 인스턴스를 배치하려고 시도합니다. 배치 그룹을 사용하여 워크로드의 요구 사항을 충족하는 상호 종속 인스턴스 그룹의 배치에 영향을 줄 수 있습니다. 아래 섹션에 배치 그룹에 대한 설명이 있습니다.

Amazon EC2에서 인스턴스를 시작할 때 인스턴스가 시작될 때 사용자 데이터를 인스턴스에 전달할 수 있는 옵션이 있습니다. 이 사용자 데이터는 일반적인 자동화 구성 작업 또는 스크립트를 실행하는 데 사용할 수 있습니다. 예를 들어 htop이 새 EC2 호스트에 설치되고 항상 활성 상태인지 확인하는 bash 스크립트를 전달할 수 있습니다.

기본적으로 EC2 인스턴스의 퍼블릭 IP 주소는 인스턴스가 일시적으로 중지되더라도 인스턴스가 중지되면 해제됩니다. 따라서 외부 DNS 호스트 이름으로 인스턴스를 참조하는 것이 가장 좋습니다. 동일한 인스턴스에 연결할 수 있는 영구 공용 IP 주소가 필요한 경우 기본적으로 고정 IP 주소인 탄력적 IP 주소를 대신 사용하십시오.

SQL 데이터베이스를 자체 관리해야 하는 요구 사항이 있는 경우 EC2가 RDS의 확실한 대안이 될 수 있습니다. 고가용성을 보장하려면 별도의 가용 영역에 다른 EC2 인스턴스가 하나 이상 있어야 DB 인스턴스가 다운되더라도 다른 인스턴스는 계속 사용할 수 있습니다.

골든 이미지는 필요한 모든 소프트웨어/데이터/구성 세부 정보가 설정되어 있고 즉시 사용할 수 있도록 원하는 대로 완전히 사용자 정의한 AMI입니다. 그러면 이 개인 AMI가 새 인스턴스를 시작하는 소스가 될 수 있습니다.

인스턴스 상태 확인은 실행 중인 EC2 서버의 상태를 확인하고 시스템 상태 확인은 기본 하이퍼바이저의 상태를 모니터링합니다. 시스템 상태 문제가 발견되면 VM이 새 하이퍼바이저에서 다시 시작되므로 인스턴스를 중지하고 다시 시작하면 됩니다(재부팅할 필요 없음).

EC2 인스턴스 요금:

온디맨드 인스턴스 는 시간 또는 초 단위의 고정 요금을 기반으로 합니다. 이름에서 알 수 있듯이 필요할 때마다 온디맨드 인스턴스를 시작할 수 있고 더 이상 필요하지 않을 때 중지할 수 있습니다. 장기 약정에 대한 요구 사항은 없습니다.

예약 인스턴스를 사용하면 1년 또는 3년 계약 조건으로 인스턴스를 독점적으로 사용할 수 있습니다. 장기 약정은 시간당 요율로 대폭 할인된 할인을 제공합니다.

스팟 인스턴스 는 Amazon의 초과 용량을 활용하고 흥미로운 방식으로 작동합니다. 이를 사용하려면 액세스를 위해 재정적으로 입찰해야 합니다. 스팟 인스턴스는 Amazon의 용량이 초과된 경우에만 사용할 수 있으므로 이 옵션은 앱의 시작 및 종료 시간이 유연한 경우에만 의미가 있습니다. 가격 변경(예: 다른 사람이 액세스에 대해 더 높은 가격을 입찰)으로 인해 인스턴스가 중지되어 결과적으로 워크로드가 완료되지 않는 경우에는 비용이 청구되지 않습니다. 그러나 인스턴스를 직접 종료하는 경우 인스턴스가 실행된 시간에 대해 요금이 부과됩니다. 스팟 인스턴스는 일반적으로 일괄 처리 작업에 사용됩니다.

스탠다드 예약 vs. 컨버터블 예약 vs. 스케쥴 예약:

스탠다드 예약 인스턴스 에는 온디맨드 인스턴스보다 75% 할인된 탄력적 예약이 있습니다. 스탠다드 예약 인스턴스는 지역 간에 이동할 수 없습니다. 예약 인스턴스가 특정 가용 영역에 적용되는지 전체 리전에 적용되는지 선택할 수 있지만 리전을 변경할 수는 없습니다.

컨버터블 예약 인스턴스 는 온디맨드 인스턴스보다 54% 할인된 가격으로 제공되는 인스턴스이지만 언제든지 인스턴스 유형을 수정할 수도 있습니다. 예를 들어, 몇 달 후에 VM을 범용에서 메모리 최적화로 변경해야 할 수도 있다고 생각하지만 아직 확실하지 않습니다. 따라서 향후 VM 유형을 변경하거나 VM 용량을 업그레이드해야 할 수도 있다고 생각되면 컨버터블 예약 인스턴스를 선택하십시오. 하지만 이 옵션을 사용하는 다운그레이드 인스턴스 유형은 없습니다.

스케쥴 예약 인스턴스 는 사용자가 설정한 지정된 타임라인에 따라 예약됩니다. 예를 들어, 학교 시간에만 제공되어야 하는 교육 소프트웨어를 실행하는 경우 스케쥴 예약 인스턴스를 사용할 수 있습니다. 이 옵션을 사용하면 필요한 용량을 반복 일정과 더 잘 일치시켜 비용을 절약할 수 있습니다.

EC2 인스턴스 수명 주기:

다음 표는 주어진 시간에 VM이 있을 수 있는 많은 인스턴스 상태를 강조 표시합니다.

Instance state	Description	Billing
`pending`	The instance is preparing to enter the `running` state. An instance enters the pending state when it launches for the first time, or when it is started after being in the `stopped` state.	Not billed
`running`	The instance is running and ready for use.	Billed
`stopping`	The instance is preparing to be stopped or stop-hibernated.	Not billed if preparing to stop. Billed if preparing to hibernate
`stopped`	The instance is shut down and cannot be used. The instance can be started at any time.	Not billed
`shutting-down`	The instance is preparing to be terminated.	Not billed
`terminated`	The instance has been permanently deleted and cannot be started.	Not billed

참고 : 해지된 예약 인스턴스는 기간이 종료될 때까지 요금이 청구됩니다.

EC2 보안:

Amazon EC2 인스턴스를 배포할 때 귀하는 게스트 운영 체제(업데이트 및 보안 패치 포함), 인스턴스에 설치된 모든 애플리케이션 소프트웨어 또는 유틸리티, AWS 제공 방화벽(보안 그룹이라고 함)의 구성을 관리할 책임이 있습니다.) 각 인스턴스에서.

EC2에서는 인스턴스의 종료 보호가 기본적으로 비활성화되어 있습니다. 즉, 인스턴스가 실수로 종료되는 것을 방지하는 보호 장치가 없습니다. 추가 보호를 원하면 이 기능을 켜야 합니다.

Amazon EC2는 공개 키 암호화를 사용하여 로그인 정보를 암호화하고 해독합니다. 공개 키 암호화는 공개 키를 사용하여 암호와 같은 데이터를 암호화하고 수신자는 개인 키를 사용하여 데이터를 해독합니다. 공개 키와 개인 키를 키 쌍이라고 합니다.

기본 OS를 설치하는 루트 장치 볼륨을 암호화할 수 있습니다. 인스턴스를 생성하는 동안 또는 Bit Locker와 같은 타사 도구를 사용하여 이 작업을 수행할 수 있습니다. 물론 추가 또는 보조 EBS 볼륨도 암호화할 수 있습니다.

기본적으로 AWS Elastic Block Store(EBS) 루트 볼륨이 연결된 EC2 인스턴스는 인스턴스가 종료될 때 함께 삭제됩니다. 그러나 동일한 인스턴스에 연결된 추가 또는 보조 EBS 볼륨은 보존됩니다. 루트 EBS 볼륨은 OS 설치 및 기타 하위 수준 설정 용 이기 때문입니다. 이 규칙은 수정할 수 있지만 일반적으로 이전 볼륨을 사용하는 것보다 새 루트 장치 볼륨으로 새 인스턴스를 부팅하는 것이 더 쉽습니다.

EC2 배치 그룹:

배치 그룹은 EC2 인스턴스 플릿과 관련하여 위험 허용 범위와 네트워크 성능 간의 균형을 유지합니다. 위험에 더 관심을 가질수록 인스턴스가 서로 더 격리되기를 원합니다. 성능에 관심을 가질수록 인스턴스가 서로 더 결합되기를 원합니다.

EC2 배치 그룹에는 세 가지 유형이 있습니다.

1.) 클러스터 배치 그룹

- 클러스터 배치 그룹화는 모든 EC2 인스턴스를 단일 가용 영역에 배치하는 것입니다. 이것은 가능한 가장 짧은 대기 시간이 필요하고 가장 높은 네트워크 처리량이 필요한 애플리케이션에 권장됩니다.

- 특정 인스턴스만 이 그룹으로 시작할 수 있습니다(컴퓨팅 최적화, GPU 최적화, 스토리지 최적화 및 메모리 최적화).

2.) 스프레드 배치 그룹

- Spread Placement Grouping은 오류가 격리되도록 각 개별 EC2 인스턴스를 고유한 하드웨어 위에 배치하는 것입니다.

- VM은 별도의 네트워크 입력과 별도의 전원 요구 사항이 있는 별도의 랙에 있습니다. 스프레드 배치 그룹은 서로 분리된 상태로 유지해야 하는 중요한 인스턴스 수가 적은 애플리케이션에 권장됩니다.

3.) 분할된 배치 그룹

- 분할 배치 그룹화는 스프레드 배치 그룹화와 유사하지만 단일 파티션 내에 여러 EC2 인스턴스를 가질 수 있다는 점에서 다릅니다. 대신 장애는 파티션으로 격리되지만(예: 1 대신 3 또는 4 인스턴스) 네트워크 성능 향상을 위해 근접성의 이점을 누릴 수 있습니다.
- 이 배치 그룹을 사용하면 하나 이상의 지역에 걸쳐 서로 다른 가용 영역 내에서 동일한 하드웨어에 함께 상주하는 여러 인스턴스가 있습니다.
- 위험 허용 범위와 네트워크 성능의 균형을 원하면 분할 배치 그룹을 사용하십시오.

AWS 내의 각 배치 그룹 이름은 고유해야 합니다.

기존 인스턴스를 중지 상태임을 보장하는 배치 그룹으로 이동할 수 있습니다. CLI 또는 AWS SDK를 통해 인스턴스를 이동할 수 있지만 콘솔은 이동할 수 없습니다. 기존 인스턴스의 스냅샷을 만들어 AMI로 변환한 다음 원하는 배치 그룹으로 시작할 수도 있습니다.

Elastic Block Store(EBS)

EBS 단순화:

Amazon EBS 볼륨은 단일 EC2 인스턴스에 연결할 수 있는 내구성 있는 블록 수준 스토리지 디바이스입니다. EBS는 클라우드 기반 가상 하드 디스크라고 생각할 수 있습니다. 인스턴스용 시스템 드라이브나 데이터베이스 애플리케이션용 스토리지와 같이 자주 업데이트해야 하는 데이터의 기본 스토리지로 EBS 볼륨을 사용할 수 있습니다. 지속적인 디스크 스캔을 수행하는 처리량이 많은 애플리케이션에도 사용할 수 있습니다.

EBS 주요 세부사항:

EBS 볼륨은 EC2 인스턴스의 실행 수명과 독립적으로 유지됩니다.

각 EBS 볼륨은 가용 영역 내에서 자동으로 복제되어 구성 요소 장애 및 재해 복구(Standard S3과 유사)로부터 보호합니다.

EBS 스토리지에는 5가지 유형이 있습니다.

- 범용(SSD)
- 프로비저닝된 IOPS(SSD, 속도를 위해 구축됨)
- 처리량에 최적화된 하드 디스크 드라이브(magnetic,대용량 데이터로드용으로 제작됨)
- Cold Hard 디스크 드라이브(magnetic, 덜 자주 액세스하는 워크로드용으로 제작됨)
- Magentic

EBS 볼륨은 99.999% SLA를 제공합니다.

EC2 인스턴스가 어디에 있든 해당 볼륨은 동일한 가용 영역에 있게 됩니다.

EBS 볼륨은 한 번에 하나의 EC2 인스턴스에만 연결할 수 있습니다.

볼륨을 생성한 후 동일한 가용 영역에 있는 모든 EC2 인스턴스에 연결할 수 있습니다.

Amazon EBS는 모든 EBS 볼륨의 스냅샷(백업)을 생성하고 볼륨의 데이터 사본을 여러 가용 영역에 중복 저장되는 S3에 기록하는 기능을 제공합니다.

EBS 스냅샷은 특정 시점의 볼륨 내용을 반영합니다.

이미지(AMI)는 동일한 것이지만 인스턴스를 부팅하는 데 사용할 수 있도록 운영 체제와 부트 로더가 포함되어 있습니다.

AMI는 미리 구워진 실행 가능한 서버로 생각할 수도 있습니다. AMI는 인스턴스를 시작할 때 항상 사용됩니다.

EC2 인스턴스를 프로비저닝할 때 AMI는 실제로 지정하라는 요청을 가장 먼저 받는 항목입니다. 미리 만들어진 AMI를 선택하거나 EBS 스냅샷에서 직접 만든 AMI를 선택할 수 있습니다.

다음 기준을 사용하여 AMI를 선택할 수도 있습니다.

- 운영 체제
- 아키텍처(32비트 또는 64비트)
- 리전(지역)
- 실행 권한
- 루트 장치 저장소(아래 관련 섹션에서 자세히 설명)

AMI를 완전히 새로운 리전에 복사할 수 있습니다.

AMI를 새 리전에 복사할 때 Amazon은 시작 권한, 사용자 정의 태그 또는 Amazon S3 버킷 권한을 원본 AMI에서 새 AMI로 복사하지 않습니다. 이러한 세부 정보가 새 지역의 인스턴스에 대해 올바르게 설정되었는지 확인해야 합니다.

크기 및 스토리지 유형을 포함하여 즉시 EBS 볼륨을 변경할 수 있습니다.

SSD VS HDD:

SSD 지원 볼륨은 주요 성능 속성이 IOPS인 빈번한 읽기/쓰기 작업과 관련된 트랜잭션 워크로드를 위해 구축되었습니다.
경험 법칙 : 워크로드가 IOPS가 많습니까? SSD

HDD 지원 볼륨은 처리량(MiB/s로 측정)이 IOPS보다 더 나은 성능 척도인 대규모 스트리밍 워크로드용으로 구축되었습니다.
경험의 법칙 : 워크로드가 처리량이 많습니까? HDD

Solid State Drives (SSD)	Hard Disk Drives (HDD)
General Purpose SSD Balanced for economy and performance	**Throughput Optimized HDD**: Inexpensive, for high use, intensive workloads
Provisioned IOPS SSD High performance, for important applications	**Cold HDD** Cheap, used for infrequent access

EBS 스냅샷:

EBS 스냅샷은 볼륨의 특정 시점 복사본입니다. 스냅샷은 디스크의 현재 상태와 디스크 안의 모든 상태를 보여주는 사진으로 생각할 수 있습니다.

스냅샷은 생성된 리전(지역)으로 제한됩니다.

스냅샷은 마지막 스냅샷이 생성된 이후의 변경 상태만 캡처합니다. 이것은 서버의 전체 상태가 아니라 각각의 새 스냅샷에 기록되는 내용입니다.

이 때문에 첫 번째 스냅샷이 생성되는 데 시간이 걸릴 수 있습니다. 첫 번째 스냅샷의 상태 변경이 전체 새 볼륨이기 때문입니다. 그 후에야 비교할 이전 항목이 있기 때문에 델타가 캡처됩니다.

EBS 스냅샷은 비동기식으로 발생하므로 스냅샷이 발생하는 동안 볼륨을 정상적으로 사용할 수 있습니다.

미래의 루트 장치에 대한 스냅샷을 생성할 때 스냅샷을 만들기 전에 원래 장치가 있는 실행 중인 인스턴스를 중지하는 것이 모범 사례로 간주됩니다.

EC2 인스턴스와 볼륨을 다른 가용 영역으로 이동하는 가장 쉬운 방법은 스냅샷을 만드는 것입니다.

스냅샷에서 이미지를 생성할 때 새 이미지에 대해 다른 볼륨 유형을 배포하려면(예: 범용 SSD -> 처리량 최적화 HDD) 새 이미지에 대한 가상화가 하드웨어 지원인지 확인해야 합니다.

EC2 인스턴스 사본 생성에 대한 간단한 요약: 이전 인스턴스 -> 스냅샷 -> 이미지(AMI) -> 새 인스턴스

등록된 AMI의 루트 디바이스로 사용되는 EBS 볼륨의 스냅샷은 삭제할 수 없습니다. 원본 스냅샷이 삭제된 경우 AMI는 이를 기반으로 새 인스턴스를 생성할 수 없습니다. 이러한 이유로 AWS는 시스템에 중요할 수 있는 EBS 스냅샷을 실수로 삭제하지 않도록 보호합니다. 등록된 AMI에 연결된 EBS 스냅샷을 삭제하려면 먼저 AMI를 제거한 다음 스냅샷을 삭제할 수 있습니다.

EBS 루트 디바이스 스토리지:

모든 AMI 루트 볼륨(EC2의 OS가 설치된 위치)은 EBS 지원 또는 인스턴스 스토어 지원의 두 가지 유형입니다.

인스턴스 스토어 지원 루트 볼륨을 사용하던 EC2 인스턴스를 삭제하면 루트 볼륨도 삭제됩니다. 그러나 추가 또는 보조 볼륨은 유지됩니다.

EBS 지원 루트 볼륨을 사용하는 경우 인스턴스가 오프라인이 될 때 루트 볼륨은 EC2 인스턴스와 함께 종료되지 않습니다. EBS 지원 볼륨은 인스턴스 스토어 지원 볼륨과 같은 임시 저장 장치가 아닙니다.

EBS 지원 볼륨은 이름에서 알 수 있듯이 AWS EBS 스냅샷에서 시작됩니다.

인스턴스 스토어 지원 볼륨은 AWS S3 저장 템플릿에서 시작됩니다. 일시적이므로 인스턴스를 종료할 때 주의하십시오!

인스턴스 저장소 지원 루트 장치에 대한 보조 인스턴스 저장소는 서버의 원래 프로비저닝 중에 설치해야 합니다. 사실 이후에는 더 추가할 수 없습니다. 그러나 서버 생성 후 동일한 인스턴스에 EBS 볼륨을 추가할 수 있습니다.

Instance Store 볼륨의 이러한 단점이 있는데 왜 하나를 선택해야 할까요? IOPS 비율이 매우 높기 때문입니다. 따라서 인스턴스 스토어는 데이터 지속성을 제공할 수 없지만 EBS와 같은 네트워크 연결 스토리지에 비해 훨씬 더 높은 IOPS를 제공할 수 있습니다.

또한 인스턴스 저장소는 버퍼, 캐시, 스크래치 데이터 및 기타 임시 콘텐츠와 같이 자주 변경되는 정보의 임시 저장 또는 로드 밸런싱된 웹 서버 풀과 같이 인스턴스 집합 전체에 복제되는 데이터에 이상적입니다. .

언제 다른 것보다 하나를 사용해야합니까?

- DB 데이터, 중요 로그 및 애플리케이션 구성에 EBS를 사용합니다.
- 진행 중인 데이터, 중요하지 않은 로그 및 일시적인 애플리케이션 상태에 대해 인스턴스 스토리지를 사용합니다.
- 입력 데이터 세트 및 처리된 결과와 같은 시스템 간에 공유되는 데이터 또는 시작할 때 각각의 새 시스템에 필요한 정적 데이터에 S3를 사용합니다.
-

EBS 암호화:

EBS 암호화는 자체 키 관리 인프라를 구축, 유지 관리 및 보호할 필요가 없는 EBS 리소스에 대한 간단한 암호화 솔루션을 제공합니다.

암호화된 볼륨 및 스냅샷을 생성할 때 AWS Key Management Service(AWS KMS) 고객 마스터 키(CMK)를 사용합니다.

EC2 인스턴스의 루트 디바이스와 보조 볼륨을 모두 암호화할 수 있습니다. 암호화된 EBS 볼륨을 생성하여 지원되는 인스턴스 유형에 연결하면 다음 유형의 데이터가 암호화됩니다.

- 볼륨 내부의 미사용 데이터
- 볼륨과 인스턴스 간에 이동하는 모든 데이터
- 볼륨에서 생성된 모든 스냅샷
- 해당 스냅샷에서 생성된 모든 볼륨

EBS는 AES-256 알고리즘을 사용하여 데이터 키로 볼륨을 암호화합니다.

암호화된 볼륨의 스냅샷도 자연스럽게 암호화됩니다. 암호화된 스냅샷에서 복원된 볼륨도 암호화됩니다. 암호화되지 않은 스냅샷만 공유할 수 있습니다.

루트 장치를 암호화하는 이전 방법은 프로비저닝된 EC2 인스턴스의 스냅샷을 생성하는 것이었습

니다. 해당 스냅샷의 복사본을 만드는 동안 복사본을 만드는 동안 암호화를 활성화했습니다. 마지막으로 복사본이 암호화되면 암호화된 복사본에서 AMI를 생성하고 루트 장치에서 암호화된 EC2 인스턴스를 사용했습니다. 이것이 얼마나 복잡하기 때문에 이제 EC2 프로비저닝 옵션의 일부로 루트 장치를 간단히 암호화할 수 있습니다.

탄력적 네트워크 인터페이스(ENI)

ENI 단순화:

탄력적 네트워크 인터페이스는 가상 네트워크 카드를 나타내는 네트워킹 구성 요소입니다. 새 인스턴스를 프로비저닝하면 ENI가 자동으로 연결되며 원하는 경우 추가 네트워크 인터페이스를 만들고 구성할 수 있습니다. 한 인스턴스에서 다른 인스턴스로 네트워크 인터페이스를 이동하면 네트워크 트래픽이 새 인스턴스로 리디렉션됩니다.

ENI 주요 세부사항:

ENI는 주로 저예산, 고가용성 네트워크 솔루션에 사용됩니다.

그러나 높은 네트워크 처리량이 필요하다고 생각되면 Enhanced Networking ENI를 사용할 수 있습니다.

향상된 네트워킹 ENI는 단일 루트 I/O 가상화를 사용하여 지원되는 인스턴스 유형에 고성능 네트워킹 기능을 제공합니다. SR-IOV는 더 높은 I/O와 더 낮은 처리량을 제공하며 더 높은 대역폭, 더 높은 PPS(초당 패킷) 성능 및 지속적으로 낮은 인스턴스 간 대기 시간을 보장합니다. SR-IOV는 인터페이스를 단일 인스턴스 전용으로 지정하고 하이퍼바이저의 일부를 효과적으로 우회하여 더 나은 성능을 제공함으로써 이를 수행합니다.

더 많은 ENI를 추가한다고 해서 반드시 네트워크 처리 속도가 빨라지는 것은 아니지만 Enhanced Networking ENI는 빨라질 것입니다.

Enhanced Networking ENI 및 더 나은 네트워크 성능을 제공하는 데 추가 비용은 없습니다. 유일한 단점은 모든 EC2 인스턴스 패밀리 및 유형에서 Enhanced Networking ENI를 사용할 수 없다는 것입니다.

다음과 같은 방법으로 네트워크 인터페이스를 EC2 인스턴스에 연결할 수 있습니다.

- 실행 중일 때(hot attach)
- 중지된 경우(warm attach)
- 인스턴스가 시작될 때(cold attach).

ENI가 올바르게 구성된 상태에서 EC2 인스턴스가 실패하면 사용자(또는 사용자를 대신하여 실행되는 코드)가 네트워크 인터페이스를 상시 대기 인스턴스에 연결할 수 있습니다. ENI 인터페이스는 자체 프라이빗 IP 주소, 탄력적 IP 주소 및 MAC 주소를 유지하므로 교체 인스턴스에 네트워크 인터페이스를 연결하는 즉시 네트워크 트래픽이 대기 인스턴스로 흐르기 시작합니다. 사용자는 인스턴스가 실패하는 시간과 네트워크 인터페이스가 대기 인스턴스에 연결되는 시간 사이에 연결이 잠시 끊어지는 것을 경험하게 되지만 VPC 라우팅 테이블이나 DNS 서버를 변경할 필요는 없습니다.

기계 학습 및 고성능 컴퓨팅과 함께 작동하는 인스턴스의 경우 EFA(Elastic Fabric Adaptor)를 사용하십시오. EFA는 위의 사용 사례에서 필요한 작업을 가속화합니다. EFA는 클라우드 기반 고

성능 컴퓨팅 시스템에서 전통적으로 사용되는 TCP 전송보다 더 낮고 일관된 대기 시간과 더 높은 처리량을 제공합니다.

EFA는 또한 ML 및 HPC 애플리케이션이 일반적으로 OS를 통해 라우팅되지 않고 Elastic Fabric Adapter와 직접 인터페이스할 수 있도록 하는 OS 우회(Linux에만 해당)를 사용할 수 있습니다. 이것은 엄청난 성능 향상을 제공합니다.

네트워크 인터페이스에서 VPC 흐름 로그를 활성화하여 네트워크 인터페이스로 들어오고 나가는 IP 트래픽에 대한 정보를 캡처할 수 있습니다.

보안 그룹

단순화된 보안 그룹:

보안 그룹은 EC2로 액세스(SSH, HTTP, RDP 등)를 제어하는 데 사용됩니다. 인스턴스가 인바운드 및 아웃바운드 트래픽을 제어할 수 있도록 가상 방화벽 역할을 합니다. VPC에서 인스턴스를 시작할 때 최대 5개의 보안 그룹을 인스턴스에 할당할 수 있으며 보안 그룹은 서브넷 수준이 아닌 인스턴스 수준에서 작동합니다.

보안 그룹 주요 세부 정보:

보안 그룹은 인스턴스의 인바운드 및 아웃바운드 트래픽(EC2 인스턴스용 방화벽 역할)을 제어하는 반면 NACL은 서브넷의 인바운드 및 아웃바운드 트래픽(서브넷용 방화벽 역할)을 제어합니다. 보안 그룹은 일반적으로 EC2 인스턴스에서 사용할 수 있는 포트 목록을 제어하고 NACL은 전체 VPC에 연결할 수 있는 네트워크 또는 IP 주소 목록을 제어합니다.

보안 그룹을 변경할 때마다 해당 변경 사항이 즉시 적용됩니다.

인바운드 규칙을 생성할 때마다 아웃바운드 규칙이 즉시 생성됩니다. 이는 보안 그룹이 상태 를 저장하기 때문 입니다. 즉, 보안 그룹에 대한 수신 규칙을 생성할 때 일치하는 해당 송신 규칙이 생성됩니다. 이는 상태 비저장 이고 인바운드 및 아웃바운드 규칙을 모두 생성하기 위해 수동 개입이 필요한 NACL과 대조됩니다.

보안 그룹 규칙은 허용을 기반으로 하며 보안 그룹과 관련하여 거부 개념이 없습니다. 즉, 보안 그룹을 통해 특정 포트를 명시적으로 거부하거나 블랙리스트에 추가할 수 없으며 허용 목록에서 제외하여 암시적으로만 거부할 수 있습니다.

위의 세부 사항 때문에 모든 것이 기본적으로 차단됩니다. 특정 포트에 대해 의도적으로 액세스를 허용해야 합니다.

보안 그룹은 단일 VPC에만 해당하므로 여러 VPC 간에 보안 그룹을 공유할 수 없습니다. 그러나 보안 그룹을 복사하여 동일한 AWS 계정에 대해 다른 VPC에서 동일한 규칙으로 새 보안 그룹을 생성할 수 있습니다.

보안 그룹은 지역적이며 AZ에 걸쳐 있을 수 있지만 지역 간은 될 수 없습니다.

API 엔드포인트 또는 DB 백엔드와 같은 다른 서비스에 서버를 연결해야 하는 경우 아웃바운드 규칙이 존재합니다. 트래픽이 EC2를 떠나 다른 AWS 서비스로 들어갈 수 있도록 올바른 포트에 대해 ALLOW 규칙을 활성화해야 합니다.

하나의 EC2 인스턴스에 여러 보안 그룹을 연결할 수 있으며 하나의 보안 그룹 아래에 여러 EC2 인스턴스를 가질 수 있습니다.

보안 그룹의 소스(기본적으로 가상 방화벽을 우회할 수 있는 사람)를 단일 /32 IP 주소, IP 범위 또는 별도의 보안 그룹으로 지정할 수 있습니다.

보안 그룹으로 특정 IP 주소를 차단할 수 없습니다(대신 NACL 사용).

AWS에 요청을 제출하여 보안 그룹 한도를 늘릴 수 있습니다.

WAF(웹 애플리케이션 방화벽)

단순화된 WAF:

AWS WAF는 CloudFront, API Gateway, Application Load Balancer, EC2 및 AWS 환경에 대한 기타 계층 7 진입점에 바인딩된 HTTP 요청을 허용하거나 차단할 수 있는 웹 애플리케이션입니다. AWS WAF를 사용하면 SQL 주입 또는 사이트 간 스크립팅과 같은 일반적인 공격 패턴을 차단하는 보안 규칙과 정의할 수 있는 특정 트래픽 패턴을 필터링하는 규칙을 생성할 수 있으므로 트래픽이 애플리케이션에 도달하는 방식을 제어할 수 있습니다. WAF의 기본 규칙 집합은 OWASP Top 10 보안 위험과 같은 문제를 해결하고 새로운 취약점이 발견될 때마다 정기적으로 업데이트됩니다.

WAF 키 세부 정보:

위에서 언급했듯이 WAF는 레이어 7 방화벽으로 작동합니다. 이를 통해 URL 쿼리 문자열 매개변수와 같은 세부적인 웹 기반 조건을 모니터링할 수 있습니다. 이 수준의 세부 정보는 요청이 AWS 환경으로 전달될 때 부정 행위와 정직한 문제를 모두 감지하는 데 도움이 됩니다.

WAF를 사용하면 어떤 IP 주소가 어떤 종류의 요청을 하거나 어떤 종류의 콘텐츠에 액세스할 수 있는 지와 같은 조건을 설정할 수 있습니다.

이러한 조건을 기반으로 해당 엔드포인트는 요청된 콘텐츠를 제공하여 요청을 허용하거나 HTTP 403 금지 상태를 반환합니다.

가장 간단한 수준에서 AWS WAF를 사용하면 다음 동작 중 하나를 선택할 수 있습니다.

- 지정한 요청을 제외한 모든 요청 허용 : CloudFront 또는 Application Load Balancer가 공개 웹 사이트에 대한 콘텐츠를 제공하도록 하고 공격자의 요청도 차단하려는 경우에 유용합니다.
- 지정한 요청을 제외한 모든 요청 차단 : 사용자가 웹사이트를 탐색하는 데 사용하는 IP 주소와 같은 웹 요청의 속성으로 쉽게 식별할 수 있는 제한된 웹사이트에 대한 콘텐츠를 제공하려는 경우에 유용합니다.
- 지정한 속성과 일치하는 요청 계산 : 웹 요청의 새 속성을 기반으로 요청을 허용하거나 차단하려는 경우 먼저 해당 요청을 허용하거나 차단하지 않고 해당 속성과 일치하는 요청을 계산하도록 AWS WAF를 구성할 수 있습니다. 이를 통해 웹 사이트에 대한 모든 트래픽을 차단하도록 AWS WAF를 실수로 구성하지 않았음을 확인할 수 있습니다. 올바른 속성을 지정했다고 확신하면 동작을 변경하여 요청을 허용하거나 차단할 수 있습니다.

WAF 보호 기능:

액세스를 제한하는 데 사용할 수 있는 다양한 요청 특성:

- 요청이 시작된 IP 주소
- 요청이 시작된 국가
- 요청 헤더에서 찾은 값
- 요청에 나타나는 모든 문자열(특정 문자열 또는 정규식 패턴과 일치하는 문자열)
- 요청의 길이
- SQL 코드의 존재(SQL 주입 시도 가능성)
- 모든 스크립트 존재(교차 사이트 스크립팅 시도 가능성)

또한 NACL을 사용하여 악성 IP 주소를 차단하고, SQL 주입/XSS를 방지하고, 특정 국가의 요청을 차단할 수 있습니다. 그러나 수비를 깊이 연습하기 좋은 형태입니다.

WAF 수준에서 악의적인 사용자를 거부하거나 차단하면 가장 바깥쪽 경계에서 AWS 에코시스템을 보호할 수 있는 추가적인 이점이 있습니다.

CloudWatch

간소화된 CloudWatch:

Amazon CloudWatch는 모니터링 및 관찰 가능성 서비스입니다. 데이터와 실행 가능한 통찰력을 제공하여 애플리케이션을 모니터링하고, 시스템 전체의 성능 변화에 대응하고, 리소스 활용을 최적화하고, 운영 상태에 대한 통합 보기를 얻습니다.

CloudWatch 키 세부 정보:

CloudWatch는 모니터링 및 운영 데이터를 로그, 메트릭 및 이벤트 형태로 수집합니다.

CloudWatch를 사용하여 환경의 비정상적인 동작을 감지하고, 경보를 설정하고, 로그와 지표를 나란히 시각화하고, 자동화된 조치를 취하고, 문제를 해결하고, 애플리케이션을 원활하게 실행하기 위한 통찰력을 발견할 수 있습니다.

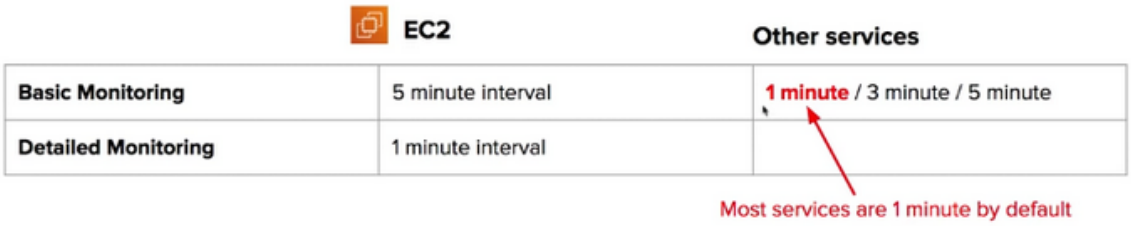
컴퓨팅 도메인 내에서 CloudWatch는 EC2 인스턴스, Autoscaling 그룹, Elastic Load Balancer 및 Route53 상태 확인의 상태에 대해 알려줄 수 있습니다. 스토리지 및 콘텐츠 전송 도메인 내에서 CloudWatch는 EBS 볼륨, 스토리지 게이트웨이 및 CloudFront의 상태에 대해 알려줄 수 있습니다.

EC2와 관련하여 CloudWatch는 CPU, 네트워크, 디스크와 같은 호스트 수준 지표 및 기본 하이퍼바이저의 상태와 같은 통찰력에 대한 상태 확인만 모니터링할 수 있습니다.

CloudWatch는 CloudTrail 이 아니므로 CloudTrail만이 보안 및 감사 이유로 AWS 액세스를 모니터

링할 수 있다는 점을 아는 것이 중요합니다. CloudWatch는 성능에 관한 모든 것입니다. CloudTrail은 감사에 관한 모든 것입니다.

CloudWatch with EC2는 기본적으로 5분마다 이벤트를 모니터링하지만 세부 모니터링을 사용하는 경우 1분 간격으로 모니터링할 수 있습니다.



통찰력을 위해 CloudWatch 대시보드를 사용자 지정할 수 있습니다.

Linux 및 Windows 기반 인스턴스에 모두 설치할 수 있는 다중 플랫폼 CloudWatch 에이전트가 있습니다. 이 에이전트를 사용하면 CPU별 코어와 같은 하위 리소스 메트릭을 포함하여 수집할 메트릭을 선택할 수 있습니다. 이 단일 에이전트를 사용하여 Amazon EC2 인스턴스와 온프레미스 서버에서 시스템 지표와 로그 파일을 모두 수집할 수 있습니다.

다음 지표는 CloudWatch를 통해 EC2 인스턴스에서 수집되지 않습니다.

- 메모리 활용
- 디스크 스왑 활용
- 디스크 공간 활용도
- 페이지 파일 활용
- 로그 수집

위의 정보가 필요한 경우 공식 CloudWatch 에이전트를 통해 검색하거나 사용자 지정 지표를 생성하고 사용자 지정 스크립트를 통해 직접 데이터를 보낼 수 있습니다.

CloudWatch의 주요 목적:

- 측정항목 수집
- 로그 수집
- 이벤트 수집
- 경보 생성
- 대시보드 만들기

CloudWatch 로그:

Amazon CloudWatch Logs를 사용하여 Amazon EC2 인스턴스, AWS CloudTrail, Amazon Route 53 및 기타 소스의 로그 파일을 모니터링, 저장 및 액세스할 수 있습니다. 그런 다음 CloudWatch Logs에서 연결된 로그 데이터를 검색할 수 있습니다.

확장성이 뛰어난 단일 서비스에서 사용하는 모든 시스템, 애플리케이션 및 AWS 서비스의 로그를 중앙 집중화하는 데 도움이 됩니다.

CloudWatch Logs의 논리 단위를 함께 조인하도록 로그 그룹을 생성할 수 있습니다.

추가 통찰력을 위해 사용자 정의 로그 파일을 스트리밍할 수 있습니다.

CloudWatch 이벤트:

Amazon CloudWatch Events는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트의 거의 실시간 스트림을 제공합니다.

예를 들어 알람을 사용하여 문제가 발생했음을 알리는 동안 이벤트를 사용하여 람다를 트리거할 수 있습니다.

CloudWatch 경보:

CloudWatch 경보는 알림을 보내거나 정의한 규칙에 따라 모니터링 중인 리소스를 자동으로 변경합니다.

예를 들어, 설정된 청구 임계값 초과와 같은 알림을 트리거하는 사용자 지정 CloudWatch 경보를 생성할 수 있습니다.

CloudWatch 경보에는 다음 중 하나의 두 가지 상태가 있습니다 ok,alarm

CloudWatch 지표:

CloudWatch 지표는 시간순으로 정렬된 데이터 포인트 세트를 나타냅니다.

기본적으로 시간별 CPU 사용률과 같이 모든 것이 정상인지 확인하는 데 도움이 되도록 시간이 지남에 따라 모니터링할 수 있는 변수입니다.

CloudWatch 지표를 사용하면 1분 미만 간격으로 초당까지 고해상도 지표를 추적할 수 있습니다.

CloudWatch 대시보드:

CloudWatch 대시보드는 단일 보기에서 리소스를 모니터링하는 데 사용할 수 있는 CloudWatch 콘솔의 사용자 지정 가능한 홈 페이지입니다.

이러한 대시보드는 CloudWatch 지표 및 CloudWatch 경보와 통합되어 AWS 리소스에 대한 지표 및 경보의 사용자 지정 보기를 생성합니다.

CloudTrail

간소화된 CloudTrail:

AWS CloudTrail은 AWS 계정의 거버넌스, 규정 준수, 운영 감사 및 위험 감사를 지원하는 서비스입니다. 이를 통해 AWS 인프라 전반의 작업과 관련된 계정 활동을 기록하고 지속적으로 모니터링하고 유지할 수 있습니다. CloudTrail은 AWS Management Console, AWS SDK, 명령줄 도구, API 호출 및 기타 AWS 서비스를 통해 수행한 작업을 포함하여 AWS 계정 활동의 이벤트 기록을 제공합니다. 지역 서비스이지만 모든 지역에서 추적을 수집하도록 CloudTrail을 구성할 수 있습니다.

CloudTrail 키 세부 정보:

CloudTrail 이벤트는 API 호출 또는 활동을 기록합니다.

CloudTrail 이벤트는 이벤트 기록에 지난 90일 간의 이벤트를 저장합니다. 이것은 기본적으로 활성화되어 있으며 추가 비용이 없습니다.

이 이벤트 기록은 보안 분석, 리소스 변경 추적 및 문제 해결을 단순화합니다.

CloudTrail에 기록할 수 있는 이벤트에는 관리 이벤트와 데이터 이벤트의 두 가지 유형이 있습니다.

관리 이벤트는 AWS 계정의 리소스에서 수행되는 관리 작업에 대한 정보를 제공합니다.

관리 이벤트는 AWS에 있을 때 사람들이 일반적으로 수행하는 것으로 생각하십시오. 예:

- 사용자 로그인
- 정책 변경
- 새로 생성된 보안 구성
- 로깅 규칙 삭제

데이터 이벤트는 리소스에서 또는 리소스에서 수행되는 리소스 작업에 대한 정보를 제공합니다.

데이터 이벤트를 다양한 AWS 엔드포인트에 도달할 때 소프트웨어에서 일반적으로 수행하는 작업으로 생각하십시오. 예:

- S3 객체 수준 API 활동
- Lambda 함수 실행 활동

기본적으로 CloudTrail은 관리 이벤트를 기록하지만 데이터 이벤트는 기록하지 않습니다.

기본적으로 CloudTrail 이벤트 로그 파일은 Amazon S3 서버 측 암호화(SSE)를 사용하여 암호화됩니다. AWS Key Management Service(AWS KMS) 키로 로그 파일을 암호화하도록 선택할 수도 있습니다. 이러한 로그는 S3에 저장되므로 Amazon S3 수명 주기 규칙을 정의하여 로그 파일을 자동으로 보관하거나 삭제할 수 있습니다. 로그 파일 전송 및 검증에 대한 알림을 원하는 경우 Amazon SNS 알림을 설정할 수 있습니다.

Elastic File System(EFS)

단순화된 EFS:

EFS는 AWS 내에서 사용할 수 있는 단순하고 완전 관리되는 탄력적 NFS 파일 시스템을 제공합니다. EFS는 애플리케이션을 중단하지 않고 파일을 추가하거나 제거할 때 파일 시스템 스토리지 용량을 자동으로 즉시 확장 또는 축소합니다.

EFS 키 세부 정보:

EFS에서 저장 용량은 탄력적이며(자동으로 증가 및 축소) 파일 추가 또는 제거에 따라 크기가 변경됩니다.

EBS가 하나의 EBS 볼륨을 하나의 인스턴스에 탑재하는 동안 여러 EC2 인스턴스에 하나의 EFS 볼륨을 연결할 수 있습니다.

EC2 인스턴스는 NFSv4 프로토콜을 사용하여 원격 파일 시스템과 통신합니다. 따라서 보안 그룹(EC2 방화벽 규칙)에 대한 NFS 포트를 열어 해당 포트에서 인바운드 트래픽을 허용해야 합니다.

EFS 볼륨 내에서 탑재 대상 상태는 탑재에 사용할 수 있는 인스턴스를 알려줍니다.

EFS를 사용하면 사용한 만큼만 비용을 지불하는 스토리지에 대해서만 비용을 지불합니다. 사전 프로비저닝이 필요하지 않습니다.

EFS는 페타바이트까지 확장할 수 있으며 수천 개의 동시 NFS 연결을 지원할 수 있습니다.

데이터는 한 지역의 여러 AZ에 저장되며 EFS는 쓰기 후 읽기 일관성을 보장합니다.

하나의 서버가 아닌 여러 서버에서 액세스하는 파일 스토리지에 가장 적합합니다.

Windows용 Amazon FSx

단순화된 Windows용 Amazon FSx:

Windows 파일 서버용 Amazon FSx는 완전 관리형 기본 Microsoft 파일 시스템을 제공합니다.

Windows용 Amazon FSx 키 세부 정보:

FSx for Windows를 사용하면 AWS에서 파일 스토리지가 필요한 Windows 기반 애플리케이션을 쉽게 이동할 수 있습니다.

Windows Server를 기반으로 하며 Microsoft 기반 애플리케이션용으로만 존재하므로 SMB 기반 파일 스토리지가 필요한 경우 FSx를 선택하십시오.

Windows용 FSx는 또한 온프레미스 서버와 AWS 간의 연결을 허용하므로 동일한 온프레미스 서버에서도 Amazon FSx를 사용할 수 있습니다.

Microsoft Active Directory를 사용하여 파일 시스템에 인증할 수 있습니다.

Windows용 Amazon FSx는 데이터 보호를 보장하기 위해 여러 수준의 보안 및 규정 준수를 제공합니다. Amazon FSx는 저장 데이터와 전송 데이터를 자동으로 암호화합니다.

EC2뿐만 아니라 다양한 컴퓨팅 리소스에서 Windows용 Amazon FSx에 액세스할 수 있습니다.

단일 AZ 또는 다중 AZ 구성에서 Windows용 Amazon FSx를 배포할 수 있습니다.

요구 사항에 따라 저장 장치에 SSD 또는 HDD를 사용할 수 있습니다.

Windows용 FSx는 일일 자동 백업을 지원하며 관리자도 필요할 때 백업을 수행합니다.

Windows용 FSx는 중복 콘텐츠를 제거하고 공통 콘텐츠를 압축합니다.

기본적으로 모든 데이터는 저장 시 암호화됩니다.

Lustre용 Amazon FSx

간소화된 Lustre용 Amazon FSx:

Amazon FSx for Lustre를 사용하면 고성능 컴퓨팅 애플리케이션을 위한 오픈 소스 Lustre 파일 시스템을 쉽고 비용 효율적으로 시작하고 실행할 수 있습니다. FSx for Lustre를 사용하면 초당 최대 수백 기가바이트의 처리량, 수백만 IOPS 및 밀리초 미만의 대기 시간으로 대규모 데이터 세트를 처리할 수 있는 파일 시스템을 시작하고 실행할 수 있습니다.

Lustre 키 세부 정보용 Amazon FSx:

FSx for Lustre는 Amazon Linux, Amazon Linux 2, RHEL(Red Hat Enterprise Linux), CentOS, SUSE Linux 및 Ubuntu를 포함하여 가장 널리 사용되는 Linux 기반 AMI와 호환됩니다.

Lustre 파일 시스템은 일반적으로 컴퓨팅 클러스터에서 실행되는 고성능 컴퓨팅 워크로드용으로 설계되었으므로 요구 사항이 이 사용 사례와 일치하지 않는 경우 일반 Linux 파일 시스템에 대해 EFS를 선택합니다.

FSx Lustre는 자체적으로 S3에 직접 데이터를 저장하고 검색하는 기능이 있습니다.

관계형 데이터베이스 서비스(RDS)

단순화된 RDS:

RDS는 AWS에서 관계형 데이터베이스를 쉽게 설정, 운영 및 확장할 수 있는 관리형 서비스입니다. 비용 효율적이고 크기 조정 가능한 용량을 제공하는 동시에 하드웨어 프로비저닝, 데이터베이스 설정, 패치 적용 및 백업과 같은 시간 소모적인 관리 작업을 자동화하거나 아웃소싱합니다.

RDS 주요 세부사항:

RDS는 6가지 종류로 제공됩니다.

- SQL 서버
- Oracle
- MySQL 서버
- PostgreSQL
- MariaDB
- Aurora

RDS를 다양한 DB가 그 위에 올려진 DB 엔진이라고 생각하면 됩니다.

확장 시 RDS에는 두 가지 주요 기능이 있습니다.

- 성능 향상을 위한 읽기 복제
- 고가용성을 위한 다중 AZ

데이터베이스 세계에서 OLTP(온라인 트랜잭션 처리) 는 수행할 쿼리 유형 측면에서 OLAP(온라인 분석 처리) 와 다릅니다 . OLTP는 궁극적으로 플랫폼 또는 애플리케이션의 핵심 기능을 구성하는 비즈니스 로직을 위한 데이터를 제공합니다. OLAP은 회사로서 더 나은 전략적 결정을 내리기 위해 저장한 데이터에 대한 통찰력을 얻는 것입니다.

RDS는 가상 머신에서 실행되지만 해당 머신에 대한 액세스 권한이 없습니다. SSH를 통해 RDS 인스턴스에 연결할 수 없으므로 OS를 패치할 수 없습니다. 이는 AWS가 RDS의 보안 및 유지 관리를 책임진다는 것을 의미합니다. RDS 엔진이 아닌 기본 서버를 직접 관리해야 하거나 관리하려는 경우 EC2 인스턴스를 데이터베이스로 프로비저닝할 수 있습니다.

VM에 직접 액세스할 수 없다고 해서 RDS가 서버가 없는 것은 아닙니다. 그러나 틈새 목적으로 사용되는 Aurora 서버리스(아래 설명)가 있습니다.

SQS 대기열은 애플리케이션이 높은 쓰기 로드로 어려움을 겪고 있는 경우 보류 중인 데이터베이스 쓰기를 저장하는 데 사용할 수 있습니다. 그런 다음 데이터베이스가 처리할 준비가 되면 이러한 쓰기를 데이터베이스에 추가할 수 있습니다. IOPS를 추가하는 것도 도움이 되지만 이것만으로는 쓰기 손실 가능성을 완전히 제거할 수 없습니다. 그러나 대기열은 DB에 대한 쓰기가 손실되지 않도록 합니다.

RDS 다중 AZ:

AWS의 재해 복구는 항상 리소스의 대기 복사본이 별도의 지리적 영역에서 유지되도록 합니다. 이렇게 하면 원래 리소스가 있는 곳에 재해(자연 재해, 정치적 갈등 등)가 발생하더라도 복사본은 영향을 받지 않습니다.

다중 AZ DB 인스턴스를 프로비저닝하면 Amazon RDS가 자동으로 기본 DB 인스턴스를 생성하고 다른 가용 영역(AZ)의 대기 인스턴스에 데이터를 동기식으로 복제합니다. 각 AZ는 물리적으로 별개의 독립적인 인프라에서 실행되며 매우 안정적으로 설계되었습니다.

다중 AZ 구성에서 EC2는 연결 문자열로 마스킹된 DNS 주소를 사용하여 RDS 데이터 저장소에 연결합니다. 기본 DB에 장애가 발생하면 다중 AZ는 해당 장애를 감지하고 보조 DB를 가리키도록 DNS 주소를 자동으로 업데이트할 만큼 충분히 스마트합니다. 수동 개입이 필요하지 않으며 AWS가 DNS의 IP 주소 교환을 처리합니다.

다중 AZ는 aurora를 제외한 모든 DB 버전에 대해 지원됩니다. Aurora는 자체적으로 완전히 내결함성이 있기 때문입니다.

다중 AZ 기능은 지역이 아닌 가용 영역 전체에서 고가용성을 허용합니다.

장애 조치 중에 복구된 이전 기본이 새 보조가 되고 승격된 보조가 기본이 됩니다. 원래 DB가 복구되면 두 DB가 서로를 한 번 미러링하여 실패한 이전 기본 데이터베이스가 놓쳤을 수 있는 새 데이터를 동기화하는 동기화 프로세스가 시작됩니다.

기본 인스턴스를 재부팅하여 다중 AZ 설정에 대한 장애 조치를 강제할 수 있습니다.

다중 AZ RDS 구성을 사용하면 대기에서 백업이 수행됩니다.

RDS 읽기 전용 복제본:

읽기 복제는 성능 향상을 위해 독점적으로 사용됩니다.

읽기 전용 복제본 구성을 사용하면 EC2는 DNS 주소를 사용하여 RDS 백엔드에 연결하고 마스터 데이터베이스에서 수신한 모든 쓰기는 마스터의 완벽한 복사본이 되도록 보조 DB에도 전달됩니다. 이는 보조 DB에 동일한 데이터를 조회할 수 있기 때문에 마스터의 트랜잭션 수를 줄이는 전반적인 효과가 있습니다.

그러나 마스터 DB에 장애가 발생하면 자동 장애 조치가 없습니다. 자체적으로 마스터가 되도록 읽기 전용 복제본 중 하나와 동기화할 새 연결 문자열을 수동으로 생성해야 합니다. 그런 다음 읽기 전용 복제본을 가리키도록 EC2 인스턴스를 업데이트해야 합니다. 읽기 복제를 사용하여 최대 5개의 마스터 DB 복사본을 가질 수 있습니다.

필요한 경우 읽기 전용 복제본을 자체 프로덕션 데이터베이스로 승격할 수 있습니다.

읽기 전용 복제본은 RDS를 기반으로 하는 DB의 6가지 유형 모두에 대해 지원됩니다.

각 읽기 전용 복제본에는 자체 DNS 엔드포인트가 있습니다.

읽기 전용 복제본을 사용하려면 자동 백업을 활성화해야 합니다.

다중 AZ가 켜져 있는 읽기 전용 복제본을 가지거나 완전히 별도의 리전에 읽기 전용 복제본을 둘 수 있습니다. 읽기 전용 복제본의 읽기 전용 복제본을 가질 수도 있지만 대기 시간이나 복제 지연에 주의하십시오. 읽기 전용 복제본에 대한 주의 사항은 소량의 복제 지연이 발생할 수 있다는 것입니다. 이는 기본 트랜잭션만큼 빠르게 업데이트되지 않기 때문에 최신 트랜잭션 중 일부가 누락될 수 있기 때문입니다. 응용 프로그램 디자이너는 약간 오래된 데이터를 허용하는 쿼리를 고려해야 합니다. 이러한 쿼리는 읽기 전용 복제본에서 실행되어야 하며 완전히 최신 데이터를 요구하는 쿼리는 기본 노드에서 실행되어야 합니다.

RDS 백업:

RDS의 경우 두 가지 종류의 백업이 있습니다.

- 자동 백업
- 데이터베이스 스냅샷

자동 백업보존 기간(1일에서 35일 사이) 내의 임의의 시점으로 데이터베이스를 복구할 수 있습니다. 자동 백업은 전체 일일 스냅샷을 생성하고 하루 종일 트랜잭션 로그도 저장합니다. DB 복구를 수행하면 RDS가 먼저 가장 최근의 일일 백업을 선택하고 해당 날짜의 관련 트랜잭션 로그를 적용합니다. 설정된 보존 기간 내에서 정확한 초 단위까지 시점 복구를 수행할 수 있는 기능을 제공합니다. 자동 백업은 기본적으로 활성화되어 있습니다. 백업 데이터는 실제 데이터베이스의 크기까지 자유롭게 저장됩니다(따라서 RDS에 저장되는 모든 GB에 대해 동일한 양이 DB의 GB 제한까지 S3에 자유롭게 저장됩니다). 백업은 정의된 창 내에서 수행되므로 데이터를 백업하기 위해 스토리지 I/O가 일시 중단되면 대기 시간이 늘어날 수 있습니다.

DB 스냅샷 은 관리자가 수동으로 수행합니다. 자동 백업과 다른 점은 원본 RDS 인스턴스가 종료된 후에도 유지된다는 점입니다. 자동 백업을 사용하면 S3의 백업 데이터가 RDS 엔진과 함께 완전히 지워집니다. 이것이 DB를 삭제할 때 DB의 최종 스냅샷을 찍을 것인지 묻는 이유입니다.

자동화된 백업 또는 DB 스냅샷을 통해 DB를 복원하려고 하면 도달하기 위해 자체 DB 엔드포인트가 있는 완전히 새로운 RDS 인스턴스가 프로비저닝됩니다.

RDS 보안:

IAM 데이터베이스 인증을 사용하여 DB 인스턴스에 인증할 수 있습니다. IAM 데이터베이스 인증은 MySQL 및 PostgreSQL에서 작동합니다. 이 인증 방법을 사용하면 DB 인스턴스에 연결할 때 암호를 사용할 필요가 없습니다. 대신 인증 토큰을 사용합니다.

인증 토큰은 Amazon RDS가 요청 시 생성하는 고유한 문자열입니다. 인증 토큰의 수명은 15분입니다. 인증은 IAM을 사용하여 외부에서 관리되기 때문에 데이터베이스에 사용자 자격 증명을 저장할 필요가 없습니다.

IAM 데이터베이스 인증은 다음과 같은 이점을 제공합니다.

- 데이터베이스를 오가는 네트워크 트래픽은 SSL(Secure Sockets Layer)을 사용하여 암호화됩니다.
- 각 DB 인스턴스에 대한 액세스를 개별적으로 관리하는 대신 IAM을 사용하여 데이터베이스 리소스에 대한 액세스를 중앙에서 관리할 수 있습니다.
- Amazon EC2에서 실행되는 애플리케이션의 경우 EC2 인스턴스와 관련된 프로필 자격 증명을 사용하여 보안을 강화하기 위해 암호 대신 데이터베이스에 액세스할 수 있습니다.

미사용 데이터 암호화는 RDS용 DB의 6가지 유형 모두에 대해 지원됩니다. 암호화는 AWS KMS 서비스를 사용하여 수행됩니다. RDS 인스턴스가 암호화되면 DB의 데이터는 물론 모든 백업(자동화 또는 스냅샷) 및 읽기 전용 복제본도 암호화됩니다.

데이터가 암호화되면 Amazon RDS는 성능에 미치는 영향을 최소화하면서 액세스 인증 및 데이터 암호 해독을 투명하게 처리합니다. 암호화를 사용하기 위해 데이터베이스 클라이언트 애플리케이션을 수정할 필요가 없습니다.

Amazon RDS 암호화는 현재 모든 데이터베이스 엔진 및 스토리지 유형에 사용할 수 있습니다. 그러나 기본 인스턴스 유형이 DB 암호화를 지원하는지 확인해야 합니다.

Amazon RDS DB 인스턴스를 생성할 때만 암호화를 활성화할 수 있으며 DB 인스턴스가 생성된 후에는 암호화를 비활성화할 수 없으며 암호화된 DB 인스턴스는 수정할 수 없습니다.

RDS 향상된 모니터링:

RDS는 향상된 모니터링 기능과 함께 제공됩니다. Amazon RDS는 DB 인스턴스가 실행되는 운영 체제(OS)에 대한 지표를 실시간으로 제공합니다. 콘솔을 사용하여 DB 인스턴스에 대한 지표를 보거나 선택한 모니터링 시스템에서 CloudWatch Logs의 Enhanced Monitoring JSON 출력을 사용할 수 있습니다.

기본적으로 Enhanced Monitoring 지표는 30일 동안 CloudWatch Logs에 저장됩니다. 지표가 CloudWatch Logs에 저장되는 시간을 수정하려면 CloudWatch 콘솔에서 RDS OS 지표 로그 그룹의 보존 기간을 변경하십시오.

CloudWatch와 Enhanced Monitoring 메트릭 간에는 주요 차이점이 있습니다. CloudWatch는 DB 인스턴스의 하이퍼바이저에서 CPU 사용률에 대한 지표를 수집하고 Enhanced Monitoring은 인스턴스의 에이전트에서 지표를 수집합니다. 결과적으로 하이퍼바이저 계층이 메트릭의 일부로 선택하고 해석할 수 있는 소량의 작업을 수행하기 때문에 측정값 간의 차이를 찾을 수 있습니다.

Aurora

Aurora 단순화:

Aurora는 기존 엔터프라이즈 데이터베이스의 성능 및 가용성과 오픈 소스 데이터베이스의 단순성 및 비용 효율성을 결합하는 것으로 알려진 AWS 주력 DB입니다. MySQL/PostgreSQL 호환 RDBMS로 경쟁사 대비 10분의 1 비용으로 상용 데이터베이스의 보안성, 가용성, 신뢰성을 제공합니다. MySQL 및 PostgreSQL에 대해 각각 5배 및 3배의 성능 배율로 인해 AWS 데이터베이스로서 훨씬 더 효과적입니다.

Aurora 키 세부 정보:

인프라 장애 시 Aurora는 자체 복제본으로 자동 장애 조치를 수행합니다.

Amazon Aurora에는 일반적으로 단일 인스턴스 대신 DB 인스턴스 클러스터가 포함됩니다. 각 연결은 특정 DB 인스턴스에서 처리됩니다. Aurora 클러스터에 연결할 때 지정한 호스트 이름과 포트는 엔드포인트라는 중간 핸들러를 가리킵니다. Aurora는 엔드포인트 메커니즘을 사용하여 이러한 연결을 추상화합니다. 따라서 일부 DB 인스턴스를 사용할 수 없을 때 모든 호스트 이름을 하드 코딩하거나 로드 밸런싱 및 연결 재라우팅을 위한 자체 로직을 작성할 필요가 없습니다.

기본적으로 모든 Aurora 데이터에 대해 총 6개의 복사본에 대해 최소 3개의 가용 영역에 2개의 복사본이 있습니다. 따라서 쓰기 가용성에 영향을 주지 않고 최대 2개의 데이터 복사본과 읽기 가용성에 영향을 주지 않고 최대 3개의 데이터 복사본의 잠재적 손실을 처리할 수 있습니다.

Aurora 스토리지는 자가 치유되며 데이터 블록과 디스크에 오류가 있는지 지속적으로 검사합니다. 오류가 발견되면 해당 오류가 자동으로 복구됩니다.

Aurora 복제는 Aurora의 복제본이 다중 AZ 구성의 일부인 대기 및 읽기 트래픽의 대상이 될 수

있다는 점에서 RDS 복제본과 다릅니다. RDS에서 다중 AZ 대기는 읽기 엔드포인트로 구성할 수 없으며 읽기 전용 복제본만 해당 기능을 제공할 수 있습니다.

Aurora 복제를 사용하면 최대 15개의 사본을 가질 수 있습니다. 복제된 복사본으로 다운스트림 MySQL 또는 PostgreSQL을 원하는 경우 5개 또는 1개만 가질 수 있습니다.

자동 장애 조치는 Aurora 읽기 복제에서만 가능합니다.

RDS 복제와 Aurora 복제의 차이점에 대한 자세한 내용은 다음을 참조하십시오.

Feature	Amazon Aurora Replicas	MySQL Replicas
Number of replicas	Up to 15	Up to 5
Replication type	Asynchronous (milliseconds)	Asynchronous (seconds)
Performance impact on primary	Low	High
Replica location	In-region	Cross-region
Act as failover target	Yes (no data loss)	Yes (potentially minutes of data loss)
Automated failover	Yes	No
Support for user-defined replication delay	No	Yes
Support for different data or schema vs. primary	No	Yes

자동 백업은 항상 Aurora 인스턴스에서 활성화되며 백업은 DB 성능에 영향을 미치지 않습니다. 성능에 영향을 미치지 않는 스냅샷을 찍을 수도 있습니다. 스냅샷은 AWS 계정 간에 공유할 수 있습니다.

RDS DB를 Aurora RD로 마이그레이션하는 일반적인 전술은 RDS MariaDB/MySQL DB의 읽기 전용 복제본을 Aurora DB로 생성하는 것입니다. 그런 다음 Aurora DB를 프로덕션 인스턴스로 승격하고 이전 MariaDB/MySQL DB를 삭제하기만 하면 됩니다.

Aurora는 10GB로 시작하여 스토리지 자동 확장을 통해 10GB당 64TB까지 확장됩니다. Aurora의 컴퓨팅 성능은 최대 32vCPU 및 244GB 메모리로 확장됩니다.

Aurora 서버리스:

Aurora Serverless는 Aurora의 MySQL/PostgreSQl 호환 버전을 위한 간단한 온디맨드 자동 확장 구성입니다. Aurora Serverless를 사용하면 인스턴스가 애플리케이션 사용량에 따라 자동으로 확장 또는 축소되고 시작되거나 시작됩니다. 이 서비스의 사용 사례는 드물고 간헐적이며 예측할 수 없는 워크로드입니다.

또한 호출당 비용을 지불하기 때문에 더 저렴하게 사용할 수 있습니다.

Aurora Serverless를 사용하면 데이터베이스 엔드포인트를 생성하고 선택적으로 원하는 데이터베이스 용량 범위를 지정한 다음 애플리케이션을 연결하기만 하면 됩니다.

데이터베이스 인스턴스 및 용량 관리의 복잡성을 제거합니다. 데이터베이스는 자동으로 시작, 종

료 및 응용 프로그램의 요구 사항에 맞게 확장됩니다. 클라이언트 연결을 중단하지 않고 필요에 따라 컴퓨팅 및 메모리 용량을 원활하게 확장합니다.

Aurora Cluster 엔드포인트:

클러스터 엔드포인트를 사용하여 사용 사례에 따라 적절한 인스턴스 또는 인스턴스 그룹에 각 연결을 매핑합니다.

Aurora DB 전체에서 다양한 역할 또는 작업과 연결된 클러스터 엔드포인트에 연결할 수 있습니다. 이는 서로 다른 인스턴스 또는 인스턴스 그룹이 서로 다른 기능을 수행하기 때문입니다.

예를 들어, DDL 문을 수행하기 위해 기본 인스턴스에 연결할 수 있습니다. 쿼리를 수행하려면 리더 엔드포인트에 연결하면 Aurora가 리더 엔드포인트 뒤에 있는 모든 Aurora 복제본 간에 로드 밸런싱을 자동으로 수행합니다. 진단 또는 조정을 위해 다른 끝점에 연결하여 세부 정보를 검사할 수 있습니다.

DB 인스턴스의 진입로는 장애 조치 후에도 동일하게 유지되므로 애플리케이션은 엔드포인트에 대한 수동 관리 개입 없이 데이터베이스 작업을 재개할 수 있습니다.

Aurora Reader 엔드포인트:

Aurora 리더 엔드포인트는 위의 클러스터 엔드포인트 개념의 하위 집합입니다. 쿼리와 같은 읽기 작업에 리더 끝점을 사용합니다. 읽기 전용 Aurora 복제본에서 이러한 명령문을 처리함으로써 이 엔드포인트는 기본 인스턴스의 오버헤드를 줄입니다.

읽기 전용 쿼리 트래픽을 처리하는 데 도움이 되는 리더 엔드포인트 때문에 최대 15개의 Aurora 읽기 전용 복제본이 있습니다.

또한 클러스터가 클러스터의 Aurora 복제본 수에 비례하여 동시 SELECT 쿼리를 처리할 수 있는 용량을 확장하는 데 도움이 됩니다. 각 Aurora DB 클러스터에는 하나의 리더 엔드포인트가 있습니다.

클러스터에 하나 이상의 Aurora 복제본이 포함된 경우 리더 엔드포인트는 Aurora 복제본 간에 각 연결 요청을 로드 밸런싱합니다. 이 경우 해당 세션에서 SELECT와 같은 읽기 전용 문만 수행할 수 있습니다. 클러스터에 기본 인스턴스만 있고 Aurora 복제본이 없는 경우 리더 엔드포인트는 기본 인스턴스에 직접 연결됩니다. 이 경우 끝점을 통해 쓰기 작업을 수행할 수 있습니다.

DynamoDB

단순화된 DynamoDB:

Amazon DynamoDB는 모든 규모에서 한 자리 밀리초 성능을 제공하는 키-값 및 문서 데이터베이스입니다. 완전 관리형, 다중 지역, 다중 마스터, 내구성 있는 비 SQL 데이터베이스입니다. 기본 제공 보안, 백업 및 복원, 인터넷 규모 애플리케이션을 위한 인메모리 캐싱이 함께 제공됩니다.

DynamoDB 키 세부 정보:

DynamoDB의 주요 구성 요소는 다음과 같습니다.

- 기본 테이블 역할을 하는 컬렉션
- SQL 데이터베이스의 행에 해당하는 문서
- 문서 또는 행 내의 필드인 키-값 쌍

비관계형 DB의 편리함은 각 행이 사용 사례에 따라 완전히 다르게 보일 수 있다는 것입니다. 획일화할 필요는 없습니다. 예를 들어 특정 항목에 대해 새 열이 필요한 경우 해당 열이 다른 항목에 대해 존재하는지 확인할 필요도 없습니다.

DynamoDB는 문서 및 키-값 기반 모델을 모두 지원합니다. 모바일, 웹, 게임, 광고 기술, IoT 등에 적합합니다.

DynamoDB는 SSD를 통해 저장되므로 매우 빠릅니다.

지리적으로 구별되는 3개의 데이터 센터에 분산되어 있습니다.

기본 일관성 모델은 최종적으로 일관된 읽기이지만 강력하게 일관된 읽기도 있습니다.

두 일관성 모델의 차이점은 1초 규칙입니다. 최종 일관성 읽기를 사용하면 일반적으로 모든 데이터 복사본에 1초 이내에 도달합니다. 짧은 시간 후에 반복된 읽기는 업데이트된 데이터를 반환해야 합니다. 그러나 업데이트된 데이터를 1초 이내에 읽어야 하고 이것이 보장되어야 하는 경우에는 강력하게 일관된 읽기가 가장 좋습니다.

스키마 또는 데이터 구조가 자주 변경되어야 하는 시나리오에 직면한 경우 새로운 유형의 데이터를 추가하거나 제거하는 비경직적이고 유연한 방법을 제공하는 데이터베이스를 선택해야 합니다. 이것은 관계형 데이터베이스와 비관계형(NoSQL) 데이터베이스 사이에서 선택하는 전형적인 예입니다. 이 시나리오에서는 DynamoDB를 선택합니다.

관계형 데이터베이스 시스템은 다음과 같은 이유로 확장되지 않습니다.

- 데이터를 정규화하고 디스크에 쓰는 데 여러 쿼리가 필요한 여러 테이블에 저장합니다.
- 일반적으로 ACID 호환 트랜잭션 시스템의 성능 비용이 발생합니다.
- 쿼리 결과의 필수 보기를 재조립하기 위해 값비싼 조인을 사용합니다.

높은 카디널리티는 DynamoDB I/O 성능에 좋습니다. 파티션 키 값이 더 뚜렷할수록 더 좋습니다.

전송된 요청이 분할된 공간에 분산되도록 합니다.

DynamoDB는 병렬 처리를 사용하여 예측 가능한 성능을 달성합니다. 각 파티션 또는 노드를 정의된 데이터 블록을 담당하는 각 파티션 또는 노드가 있는 고정 크기의 독립 DB 서버로 시각화할 수 있습니다. SQL 용어로 이 개념을 샤딩이라고 하지만 물론 DynamoDB는 SQL 기반 DB가 아닙니다. DynamoDB를 사용하면 데이터가 SSD(Solid State Drive)에 저장됩니다.

DynamoDB 가속기(DAX):

Amazon DynamoDB Accelerator(DAX)는 완전 관리형 고가용성 인메모리 캐시로, Amazon DynamoDB 응답 시간을 초당 수백만 건의 요청에서도 밀리초에서 마이크로초로 단축할 수 있습니다.

DAX를 사용하면 전례 없는 요청 볼륨이 발생하는 경우에도 애플리케이션이 빠르고 응답성이 유지됩니다. 튜닝이 필요하지 않습니다.

DAX를 사용하면 온디맨드로 10노드 클러스터로 확장하여 초당 수백만 건의 요청을 제공할 수 있습니다.

DAX는 캐시를 통해 쓰기를 통해 읽기 성능을 높이는 것 이상을 수행합니다. 이렇게 하면 쓰기 성능도 향상됩니다.

DynamoDB와 마찬가지로 DAX는 완전 관리형입니다. 더 이상 하드웨어 또는 소프트웨어 프로비저닝, 설정 및 구성, 소프트웨어 패치, 안정적인 분산 캐시 클러스터 운영 또는 확장 시 여러 인스턴스에 데이터 복제와 같은 관리 작업에 대해 걱정할 필요가 없습니다.

즉, 개발자가 캐싱 논리를 관리할 필요가 없습니다. DAX는 기존 DynamoDB API 호출과 완전히 호환됩니다.

DAX를 사용하면 여러 DynamoDB 테이블에 대해 하나의 DAX 클러스터를 프로비저닝하거나 단일 DynamoDB 테이블에 대해 여러 DAX 클러스터를 프로비저닝할 수 있으므로 유연성을 극대화할 수 있습니다.

DAX는 HA용으로 설계되었으므로 한 AZ에 장애가 발생하면 다른 AZ에 있는 복제본 중 하나로 장애 조치됩니다. 이 또한 자동으로 관리됩니다.

DynamoDB 스트림:

DynamoDB 스트림은 Amazon DynamoDB 테이블의 항목 변경 사항에 대한 정보의 순서화된 흐름입니다. 테이블에서 스트림을 활성화하면 DynamoDB는 테이블의 데이터 항목에 대한 모든 수정 사항에 대한 정보를 캡처합니다.

Amazon DynamoDB는 AWS Lambda와 통합되어 DynamoDB 스트림의 이벤트에 자동으로 응답하는 코드 조각인 트리거를 생성할 수 있습니다.

테이블의 항목이 수정된 직후 테이블의 스트림에 새 레코드가 나타납니다. AWS Lambda는 스트림

을 폴링하고 새 스트림 레코드를 감지하면 Lambda 함수를 동기적으로 호출합니다. Lambda 함수는 알림 보내기 또는 워크플로 시작과 같이 지정한 모든 작업을 수행할 수 있습니다.

트리거를 사용하면 DynamoDB 테이블의 데이터 수정에 반응하는 애플리케이션을 구축할 수 있습니다.

애플리케이션이 테이블의 항목을 생성, 업데이트 또는 삭제할 때마다 DynamoDB Streams는 수정된 항목의 기본 키 속성으로 스트림 레코드를 씁니다. 스트림 레코드에는 DynamoDB 테이블의 단일 항목에 대한 데이터 수정에 대한 정보가 포함됩니다. 스트림 레코드가 수정된 항목의 "이전" 및 "이후" 이미지와 같은 추가 정보를 캡처하도록 스트림을 구성할 수 있습니다.

DynamoDB 전역 테이블

Global Tables는 전 세계적으로 분산된 앱의 빠른 로컬 성능을 위한 다중 지역, 다중 마스터 복제 솔루션입니다.

Global Tables는 선택한 AWS 리전에서 Amazon DynamoDB 테이블을 자동으로 복제합니다.

DynamoDB 스트림을 기반으로 하며 데이터 복구 또는 고가용성을 위해 다중 지역 중복입니다. 애플리케이션 장애 조치는 애플리케이션의 DynamoDB 호출을 다른 AWS 리전으로 리디렉션하는 것만큼 간단합니다.

Global Tables는 지역 간 데이터 복제 및 업데이트 충돌 해결의 어려운 작업을 제거하여 애플리케이션의 비즈니스 로직에 집중할 수 있도록 합니다. 글로벌 테이블을 사용하기 위해 애플리케이션을 다시 작성할 필요가 없습니다.

전역 테이블의 복제 대기 시간은 일반적으로 1초 미만입니다.

Redshift

단순화된 Redshift:

Amazon Redshift는 클라우드에서 페타바이트 규모의 완전 관리형 데이터 웨어하우스 서비스입니다. Amazon Redshift 서비스는 데이터 웨어하우스를 설정, 운영 및 확장하는 모든 작업을 관리합니다. 이러한 작업에는 용량 프로비저닝, 클러스터 모니터링 및 백업, Amazon Redshift 엔진에 패치 및 업그레이드 적용이 포함됩니다.

Redshift 키 세부 정보:

Amazon Redshift 클러스터는 리더 노드와 하나 이상의 컴퓨팅 노드로 구성된 노드 집합입니다. 필요한 컴퓨팅 노드의 유형과 수는 데이터 크기, 실행할 쿼리 수, 필요한 쿼리 실행 성능에 따라 다릅니다.

Redshift는 비즈니스 인텔리전스에 사용되며 데이터에서 통찰력을 수집하기 위해 복잡한 쿼리를 수행하기 위해 매우 크고 복잡한 데이터 세트를 가져옵니다.

OLAP(온라인 분석 처리)의 사용 사례에 맞습니다. Redshift는 거의 무제한에 가까운 보고서 보기, 복잡한 분석 계산 및 예측 가능한 "가정" 시나리오(예산, 예측 등) 계획을 위한 기능을 포함하여 데이터 검색을 위한 강력한 기술입니다.

데이터 웨어하우징 요구 사항에 따라 작은 단일 노드 클러스터로 시작하여 요구 사항이 변경됨에 따라 더 큰 다중 노드 클러스터로 쉽게 확장할 수 있습니다. 서비스 중단 없이 클러스터에 컴퓨팅 노드를 추가하거나 제거할 수 있습니다.

클러스터를 1년 이상 계속 실행하려는 경우 1년 또는 3년 동안 컴퓨팅 노드를 예약하여 비용을 절약할 수 있습니다.

스냅샷은 클러스터의 특정 시점 백업입니다. 이러한 백업은 기본적으로 1일의 보존 기간으로 활성화됩니다. 최대 보존 기간은 35일입니다.

Redshift는 원하는 경우 스냅샷을 다른 지역에 비동기식으로 복제할 수도 있습니다.

고가용성 Redshift 클러스터에는 3개의 데이터 복사본이 필요합니다. 하나의 사본은 Redshift에 라이브 상태가 되고 나머지는 S3에 대기 상태가 됩니다.

Redshift는 다중 노드 클러스터에서 최대 128개의 컴퓨팅 노드를 가질 수 있습니다. 리더 노드는 항상 클라이언트 연결을 관리하고 실제 데이터를 저장하고 쿼리를 수행하는 컴퓨팅 노드에 쿼리를 릴레이합니다.

Redshift는 유사한 데이터가 포함된 데이터 저장소의 열 형식 압축을 사용하여 아키텍처의 많은 부분과 부분에도 불구하고 효율성을 달성할 수 있습니다. 또한 Redshift는 인덱스나 구체화된 뷰가 필요하지 않으므로 동일한 양의 정보를 포함하는 OLTP 데이터베이스에 비해 크기가 상대적으로 작을 수 있습니다. 마지막으로 Redshift 테이블에 데이터를 로드할 때 Redshift는 자동으로 데이터를 다운샘플링하고 가장 적절한 압축 방식을 선택합니다.

Redshift는 또한 다중 노드 클러스터의 모든 노드를 활용하기 위해 대규모 병렬 처리(MPP)와 함께 제공됩니다. 이는 데이터와 쿼리 로드를 모든 노드에 고르게 분산하여 수행됩니다. 이 때문에 수평 확장은 여전히 우수한 성능을 유지합니다.

Redshift는 SSL을 사용하여 전송 시 암호화되고 AES-256을 사용하여 저장 시 암호화됩니다. 기본적으로 Redshift는 모든 키를 관리하지만 AWS CloudHSM 또는 AWS KMS를 통해서도 관리할 수 있습니다.

Redshift는 다음에 대해 청구됩니다.

- 컴퓨팅 노드 시간(리더가 아닌 노드가 데이터를 쿼리하는 데 사용한 총 시간)
- 백업
- VPC 내 데이터 전송(외부는 아님)

Redshift는 다중 AZ가 아닙니다. 다중 AZ를 원할 경우 동일한 입력을 수집하는 별도의 클러스터를 가동해야 합니다. 중단 시 스냅샷을 새 AZ로 수동으로 복원할 수도 있습니다.

Amazon Redshift 클러스터를 프로비저닝하면 기본적으로 잠겨 있으므로 아무도 액세스할 수 없습니다. 다른 사용자에게 Amazon Redshift 클러스터에 대한 인바운드 액세스 권한을 부여하려면 클

러스터를 보안 그룹과 연결합니다.

Amazon Redshift는 클러스터를 삭제할 때까지 클러스터의 스토리지 용량과 동일한 스냅샷용 무료 스토리지를 제공합니다. 무료 스냅샷 스토리지 한도에 도달하면 추가 스토리지에 대해 정상 요율로 요금이 부과됩니다. 이 때문에 자동 스냅샷을 보관해야 하는 일수를 평가하고 그에 따라 보존 기간을 구성하고 더 이상 필요하지 않은 수동 스냅샷을 삭제해야 합니다.

자동 스냅샷 활성화 여부에 관계없이 원할 때마다 수동 스냅샷을 생성할 수 있습니다. Amazon Redshift는 수동 스냅샷을 자동으로 삭제하지 않습니다. 수동 스냅샷은 Redshift 클러스터를 삭제한 후에도 유지됩니다. 수동 스냅샷에는 스토리지 요금이 발생하므로 더 이상 필요하지 않은 경우 수동으로 삭제하는 것이 중요합니다.

Redshift 스펙트럼:

Amazon Redshift Spectrum은 로드 또는 ETL 없이 Amazon S3의 엑사바이트 규모의 비정형 데이터에 대한 쿼리를 실행하는 데 사용됩니다.

Redshift Spectrum 쿼리는 대규모 병렬 처리를 사용하여 대규모 데이터 세트에 대해 매우 빠르게 실행합니다. 처리의 대부분은 Redshift Spectrum 계층에서 발생하며 대부분의 데이터는 Amazon S3에 남아 있습니다.

Redshift Spectrum 쿼리는 다른 쿼리보다 클러스터의 처리 용량을 훨씬 적게 사용합니다.

Amazon S3의 클러스터와 데이터 파일은 동일한 AWS 리전에 있어야 합니다.

외부 S3 테이블은 읽기 전용입니다. 외부 테이블에서는 삽입, 업데이트 또는 삭제 작업을 수행할 수 없습니다.

Redshift 향상된 VPC 라우팅:

Amazon Redshift Enhanced VPC Routing을 사용하는 경우 Redshift는 Amazon VPC를 통해 클러스터와 데이터 리포지토리 간의 모든 트래픽(예: COPY 및 UNLOAD 트래픽)을 강제 실행합니다.

Enhanced VPC Routing이 활성화되지 않은 경우 Amazon Redshift는 AWS 네트워크 내의 다른 서비스에 대한 트래픽을 포함하여 인터넷을 통해 트래픽을 라우팅합니다.

향상된 VPC 라우팅을 사용하면 VPC 보안 그룹, 네트워크 ACL(액세스 제어 목록), VPC 엔드포인트, VPC 엔드포인트 정책, 인터넷 게이트웨이, DNS(Domain Name System) 서버와 같은 표준 VPC 기능을 사용할 수 있습니다.

ElastiCache

간소화된 ElastiCache:

ElastiCache 서비스를 사용하면 클라우드에서 인메모리 캐시를 쉽게 배포, 운영 및 확장할 수 있습니다. 처리량이 높고 대기 시간이 짧은 인메모리 데이터 저장소에서 데이터를 검색하여 기존 데이터베이스의 성능을 높이는 데 도움이 됩니다.

ElastiCache 키 세부 정보:

이 서비스는 상대적으로 멀리 떨어진 DB에만 의존하지 않고 로컬에서 정보를 받을 수 있도록 하여 웹 애플리케이션의 성능을 향상시키는 데 좋습니다.

Amazon ElastiCache는 밀리초 미만의 응답 시간이 필요한 가장 까다로운 애플리케이션을 위해 완전 관리형 Redis 및 Memcached를 제공합니다.

자주 변경되지 않고 자주 요청되는 데이터의 경우 데이터베이스에서 쿼리하는 것보다 해당 데이터를 캐시하는 것이 훨씬 합리적입니다.

DB 성능을 향상시키는 일반적인 구성에는 기본 DB의 읽기 전용 복제본 도입 및 스토리지 아키텍처에 캐싱 계층 삽입이 포함됩니다.

MemcacheD는 수평 확장 및 다중 스레드 성능을 갖춘 단순한 캐싱용이지만 캐싱 환경에 더 복잡한 것이 필요한 경우 Redis를 선택하십시오.

ElastiCache용 MemcacheD와 Redis 간의 추가 비교:

Requirement	Memcached	Redis
Simple Cache to offload DB	Yes	Yes
Ability to scale horizontally	Yes	Yes
Multi-threaded performance	Yes	No
Advanced data types	No	Yes
Ranking/Sorting data sets	No	Yes
Pub/Sub capabilities	No	Yes
Persistence	No	Yes
Multi-AZ	No	Yes
Backup & Restore Capabilities	No	Yes

ElastiCache를 사용하는 또 다른 장점은 쿼리 결과를 캐싱하여 데이터가 변경되지 않는 한 쿼리를 다시 실행할 필요 없이 DB 쿼리 가격을 한 번만 지불하면 된다는 것입니다.

Amazon ElastiCache는 변동하는 애플리케이션 요구 사항을 충족하기 위해 확장, 축소 및 확장할 수 있습니다. 쓰기 및 메모리 확장은 샤딩으로 지원됩니다. 복제본은 읽기 확장을 제공합니다.

Route53

Route53 단순화:

Amazon Route 53은 가용성과 확장성이 뛰어난 DNS(Domain Name System) 서비스입니다. Route 53을 사용하여 도메인 등록, DNS 라우팅 및 상태 확인의 세 가지 주요 기능을 조합하여 수행할 수 있습니다.

Route53 주요 세부 정보:

DNS는 전화번호부가 회사 이름을 전화번호와 매핑하는 방식과 유사하게 사람이 읽을 수 있는 도메인 이름을 인터넷 프로토콜 주소에 매핑하는 데 사용됩니다.

AWS에는 자체 도메인 등록 대행자가 있습니다.

도메인 이름을 구입할 때 모든 DNS 주소는 SOA(권한 시작) 레코드로 시작합니다. SOA 레코드는 소유권 이전을 시작한 서버 이름, 이제 도메인을 사용할 관리자, 사용 가능한 현재 메타데이터, 기본 시간(초) 또는 TTL에 대한 정보를 저장합니다.

NS 레코드 또는 이름 서버 레코드는 최상위 도메인 호스트(.org, .com, .uk 등)에서 트래픽을 콘텐츠 서버로 보내는 데 사용됩니다. 콘텐츠 DNS 서버에는 권한 있는 DNS 레코드가 포함되어 있습니다.

브라우저는 쿼리를 받을 때마다 최상위 도메인과 대화하고 인식할 수 없는 도메인 이름을 발견합니다.

- 브라우저는 도메인과 연결된 신뢰할 수 있는 DNS 레코드를 요청합니다.
- 최상위 도메인에는 NS 레코드가 포함되어 있기 때문에 TLD는 차례로 자체 SOA에 대해 이름 서버에 쿼리할 수 있습니다.
- SOA 내에는 요청된 정보가 있습니다.
- 이 정보가 수집되면 정보를 요청하는 원래 브라우저로 다시 반환됩니다.

요약하면 브라우저 -> TLD -> NS -> SOA -> DNS 레코드입니다. 올바른 DNS 레코드가 발견되면 파이프라인이 반전됩니다.

권한 있는 이름 서버는 일반적으로 DNS 등록 및 호스팅을 모두 제공하는 GoDaddy와 같은 DNS 호스팅 공급자 또는 도메인 등록 기관인 DNS 레코드 정보를 저장합니다.

Route53에는 수많은 DNS 레코드가 있습니다. 다음은 더 일반적인 몇 가지입니다.

- A 레코드 : DNS 레코드의 기본 유형입니다. A 레코드의 "A"는 "주소"를 나타냅니다. 이러한 레코드는 컴퓨터에서 도메인 이름을 IP 주소와 직접 연결하는 데 사용됩니다. IPv4 및 IPv6은 모두 IPv6 버전을 나타내는 "AAAA"로 지원됩니다. A: URL -> IPv4 및 AAAA: URL -> IPv6 .

- CName 레코드 : 정식 이름이라고도 합니다. 이러한 레코드는 한 도메인 이름을 다른 도메인 이름으로 확인하는 데 사용됩니다. 예를 들어, 웹사이트의 모바일 버전의 도메인은 별도의 IP 주소가 아니라 동일한 웹사이트의 브라우저 버전 도메인의 CName일 수 있습니다. 이렇게 하면 사이트를 방문하는 모바일 사용자가 모바일 버전을 받을 수 있습니다. CNAME: URL -> URL .

- Alias(별칭) 레코드 : 이 레코드는 도메인을 로드 밸런서, CDN 엔드포인트 및 S3 버킷과 같은 AWS 리소스에 매핑하는 데 사용됩니다. 별칭 레코드는 한 도메인을 다른 도메인에 매핑한다는 점에서 CName과 유사하게 작동합니다. 그러나 주요 차이점은 별칭 레코드를 도메인 이름이 아닌 서비스로 지정하면 필요한 경우 도메인 이름을 자유롭게 변경할 수 있고 매핑될 레코드에 대해 걱정할 필요가 없다는 것입니다. 별칭 레코드는 동적 기능을 제공합니다. 별칭: URL -> AWS 리소스 .

- PTR 레코드 : 이 레코드는 A 레코드의 반대입니다. PTR 레코드는 IP를 도메인에 매핑하고 IP 주소의 도메인 이름을 얻기 위한 방법으로 역 DNS 조회에 사용됩니다. PTR: IPv4 -> URL .

CName과 Alias 레코드의 또 다른 주요 차이점은 CName을 네이키드 도메인 이름(전체 DNS 구성의 apex 레코드/사용할 기본 레코드)에 사용할 수 없다는 것입니다. CName은 항상 다른 보조 레코드나 정점 레코드에 매핑할 수 있는 보조 레코드여야 합니다. 기본이 작동하려면 항상 별칭 또는 A 레코드 유형이어야 합니다.

Alias 레코드의 동적 특성으로 인해 대부분의 사용 사례에 권장되는 경우가 많으며 가능한 경우 사용해야 합니다.

TTL은 DNS 레코드가 확인 서버나 사용자 자신의 캐시에 캐시되어 IP를 도메인에 대한 최신 매핑을 검색할 수 있도록 하는 길이입니다. TTL(Time To Live)은 초 단위로 측정되며 TTL이 낮을수록 더 빠른 DNS 변경이 인터넷에 전파됩니다. 예를 들어, 대부분의 공급자는 48시간 동안 지속되는 TTL을 가지고 있습니다.

DNS 설정에 문제가 발생할 경우 상태 확인을 생성하여 간단한 알림을 보낼 수 있습니다.

또한 Route53 상태 확인은 인터넷을 통해 액세스할 수 있는 모든 AWS 엔드포인트에 사용할 수 있습니다. 따라서 AWS 엔드포인트의 상태를 모니터링하는 데 이상적인 옵션입니다.

Route53 라우팅 정책:

레코드를 생성할 때 Amazon Route 53이 DNS 쿼리에 응답하는 방식을 결정하는 라우팅 정책을 선택합니다. 사용 가능한 라우팅 정책은 다음과 같습니다.

- 단순 라우팅
- 가중치 라우팅
- 지연 기반 라우팅
- 장애 조치 라우팅
- 지리적 위치 라우팅
- 지리적 근접 라우팅
- 다중값 응답 라우팅

단순 라우팅 은 로드 균형을 조정하려는 경우 레코드 뒤에 하나 이상의 IP 주소가 있는 DNS의 단일 레코드만 필요할 때 사용됩니다. 단순 라우팅 정책에서 여러 값을 지정하는 경우 Route53은 사용 가능한 옵션에서 임의의 IP를 반환합니다.

가중치 기반 라우팅 은 할당된 가중치를 기반으로 트래픽을 분할하려는 경우에 사용됩니다. 예를 들어 트래픽의 80%를 한 AZ로 이동하고 나머지를 다른 AZ로 이동하려면 가중 라우팅을 사용하십시오. 이 정책은 기능 변경을 테스트하는 데 매우 유용하며 트래픽 분할 특성으로 인해 블루-그린 배포를 수행하는 수단으로 두 배가 될 수 있습니다. 가중치 기반 라우팅을 생성할 때 각 IP 주소에 대해 새 레코드를 지정해야 합니다. Simple Routing과 같이 다양한 IP를 하나의 레코드로 그룹화할 수 없습니다.

레이턴시 기반 라우팅 은 이름에서 알 수 있듯이 주어진 사용자의 가장 낮은 레이턴시를 기반으로 라우팅 설정을 기반으로 합니다. 지연 기반 라우팅을 사용하려면 트래픽을 수신하는 해당 EC2 또는 ELB 리소스와 동일한 리전에 지연 리소스 레코드 세트를 생성해야 합니다. Route53은 사이트에 대한 쿼리를 수신하면 사용자에게 가장 빠른 속도를 제공하는 레코드 세트를 선택합니다. Latency 기반 라우팅을 생성할 때 각 IP에 대해 새 레코드를 지정해야 합니다.

장애 조치 라우팅 은 능동-수동 장애 조치 설정을 구성하려는 경우에 사용됩니다. Route53은 필요할 때 장애 조치할 수 있도록 기본 상태를 모니터링합니다. 더 자세한 규칙을 원하는 경우 모든 엔드포인트를 모니터링하도록 상태 확인을 수동으로 설정할 수도 있습니다.

지리적 위치 라우팅 을 사용하면 사용자의 지리적 위치를 기반으로 트래픽을 보낼 위치를 선택할 수 있습니다.

지리적 근접 라우팅 을 사용하면 사용자 및 리소스의 지리적 위치를 기반으로 트래픽이 전송될 위치를 선택할 수 있습니다. 편향이라고 하는 지정된 가중치를 기반으로 트래픽을 더 많거나 적게 라우팅하도록 선택할 수 있습니다. 이러한 편향은 지리적 영역의 가용성을 확장하거나 축소하여 한 위치의 리소스에서 다른 위치의 리소스로 트래픽을 쉽게 이동할 수 있도록 합니다. 이 라우팅 방법을 사용하려면 Route53 트래픽 흐름을 활성화해야 합니다. 글로벌 트래픽을 제어하려면 지리적 근접 라우팅을 사용하십시오. 트래픽이 로컬 지역에 머물도록 하려면 지리적 위치 라우팅

을 사용하십시오.

다중값 라우팅 은 단순 라우팅과 거의 동일하지만 다중값 라우팅을 사용하면 각 레코드 세트에 상태 확인을 할 수 있습니다. 이렇게 하면 IP가 아닌 정상적인 IP만 무작위로 반환됩니다.

Elastic Load Balancers(ELB)

ELB 단순화:

Elastic Load Balancing은 Amazon EC2 인스턴스, Docker 컨테이너, IP 주소 및 Lambda 함수와 같은 여러 대상에 수신 애플리케이션 트래픽을 자동으로 분산합니다. 단일 가용 영역 또는 여러 가용 영역에서 애플리케이션 트래픽의 다양한 로드를 처리할 수 있습니다. Elastic Load Balancing은 애플리케이션 내결함성을 만드는 데 필요한 고가용성, 자동 조정 및 강력한 보안을 모두 제공하는 세 가지 유형의 로드 밸런서를 제공합니다.

ELB 주요 세부사항:

로드 밸런서는 인터넷 연결 또는 애플리케이션 내부일 수 있습니다.

도메인 트래픽을 ELB 로드 밸런서로 라우팅하려면 Amazon Route 53을 사용하여 로드 밸런서를 가리키는 별칭 레코드를 생성하십시오. 별칭 레코드는 CName보다 선호되지만 둘 다 작동할 수 있습니다.

ELB에는 사전 정의된 IPv4 주소가 없습니다. 대신 DNS로 해결해야 합니다. 로드 밸런서는 기본적으로 자체 IP를 갖지 않지만 네트워크 LB는 고성능을 위한 것이기 때문에 네트워크 로드 밸런서에 대한 고정 IP를 생성할 수 있습니다.

ELB 뒤의 인스턴스는 InService또는 로 보고됩니다 OutOfService. ELB 뒤에 있는 EC2 인스턴스가 상태 확인에 실패하면 ELB가 해당 인스턴스로의 트래픽 전송을 중지합니다.

로드 밸런서를 위한 이중 스택 구성은 IPv4 및 IPv6을 통한 로드 밸런싱을 의미합니다.

AWS에는 세 가지 유형의 LB가 있습니다.

- 애플리케이션 LB
- 네트워크 LB
- 클래식 LB.

애플리케이션 LB 는 HTTP(S) 트래픽에 가장 적합하며 레이어 7에서 로드 밸런싱을 수행합니다. 애플리케이션을 인식할 만큼 충분히 지능적이며 Application Load Balancer는 또한 경로 기반 라우팅, 호스트 기반 라우팅 및 컨테이너화된 애플리케이션 지원을 지원합니다. 예를 들어, 웹 브라우저의 언어를 프랑스어로 변경하면 응용 프로그램 LB는 사용하는 언어에 대한 세부 정보가 포함된 브라우저에서 수신하는 메타데이터를 볼 수 있습니다. 브라우징 경험을 최적화하기 위해 LB 뒤의 백엔드에 있는 프랑스어 서버로 라우팅합니다. 또한 특정 사례에 대해 스스로 설정한 규칙

에 따라 트래픽을 특정 서버로 이동하는 고급 요청 라우팅을 생성할 수 있습니다.

네트워크 LB 는 성능이 필요한 TCP 트래픽에 가장 적합하며 계층 4에서 부하를 분산합니다. 매우 짧은 대기 시간을 유지하면서 초당 수백만 개의 요청을 관리할 수 있습니다.

클래식 LB 는 레거시 ELB 제품이며 HTTP(S) 또는 TCP에서 균형을 유지하지만 둘 다에서는 균형을 유지하지 않습니다. 가장 오래된 LB이지만 고정 세션 및 X-Forwarded-For 헤더와 같은 기능을 여전히 지원합니다.

유연한 애플리케이션 관리 및 TLS 종료가 필요한 경우 Application Load Balancer를 사용해야 합니다. 애플리케이션에 극한의 성능과 고정 IP가 필요한 경우에는 Network Load Balancer를 사용해야 합니다. 애플리케이션이 EC2 Classic 네트워크 내에 구축된 경우 Classic Load Balancer를 사용해야 합니다.

ELB 뒤에 있는 웹사이트를 보기 위한 요청의 수명 주기:

- 브라우저는 DNS에서 로드 밸런서의 IP 주소를 요청합니다.
- DNS는 IP를 제공합니다.
- IP를 사용하여 브라우저는 로드 밸런서에서 HTML 페이지에 대한 HTTP 요청을 수행합니다.
- AWS 경계 디바이스는 요청을 LB로 전달하기 전에 확인하고 확인합니다.
- LB는 HTTP 요청을 전달할 활성 웹 서버를 찾습니다.
- 웹 서버는 요청된 HTML 파일을 반환합니다.
- 브라우저는 요청한 HTML 파일을 수신하고 화면에 그래픽 표현을 렌더링합니다.

로드 밸런서는 지역 서비스입니다. 다른 지역 간에 부하를 분산하지 않습니다. 운영하는 각 지역에서 새 ELB를 프로비저닝해야 합니다.

애플리케이션이 응답을 중지하면 로드 밸런서에 도달할 때 504 오류가 수신됩니다. 이는 애플리케이션에 문제가 있고 오류가 그 뒤에 있는 서비스에서 로드 밸런서로 버블링되었을 수 있음을 의미합니다. LB 자체에 문제가 있다는 의미는 아닙니다.

ELB 고급 기능:

IPv6 DNS 확인을 활성화하려면 ALIAS AAAA 레코드가 IPv4 레코드와 함께 로드 밸런서로 확인되도록 두 번째 DNS 리소스 레코드를 생성해야 합니다.

프록시 프로토콜을 통한 X-Forwarded-For 헤더는 단순히 로드 밸런서가 LB 뒤에 있는 서버의 정보에 대한 실제 요청과 함께 요청자의 IP 주소를 전달하는 아이디어입니다. 일반적으로 LB 뒤에 있는 서버는 트래픽을 보내는 IP가 로드 밸런서에 속하는 것으로만 봅니다. 그들은 무언가를 하도록 요청하는 컴퓨터(LB)에 대해서만 알고 있기 때문에 일반적으로 요청의 진정한 출처에 대해 전혀 모릅니다. 그러나 때로는 특정 사용 사례에 대해 원래 IP를 백엔드 서버로 라우팅하고 LB의 IP 주소를 무시하고 싶을 수 있습니다. X-Forwarded-For 헤더가 이를 가능하게 합니다.

고정 세션은 특정 사용자가 애플리케이션 또는 웹사이트에 머무는 동안 특정 인스턴스에 바인딩됩니다. 이것은 응용 프로그램과의 모든 상호 작용이 매번 동일한 호스트로 전달됨을 의미합니다. 애플리케이션이 작동하기 위해 로컬 디스크가 필요한 경우 사용자가 특정 인스턴스의 동일한 임시 스토리지에 일관된 액세스를 보장하므로 고정 세션이 좋습니다. 고정 세션의 단점은 부적절하게 수행될 경우 로드 밸런싱의 목적을 무력화할 수 있다는 것입니다. 모든 트래픽이 균등하게 분산되는 대신 가상적으로 동일한 인스턴스에 바인딩될 수 있습니다.

경로 패턴은 해당 사용자 요청 내에 설정된 URL 경로를 기반으로 요청을 전달하는 규칙이 있는 리스너를 생성합니다. 경로 기반 라우팅이라고 하는 이 방법을 사용하면 트래픽이 특히 여러 백엔드 서비스로 전달될 수 있습니다. 예를 들어 경로 패턴을 사용하면 일반 요청을 한 대상 그룹으로 라우팅하고 이미지 렌더링 요청을 다른 대상 그룹으로 라우팅할 수 있습니다. 따라서 URL " www.example.com/" 은 일반 콘텐츠에 사용되는 서버로 전달되고 " www.example.com/photos" 는 이미지를 렌더링하는 다른 서버로 전달됩니다.

ELB 교차 영역 로드 밸런싱:

교차 영역 로드 밸런싱은 단일 AZ 내에서가 아니라 AZ 전반에 걸쳐 균일한 배포를 보장합니다.

교차 영역 로드 밸런싱이 비활성화된 경우 각 로드 밸런서 노드는 해당 가용 영역의 등록된 인스턴스에만 요청을 고르게 분산합니다.

교차 영역 로드 밸런싱은 활성화된 각 가용 영역에서 동일한 수의 인스턴스를 유지 관리할 필요성을 줄이고 하나 이상의 인스턴스 손실을 처리하는 애플리케이션의 능력을 향상시킵니다.

그러나 더 높은 내결함성을 위해 활성화된 각 가용 영역에서 거의 동일한 수의 인스턴스를 유지하는 것이 좋습니다.

클라이언트가 DNS 조회를 캐시하는 환경의 경우 수신 요청이 가용 영역 중 하나를 선호할 수 있습니다. 교차 영역 로드 밸런싱을 사용하면 요청 로드의 불균형이 대신 해당 지역의 사용 가능한 모든 인스턴스에 분산됩니다.

ELB 보안:

ELB는 SSL/TLS 및 HTTPS 종료를 지원합니다. 해독은 리소스와 CPU를 많이 사용하므로 로드 밸런서에서 종료하는 것이 좋습니다. 로드 밸런서에 복호화 부담을 가하면 EC2 인스턴스가 애플리케이션 작업에 처리 능력을 사용할 수 있으므로 전반적인 성능을 개선하는 데 도움이 됩니다.

Elastic Load Balancer(CloudFront와 함께)는 Perfect Forward Secrecy를 지원합니다. 이는 고유한 임의 세션 키를 사용하여 전송 중인 암호화된 데이터의 도청에 대한 추가 보호 기능을 제공하는 기능입니다. 이는 암호화 시스템의 사용 중인 부분이 정보를 암호화 및 해독하는 데 사용하는 키를 자동으로 자주 변경하도록 함으로써 수행됩니다. 따라서 이 최신 키가 손상되면 사용자의 최근 데이터 중 일부만 노출됩니다.

Classic Load Balancer는 SNI(서버 이름 표시)를 지원하지 않습니다. SNI를 사용하면 서버(이 경

우 LB)가 단일 IP 주소(이 경우 별칭 레코드 또는 CName 레코드)에서 여러 사이트에 대한 여러 TLS 인증서를 안전하게 호스팅할 수 있습니다. SNI를 허용하려면 대신 Application Load Balancer를 사용하거나 CloudFront 웹 배포와 함께 사용해야 합니다.

Auto Scaling

Auto Scaling 간소화:

AWS Auto Scaling을 사용하면 다양한 리소스 그룹이 수요 변화에 대응하는 방식을 자동화하는 조정 계획을 구축할 수 있습니다. 가용성, 비용 또는 이 둘의 균형을 최적화할 수 있습니다. AWS Auto Scaling은 모든 조정 정책을 자동으로 생성하고 기본 설정에 따라 대상을 설정합니다.

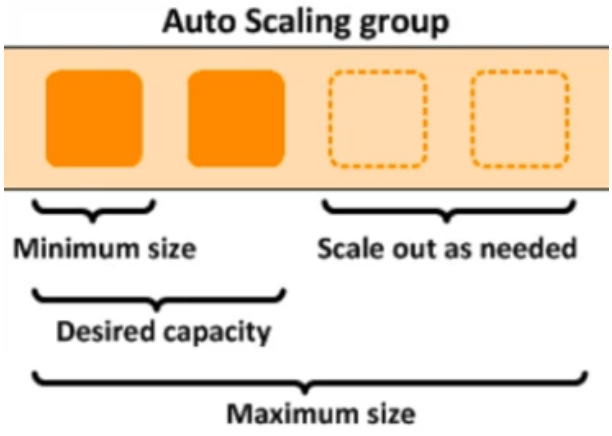
Auto Scaling 주요 세부 정보:

Auto Scaling은 클라우드의 규모의 경제가 가져다주는 가장 큰 이점이므로, Scaling에 대한 요구 사항이 있으면 자동으로 Auto Scaling 서비스를 사용하는 것을 생각하십시오.

Auto Scaling에는 세 가지 구성 요소가 있습니다.

- 그룹 : 논리적 구성 요소입니다. EC2 인스턴스의 웹 서버 그룹, RDS 인스턴스의 데이터베이스 그룹 등
- 구성 템플릿 : 그룹은 템플릿을 사용하여 확장 요구 사항에 더 잘 부합하도록 새 인스턴스를 구성하고 시작합니다. 사용할 AMI, 인스턴스 유형, 보안 그룹, 인스턴스와 연결할 차단 장치 등과 같은 새 인스턴스에 대한 정보를 지정할 수 있습니다.
- Scaling Options : Scaling Options는 Auto Scaling 그룹을 확장할 수 있는 여러 가지 방법을 제공합니다. 지정된 조건의 발생 또는 일정에 따라 조정 트리거를 기반으로 할 수 있습니다.

다음 이미지는 Auto scaling 그룹의 상태를 강조 표시합니다. 주황색 사각형은 활성 인스턴스를 나타냅니다. 점선 사각형은 필요할 때마다 회전할 수 있고 실행할 수 있는 잠재적인 인스턴스를 나타냅니다. 최소 개수, 최대 개수 및 원하는 인스턴스 용량은 모두 완전히 구성할 수 있습니다.



Auto Scaling을 사용하면 애플리케이션에서 다음과 같은 이점을 얻을 수 있습니다.

- 더 나은 내결함성 : Auto Scaling은 인스턴스가 비정상일 때 감지하고, 종료하고, 교체할 인스턴스를 시작할 수 있습니다. 여러 가용 영역을 사용하도록 Auto Scaling을 구성할 수도 있습니다. 한 가용 영역을 사용할 수 없게 되면 Auto Scaling은 이를 보완하기 위해 다른 가용 영역에서 인스턴스를 시작할 수 있습니다.
- 가용성 향상: Auto Scaling을 사용하면 애플리케이션이 현재 트래픽 수요를 처리할 수 있는 적절한 용량을 항상 확보할 수 있습니다.

실제로 인스턴스 그룹을 확장할 때 Auto Scaling 서비스는 유연하며 다양한 방법으로 수행할 수 있습니다.

- Auto Scaling은 인스턴스에 대한 수요에 따라 확장할 수 있습니다. 이 옵션은 특정 임계값에 도달하면 조정을 트리거하도록 지정하여 조정 프로세스를 자동화합니다. 이것은 Auto Scaling의 가장 인기 있는 구현입니다.
- Auto Scaling은 항상 현재 인스턴스 수를 보장할 수 있습니다. 이 옵션은 장애가 발생하더라도 실행하려는 서버의 수를 항상 유지합니다.
- Auto Scaling은 수동 개입으로만 확장할 수 있습니다. 모든 크기 조정을 직접 제어하려면 이 옵션이 적합합니다.
- Auto Scaling은 일정에 따라 확장할 수 있습니다. 트래픽 급증을 안정적으로 예측할 수 있다면 이 옵션이 적합합니다.
- 예측적 스케일링을 기반으로 하는 Auto Scaling. 이 옵션을 사용하면 AWS AI/ML이 환경에 대해 더 많이 학습하여 성능 향상과 비용 절감을 위한 최적의 확장 시간을 예측할 수 있습니다.

현재 실행 중인 인스턴스를 유지 관리할 때 Auto Scaling은 실행 중인 인스턴스에 대해 가끔 상태 확인을 수행하여 모두 정상인지 확인합니다. 서비스가 인스턴스가 비정상임을 감지하면 해당 인스턴스를 종료한 다음 새 인스턴스를 온라인으로 가져옵니다.

Auto Scaling용 HA를 설계할 때 가능한 여러 AZ와 여러 리전을 사용하십시오.

Auto Scaling을 사용하면 Auto Scaling 그룹에서 하나 이상의 Auto Scaling 프로세스를 일시 중단했다가 다시 시작할 수 있습니다. 이는 변경 시 Auto Scaling 프로세스를 트리거하지 않고 애플리케이션의 문제를 조사하려는 경우에 매우 유용할 수 있습니다.

여러 Auto Scaling 그룹으로 시작 구성을 지정할 수 있습니다. 그러나 Auto Scaling 그룹에 대해 한 번에 하나의 시작 구성만 지정할 수 있습니다.

시작 구성을 만든 후에는 수정할 수 없습니다. Auto Scaling 그룹의 시작 구성을 변경하려면 새 시작 구성을 생성하고 이 새 시작 구성을 상속하도록 Auto Scaling 그룹을 업데이트해야 합니다.

AWS Trusted Advisor: AWS 모범사례에 따라 리소스를 프로비저닝하는데
도움이 되도록 실시간 지침 제공 /AWS 인프라 최적화, 보안↑
비용↓, 성능↑ 모니터링

Auto Scaling 기본 종료 정책:

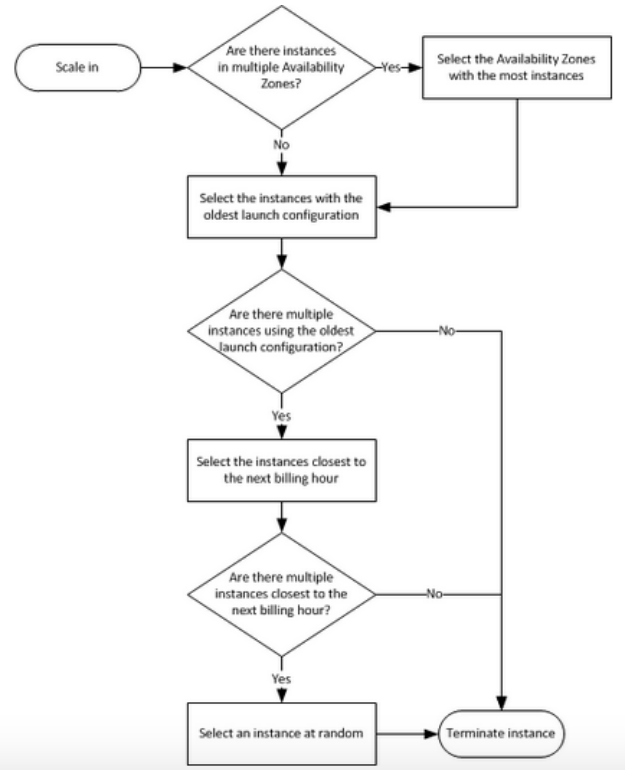
Auto Scaling Group에 대한 기본 종료 정책은 중지된 인스턴스를 자동으로 종료하는 것이므로 달리 구성하지 않는 한 인스턴스를 중지하면 원하는지 여부에 관계없이 인스턴스가 종료됩니다. 새 인스턴스가 그 자리에 스핀업됩니다.

기본 종료 정책은 일부 서버에서 중요한 시스템이나 응용 프로그램을 실행하는 경우 사용자가 알려주는 인스턴스를 절약합니다. 이러한 중요한 서버는 요구 사항에 불필요한 것으로 간주되는 인스턴스의 삭제 프로세스인 "스케일 인"으로부터 보호됩니다.

기본 종료 정책은 네트워크 아키텍처가 가용 영역을 균일하게 확장할 수 있도록 설계되었습니다. 기본 종료 정책에 따라 Auto Scaling 그룹의 동작은 다음과 같습니다.

- 여러 가용 영역에 인스턴스가 있는 경우 인스턴스가 가장 많은 가용 영역에서 인스턴스를 종료합니다. 최대 인스턴스 수가 동일한 두 개 이상의 가용 영역이 있는 경우 인스턴스가 가장 오래된 시작 구성을 사용하는 가용 영역을 선택합니다.
- 그런 다음 선택한 가용 영역에서 가장 오래된 시작 구성을 사용하는 보호되지 않은 인스턴스를 결정합니다. 그러한 인스턴스가 하나 있으면 종료됩니다.
- 종료할 인스턴스가 여러 개인 경우 다음 청구 시간에 가장 가까운 비보호 인스턴스를 결정합니다. (이렇게 하면 EC2 인스턴스의 사용을 극대화하고 Amazon EC2 사용 비용을 관리하는 데 도움이 됩니다.) 이 기준과 일치하는 일부 인스턴스가 있는 경우 해당 인스턴스가 종료됩니다.

이 순서도는 기본 Auto Scaling 정책이 삭제할 인스턴스를 결정하는 방법을 보다 명확하게 제공할 수 있습니다.



Auto Scaling 휴지 기간:

휴지 기간은 이전 조정 활동이 적용되기 전에 추가 인스턴스를 시작하거나 종료하지 않도록 하는 Auto Scaling 그룹에 대해 구성 가능한 설정입니다.

Auto Scaling Group은 정책을 사용하여 확장한 후 필요한 경우 추가 확장 활동을 재개하기 전에 휴지 기간이 완료될 때까지 기다립니다.

기본 대기 시간은 300초이지만 수정할 수 있습니다.

Virtual Private Cloud(VPC)

단순화된 VPC:

VPC를 사용하면 정의한 가상 네트워크 내에서 서비스와 시스템을 시작할 수 있는 논리적으로 격리된 AWS 클라우드 섹션을 프로비저닝할 수 있습니다. VPC는 공개 대상과 공개 대상이 아닌 AWS 리소스를 선택할 수 있어 보안에 대해 훨씬 더 세분화된 제어를 제공합니다.

VPC 키 세부 정보:

VPC를 클라우드의 자체 가상 데이터 센터로 생각할 수 있습니다. 자신의 네트워크를 완전히 제어할 수 있습니다. IP 범위, 하위 네트워크(서브넷) 생성, 라우팅 테이블 구성 및 사용된 네트워크 게이트웨이를 포함합니다.

그런 다음 선택한 서브넷에서 EC2 인스턴스를 시작하고, 인스턴스에 사용할 수 있는 IP를 선택하고, 보안 그룹을 할당하고, 서브넷 자체에 대한 NACL(네트워크 액세스 제어 목록)을 생성하여 추가 보호를 수행할 수 있습니다.

이 사용자 지정을 통해 인프라 설정을 지정하고 개인화할 수 있는 훨씬 더 많은 제어 권한을 얻을 수 있습니다. 예를 들어 웹 서버가 HTTP 트래픽을 수신할 하나의 공개 서브넷을 갖고 인터넷 액세스가 금지된 데이터베이스 서버에 대해 다른 비공개 서브넷을 가질 수 있습니다.

서브넷을 사용하여 호스트 수가 많은 네트워크를 효율적으로 활용합니다.

VPC는 설계상 심층 방어와 함께 제공됩니다. 하위 네트워크(NACL)에서 개별 서버(보안 그룹), 더 나아가 애플리케이션 자체(보안 코딩 관행)에 이르기까지 악의적인 사용자 및 프로그램에 대해 여러 수준의 보호를 설정할 수 있습니다.

AWS 환경의 기본 VPC는 모든 서브넷이 인터넷으로 나가는 경로를 갖도록 허용하므로 기본 VPC의 모든 서브넷이 인터넷에 액세스할 수 있습니다. 기본 설정을 사용하면 인스턴스를 즉시 배포할 수 있으며 각 EC2 인스턴스에는 퍼블릭 및 프라이빗 IP 주소가 모두 있습니다.

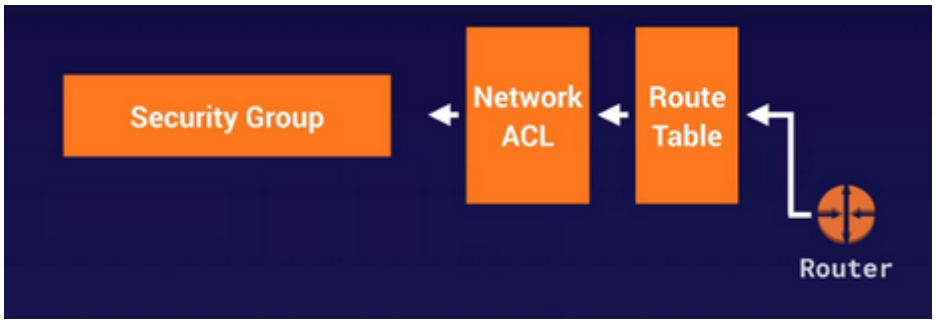
리전당 하나의 기본 VPC가 있습니다. 그러나 원하는 만큼 사용자 지정 VPC를 가질 수 있으며 모두 기본적으로 비공개입니다.

사용자 지정 VPC를 생성할 때 기본적으로 새 서브넷이 생성되지 않습니다. 별도로 생성해야 합니다. 인터넷 게이트웨이도 마찬가지입니다. VPC가 인터넷에 액세스할 수 있도록 하려면 전 세계에서 네트워크에 공개적으로 연결할 수 있도록 게이트웨이도 생성해야 합니다.

이 때문에 IGW를 만들 때 처음에는 분리된 상태가 됩니다. 사용자 지정 VPC에 수동으로 할당해야 합니다.

그러나 사용자 지정 VPC를 생성하면 기본적으로 다음이 생성됩니다.

- 라우팅 테이블
- NACL
- 보안 그룹



아직 알려지지 않은 경우를 위해 더 자세히 설명할 이러한 구성 요소는 실제로 데이터가 인스턴스에 도달하는 방법에 대한 트래픽 흐름에 해당합니다. 트래픽이 VPC 외부에서 발생하든 VPC 내부에서 발생하든 원하는 목적지가 어디인지 알기 위해서는 먼저 라우터를 통해 라우팅 테이블을 통과해야 합니다. 일단 그것이 알려지면 트래픽은 NACL에 설명된 대로 서브넷 수준 보안을 통과합니다. NACL이 트래픽을 유효한 것으로 간주하면 트래픽은 보안 그룹에서 설명한 대로 인스턴스 수준 보안으로 전달됩니다. 이 시점에서 트래픽이 삭제되지 않은 경우에만 의도한 인스턴스에 도달합니다.

VPC 마법사는 사용자 지정 VPC를 생성하는 데 유용한 자동화된 도구입니다.

네트워크가 물리적 수준에서 배타적이도록 전용 하드웨어에 VPC를 둘 수 있지만 이 옵션은 매우 비쌉니다. 다행히 VPC가 전용 호스팅에 있는 경우 언제든지 기본 호스팅으로 다시 변경할 수 있습니다. 이는 AWS CLI, SDK 또는 API를 통해 수행할 수 있습니다. 그러나 전용 하드웨어의 기존 호스트는 먼저 stopped상태에 있어야 합니다.

VPC를 생성할 때 IPv4 CIDR 블록을 할당해야 합니다. 이 CIDR 블록은 인스턴스를 생성할 때 인스턴스에 상속되는 프라이빗 IPv4 주소 범위입니다.

기본 VPC의 IP 범위는 항상 /16 입니다.

서브넷에 대한 IP 범위를 생성할 때 /16 CIDR 블록은 사용할 수 있는 가장 큰 IP 범위입니다. 서브넷에는 속한 VPC와 동일한 수의 IP 또는 더 적은 수의 IP가 있어야 하기 때문입니다. /28 CIDR 블록은 서브넷에 사용할 수 있는 가장 작은 IP 범위입니다 .

일반적으로 CIDR에서 /32 는 단일 IP 주소를 나타내고 /0 은 전체 네트워크를 나타냅니다. CIDR

에서 위로 올라갈수록 IP 범위가 더 좁아집니다.

IP에 대한 위의 정보는 공용 및 사설 IP 주소에 관한 것입니다.

사설 IP 주소는 인터넷을 통해 연결할 수 없으며 대신 VPC의 인스턴스 간 통신에 사용됩니다. VPC에서 인스턴스를 시작하면 서브넷의 IPv4 주소 범위에 있는 프라이빗 IP 주소가 인스턴스의 기본 네트워크 인터페이스(eth0)에 할당됩니다.

즉, VPC 내의 모든 인스턴스에는 프라이빗 IP가 있지만 외부 세계와 통신하도록 선택된 인스턴스에만 퍼블릭 IP가 있습니다.

인터넷 게이트웨이를 통해 퍼블릭 액세스 권한이 있는 서브넷으로 인스턴스를 시작하면 퍼블릭 IP 주소와 프라이빗 IP 주소가 모두 생성됩니다. 퍼블릭 IP 주소는 대신 인스턴스에 대해 생성된 기본 네트워크 인터페이스(eth0)에 할당됩니다. 외부적으로는 NAT(Network Address Translation)를 통해 공인 IP 주소를 사설 IP 주소로 매핑합니다.

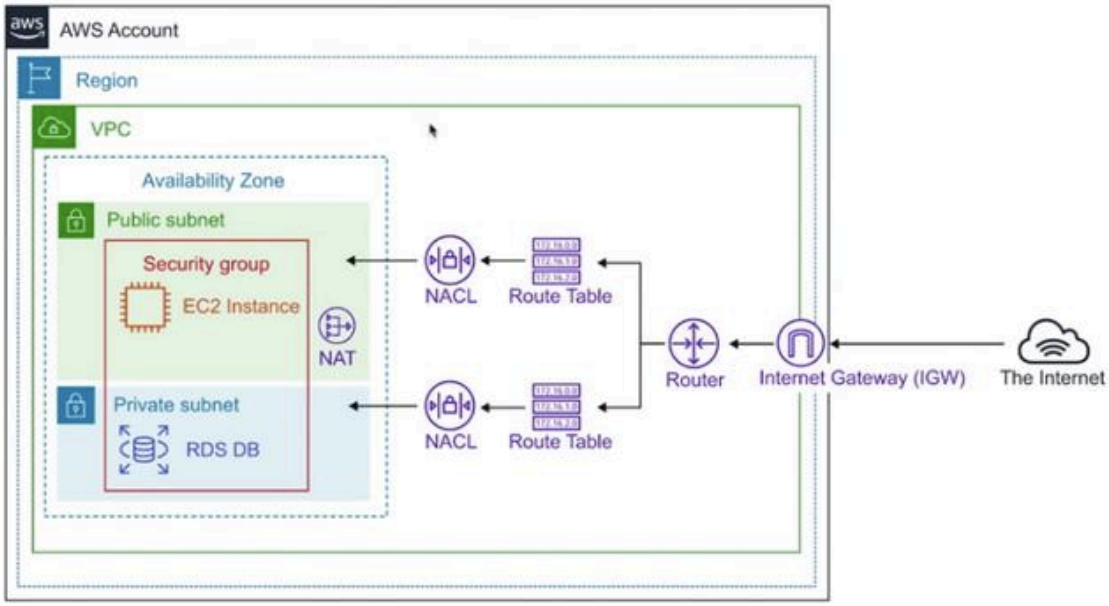
선택적으로 IPv6 CIDR 블록을 VPC 및 서브넷과 연결하고 해당 블록의 IPv6 주소를 VPC의 리소스에 할당할 수 있습니다.

VPC는 리전별로 다르며 리전당 최대 5개의 VPC를 가질 수 있습니다.

기본적으로 AWS는 애플리케이션이 있는 리전의 각 AZ에 하나의 서브넷을 갖도록 구성됩니다.

이상적이고 안전한 VPC 아키텍처에서는 퍼블릭 서브넷에서 웹 서버 또는 탄력적 로드 밸런서를 시작하고 프라이빗 서브넷에서 데이터베이스 서버를 시작합니다.

다음은 일반적인 VPC 설정 뒤에 있는 가상 애플리케이션의 예입니다.



보안 그룹은 서브넷에 걸쳐 있을 수 있지만 VPC에 걸쳐 있지는 않습니다. ICMP는 한 보안 그룹의 인스턴스가 다른 보안 그룹의 다른 인스턴스를 ping할 수 있도록 합니다. IPv4 및 IPv6과 호환됩니다.

VPC 서브넷:

네트워크에 논리적으로 그룹화된 세분화 없이 많은 수의 호스트가 있는 경우 많은 호스트를 관리하는 것은 지루한 작업이 될 수 있습니다. 따라서 서브넷을 사용하여 네트워크를 분할하여 관리가 더 쉬워집니다.

서브넷을 생성할 때 배치할 VPC를 지정해야 합니다. 서브넷에 IPv4 및 IPv6 범위를 모두 할당할 수 있습니다.

서브넷의 주요 이점:

- 트래픽 흐름을 개선하여 전체 네트워크의 속도와 성능을 향상시킵니다. IGW(Internet Gateway)는 패킷을 수신하여 5개의 서브넷 중 어느 서브넷으로 패킷을 전달해야 하는지 확인하는 것이 100개의 인스턴스를 개별적으로 확인하는 것보다 훨씬 빠릅니다. 그리고 패킷의 대상이 패킷이 시작된 서브넷 내에 있는 경우 트래픽은 서브넷 내부에 유지되고 나머지 VPC를 복잡하게 만들지 않습니다.
- 서브넷은 엔티티를 내부에 배치하는 논리적 그룹의 기능을 합니다. 모든 개별 인스턴스가 아닌 그룹으로 유사한 리소스를 훨씬 쉽게 구성할 수 있습니다.

Amazon은 서브넷 내에서 항상 5개의 IP 주소를 예약합니다. 각 서브넷 CIDR 블록의 처음 4개 IP 주소와 마지막 IP 주소는 항상 사용할 수 없습니다.

네트워크 액세스 제어 목록:

네트워크 액세스 제어 목록(또는 NACL)은 보안 그룹과 비슷하지만 인스턴스가 아닌 서브넷용입니다. 보안 그룹과 NACL의 주요 차이점은 보안 그룹이 상태 저장 이라는 것입니다. 즉, 해당 규칙에 대해 트래픽이 인바운드 또는 아웃바운드인지에 따라 다를 수 있는 허용 및 거부 규칙을 모두 수행할 수 있습니다.

다음 표는 NACL과 서브넷의 차이점을 강조합니다.

NACL	Security Group
Operates at the subnet level	Operates at the instance level
Supports allow rules and deny rules	Supports allow rules only
Is stateless: Return traffic must be explicitly allowed by rules	Is stateful: Return traffic is automatically allowed, regardless of any rules
We process rules in order, starting with the lowest numbered rule, when deciding whether to allow traffic	We evaluate all rules before deciding whether to allow traffic
Automatically applies to all instances in the subnets that it's associated with (therefore, it provides an additional layer of defense if the security group rules are too permissive)	Applies to an instance only if someone specifies the security group when launching the instance, or associates the security group with the instance later on

NACL은 상태 비저장이므로 수신 및 송신이 원활하게 흐를 수 있도록 아웃바운드 규칙이 인바운드 규칙과 함께 존재하는지 확인해야 합니다.

새 VPC와 함께 제공되는 기본 NACL에는 모든 인바운드 및 아웃바운드를 허용하는 기본 규칙이 있습니다. 즉, 존재하지만 모든 트래픽이 자유롭게 통과하므로 아무 작업도 수행하지 않습니다.

그러나 새 NACL을 생성하면(VPC와 함께 제공되는 기본값을 사용하는 대신) 기본 규칙은 모든 인바운드 및 아웃바운드를 거부합니다.

새 NACL을 만드는 경우 NACL의 규칙 집합을 상속할 수 있도록 원하는 서브넷을 수동으로 연결해야 합니다. 서브넷을 NACL에 명시적으로 할당하지 않으면 AWS가 서브넷을 기본 NACL과 연결합니다.

NACL은 보안 그룹보다 먼저 평가되며 보안 그룹이 아닌 NACL로 악성 IP를 차단합니다.

서브넷은 한 번에 하나의 NACL에 의해 나열된 규칙만 따를 수 있습니다. 그러나 NACL은 서브넷 수에 관계없이 규칙을 설명할 수 있습니다. 규칙은 즉시 적용됩니다. 여러 서브넷 연결X

네트워크 ACL 규칙은 가장 낮은 것부터 가장 높은 것까지 규칙 번호로 평가되며 일치하는 허용/거부 규칙이 발견되면 즉시 실행됩니다. 이 때문에 규칙 번호와 함께 순서가 중요합니다.

목록의 규칙 번호가 낮을수록 해당 규칙의 우선 순위가 높아집니다. 그에 따라 규칙을 나열하십시오.

NACL과 함께 NAT 게이트웨이를 사용하는 경우 NACL 규칙 내에서 NAT 게이트웨이 임시 포트 범위의 가용성을 확인해야 합니다. NAT 게이트웨이 트래픽은 연결 기간 동안 범위의 모든 포트에 나타날 수 있으므로 가능한 모든 포트가 고려되고 열려 있는지 확인해야 합니다.

NACL은 프라이빗 서브넷의 EC2 인스턴스가 VPC 엔드포인트를 포함한 모든 서비스와 통신하는 방식에 약간의 영향을 미칠 수 있습니다.

NAT 인스턴스 VS NAT 게이트웨이:

인터넷 게이트웨이를 VPC에 연결하면 공용 IP가 있는 인스턴스가 인터넷에 직접 액세스할 수 있습니다. NAT도 유사한 작업을 수행하지만 공용 IP가 없는 경우입니다. 이는 개인 인스턴스가 인터넷에 액세스하기 전에 먼저 자신의 개인 IP를 NAT의 공용 IP로 마스킹할 수 있도록 하는 중간 단계의 역할을 합니다.

개인 인스턴스가 인터넷에 액세스하여 정상적인 소프트웨어 업데이트를 받을 수 있기를 원할 것입니다. NAT는 인터넷에서 연결을 시작하는 것을 방지합니다.

NAT 인스턴스 는 인터넷에 안전하게 액세스할 수 있는 수단을 프라이빗 서브넷에 제공하는 기능을 수행하는 개별 EC2 인스턴스입니다.

개별 인스턴스이기 때문에 고가용성은 기본 제공 기능이 아니며 VPC의 초크 포인트가 될 수 있습니다. 내결함성이 없으며 단일 실패 지점 역할을 합니다. 병목 현상을 방지하기 위해 자동 크기 조정 그룹, 장애 조치를 자동화하는 스크립트 등을 사용할 수 있지만 확장 가능한 솔루션의 대안으로 NAT 게이트웨이를 사용하는 것이 훨씬 좋습니다.

NAT 게이트웨이 는 기본적으로 HA를 달성하기 위해 가용 영역 내에서 함께 연결된 여러 인스턴스로 구성된 관리형 서비스입니다.

추가 HA 및 영역 독립 아키텍처를 달성하려면 각 가용 영역에 대해 NAT 게이트웨이를 생성하고 리소스가 해당 가용 영역에서 NAT 게이트웨이를 사용하도록 라우팅을 구성하십시오.

NAT 인스턴스는 더 이상 사용되지 않지만 여전히 사용할 수 있습니다. NAT 게이트웨이는 네트워크 주소 변환을 달성하기 위해 선호되는 수단입니다.

서비스는 AWS에서 관리하므로 NAT 게이트웨이를 패치할 필요가 없습니다. NAT 인스턴스는 개별 EC2 인스턴스이기 때문에 패치해야 합니다.

통신은 항상 프라이빗 인스턴스에서 시작되어야 하므로 프라이빗 서브넷에서 NAT 게이트웨이로 트래픽을 라우팅하는 라우팅 규칙이 필요합니다.

NAT 인스턴스/게이트웨이는 퍼블릭 서브넷이 인터넷에 액세스할 수 있도록 구성된 서브넷이므로 퍼블릭 서브넷에 있어야 합니다.

NAT 인스턴스를 생성할 때 EC2 인스턴스에는 기본적으로 소스/대상 확인이 있다는 점을 기억하는 것이 중요합니다. 이러한 검사는 트래픽이 발생하는 모든 트래픽이 인스턴스에 의해 생성되거나 해당 트래픽의 의도된 수신자여야 하는지 확인합니다. 그렇지 않으면 EC2 인스턴스가 소스도 대상도 아니기 때문에 트래픽이 삭제됩니다.

따라서 NAT 인스턴스는 일종의 프록시 역할을 하기 때문에 NAT 인스턴스를 사용할 때 소스/대상 확인을 비활성화 해야 합니다.

Bastion 호스트:

Bastion 호스트는 공격을 견디도록 설계 및 구성된 특수 목적 컴퓨터입니다. 이 서버는 일반적으로 단일 프로그램을 실행하며 공격 경로를 줄이기 위해 이 목적 이상으로 제거됩니다.

Bastion 호스트의 목적은 인터넷 게이트웨이를 통해 호스트를 노출하지 않고 시스템 관리 목적으로 프라이빗 서브넷 뒤에 있는 인스턴스에 원격으로 액세스하는 것입니다.

Bastion 호스트를 구현하는 가장 좋은 방법은 단일 IP 주소에 대한 보안 그룹 규칙만 있는 소규모 EC2 인스턴스를 생성하는 것입니다. 이것은 최대의 보안을 보장합니다.

인스턴스는 서로 다른 서버를 서로 연결하는 점프 서버로만 사용되기 때문에 큰 인스턴스보다 작은 인스턴스를 사용하는 것이 좋습니다.

프라이빗 서브넷의 인스턴스로 RDP 또는 SSH를 사용하려는 경우 Bastion 호스트를 사용합니다. 프라이빗 서브넷의 인스턴스에 인터넷 트래픽을 제공하려는 경우 NAT를 사용하십시오.

NAT 게이트웨이 및 NAT 인스턴스와 유사하게 Bastion 호스트는 공개 서브넷 내에 있습니다.

Pre-baked(미리 구운) Bastion Host AMI가 있습니다.

라우팅 테이블:

라우팅 테이블은 서브넷이 서로 통신할 수 있고 트래픽이 어디로 가야 하는지 알고 있는지 확인하는 데 사용됩니다.

생성하는 모든 서브넷은 VPC의 기본 라우팅 테이블과 자동으로 연결됩니다.

여러 라우팅 테이블을 가질 수 있습니다. 새 서브넷을 기본 라우팅 테이블과 연결하지 않으려면 다른 라우팅 테이블과 연결하도록 지정해야 합니다.

이 기본 동작으로 인해 잠재적인 보안 문제가 있음을 알아야 합니다. 기본 라우팅 테이블이 공용이면 이와 연결된 새 서브넷도 공용이 됩니다.

가장 좋은 방법은 새 서브넷이 연결된 기본 라우팅 테이블이 프라이빗인지 확인하는 것입니다.

즉, 기본 라우팅 테이블에 대해 인터넷으로 나가는 경로가 없는지 확인합니다. 그런 다음 대신 공개된 사용자 지정 라우팅 테이블을 생성할 수 있습니다. 새 서브넷에는 인터넷으로 나가는 경로가 자동으로 없습니다. 공개적으로 액세스할 수 있는 새 서브넷을 원하는 경우 사용자 지정 라우팅 테이블과 연결하기만 하면 됩니다.

라우팅 테이블은 인터넷뿐만 아니라 엔드포인트(비공개적으로 액세스하는 공개 서비스)에 액세스하도록 구성할 수 있습니다.

인터넷 게이트웨이:

인터넷에서 인스턴스에 액세스하기 위한 전제 조건인 인터넷 게이트웨이가 VPC에 연결되어 있지 않으면 당연히 VPC의 인스턴스에 연결할 수 없습니다.

일부 서브넷뿐만 아니라 모든 VPC를 비공개로 유지하려면 IGW를 연결하지 마십시오.

퍼블릭 IP 주소가 EC2 인스턴스에 할당되면 인터넷 게이트웨이에 의해 유효한 퍼블릭 엔드포인트로 효과적으로 등록됩니다. 그러나 각 인스턴스는 공용 IP가 아닌 사설 IP만 인식합니다. IGW만이 인스턴스에 속하는 공용 IP를 알고 있습니다.

EC2 인스턴스가 퍼블릭 인터넷에 대한 연결을 시작하면 인스턴스가 이에 대해 알지 못하더라도 퍼블릭 IP를 소스로 사용하여 요청이 전송됩니다. 이것은 IGW가 VPC로 들어오고 나가는 트래픽에 대해 프라이빗 IP가 퍼블릭 IP에 매핑되고 그 반대로 매핑되는 자체 NAT 변환을 수행하기 때문에 작동합니다.

따라서 인터넷의 트래픽이 인스턴스의 퍼블릭 IP 엔드포인트로 향하는 경우 IGW는 이를 수신하고 내부 프라이빗 IP를 사용하여 트래픽을 EC2 인스턴스로 전달합니다.

VPC당 하나의 IGW만 가질 수 있습니다.

요약 : IGW는 VPC를 인터넷에 연결합니다 .

Virtual Private Networks 가상 사설망(VPN):

VPC는 기업 데이터 센터와 AWS 클라우드 간의 다리 역할도 할 수 있습니다. VPC VPN(가상 사설망)을 사용하면 VPC가 온프레미스 환경의 확장이 됩니다.

당연히 VPC에서 시작한 인스턴스는 자체 온프레미스 서버와 통신할 수 없습니다. 먼저 다음을 수행하여 액세스를 허용할 수 있습니다.

- VPC에 가상 프라이빗 게이트웨이 연결
- 연결을 위한 사용자 지정 라우팅 테이블 생성
- 연결에서 오는 트래픽을 허용하도록 보안 그룹 규칙 업데이트
- 관리되는 VPN 연결 자체를 생성합니다.

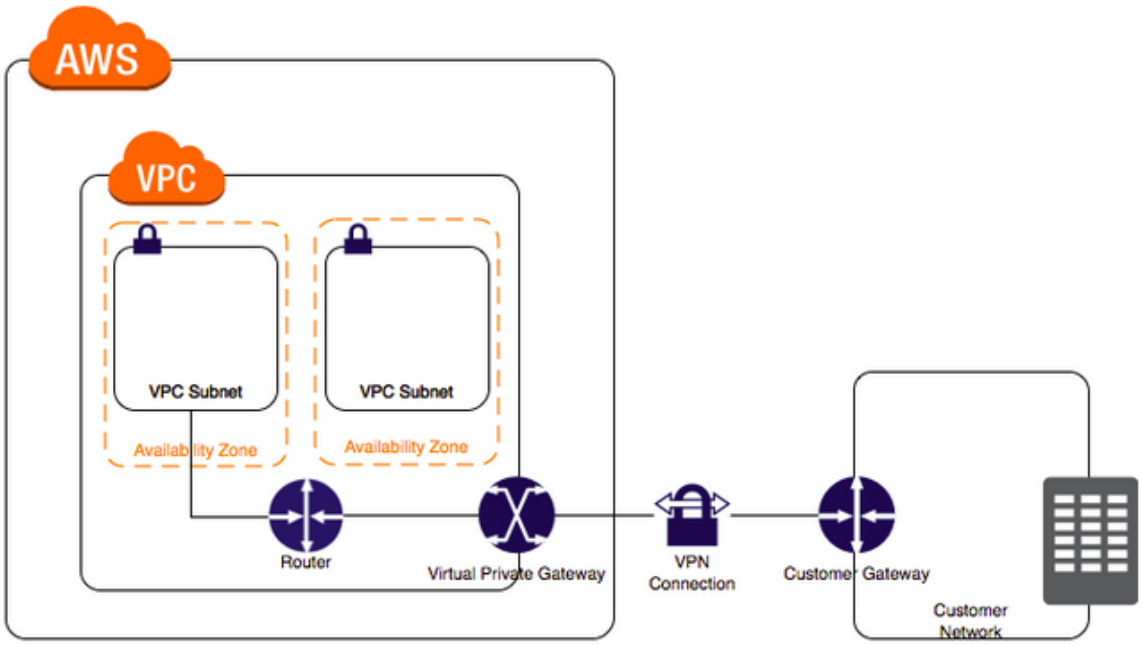
VPN 연결을 시작하려면 고객 게이트웨이 디바이스에 대한 AWS 정보를 제공하는 고객 게이트웨이 리소스도 AWS에 정의해야 합니다. 그리고 고객 게이트웨이 외부 인터페이스의 인터넷 라우팅 가능한 IP 주소를 설정해야 합니다.

고객 게이트웨이는 VPN 연결의 온프레미스 측에 있는 물리적 장치 또는 소프트웨어 애플리케이션입니다.

"VPN 연결"이라는 용어는 일반적인 개념이지만 AWS에 대한 VPN 연결은 항상 VPC와 자체 네트워크 간의 연결을 나타냅니다. AWS는 인터넷 프로토콜 보안(IPsec) VPN 연결을 지원합니다.

온프레미스 인프라기능

다음 다이어그램은 단일 VPN 연결을 보여줍니다.



위의 VPC에는 연결된 가상 프라이빗 게이트웨이(참고: 인터넷 게이트웨이 아님)가 있으며 VPN 연결을 활성화하기 위해 구성해야 하는 고객 게이트웨이가 포함된 원격 네트워크가 있습니다. 네트워크에 바인딩된 VPC의 모든 트래픽이 가상 프라이빗 게이트웨이로 라우팅되도록 라우팅을 설정합니다.

요약 : VPN 은 인터넷을 통해 온프레미스를 VPC와 연결합니다.

AWS 다이렉트커넥트:

Direct Connect는 프레미스와 AWS 간에 전용 네트워크 연결을 설정하는 AWS 서비스입니다. 이 개인 연결을 만들어 일반 인터넷 기반 연결에 비해 네트워크 비용을 줄이고 대역폭을 늘리며 보다 일관된 네트워크 환경을 제공할 수 있습니다.

Direct Connect의 사용 사례는 처리량이 많은 워크로드이거나 안정적이거나 안정적인 연결이 필요한 경우입니다.

VPN은 인터넷을 통해 온프레미스에 연결하고 DirectConnect는 사설 터널을 통해 온프레미스에 연결합니다.

AWS DirectConnect 연결 설정 단계:

- DirectConnect 콘솔에서 가상 인터페이스를 만듭니다. 이것은 공개 가상 인터페이스입니다.
- VPC 콘솔로 이동한 다음 VPN 연결로 이동합니다. 온프레미스에 대한 고객 게이트웨이를 만듭니다.
- 가상 프라이빗 게이트웨이를 생성하여 원하는 VPC 환경에 연결합니다.

- VPN 연결을 선택하고 새 VPN 연결을 만듭니다. 고객 게이트웨이와 가상 프라이빗 게이트웨이를 모두 선택합니다.
- VPN 연결을 사용할 수 있게 되면 고객 게이트웨이 또는 온프레미스 방화벽 자체에서 VPN을 설정합니다.

DirectConnect를 통해 AWS로 데이터 흐름은 다음과 같습니다. 온프레미스 라우터 -> 전용 회선 -> 자체 케이지/DMZ -> 교차 연결 회선 -> AWS Direct Connect 라우터 -> AWS 백본 -> AWS 클라우드

요약 : DirectConnect는 비공개 터널을 통해 온프레미스를 VPC와 연결합니다.

VPC 엔드포인트:

VPC 엔드포인트를 사용하면 인터넷 게이트웨이, NAT 디바이스, VPN 연결 또는 AWS Direct Connect 없이 VPC를 지원되는 AWS 서비스에 연결할 수 있습니다. VPC와 다른 AWS 서비스 간의 트래픽은 Amazon 에코시스템 내에 유지되며 이러한 엔드포인트는 HA이고 대역폭 제약이 없는 가상 장치입니다.

이는 기본적으로 광범위한 AWS 서비스와 쉽게 통신할 수 있는 EC2 인스턴스에 ENI를 연결하여 작동합니다.

게이트웨이 엔드포인트 는 라우팅 테이블에 항목을 생성하고 S3 또는 DynamoDB에 사용되는 프라이빗 엔드포인트를 가리키는 데 의존합니다. 게이트웨이 엔드포인트는 주로 사용자가 설정한 대상일 뿐입니다.

인터페이스 엔드포인트 는 AWS PrivateLink를 사용하고 프라이빗 IP 주소를 가지므로 라우팅 테이블의 대상이 아니라 자체 엔터티입니다. 이 때문에 시간당 $.01의 비용이 듭니다. 게이트웨이 엔드포인트는 설정을 위한 새로운 경로일 뿐이므로 무료입니다.

인터페이스 엔드포인트는 VPC 내에서 Elastic Network 인터페이스 또는 ENI(네트워크 카드 생각)를 프로비저닝합니다. 지원되는 다른 AWS 서비스를 오가는 트래픽의 진입 및 퇴장 역할을 합니다. DNS 레코드를 사용하여 트래픽을 인터페이스의 사설 IP 주소로 보냅니다. 게이트웨이 엔드포인트는 라우팅 테이블의 경로 접두사를 사용하여 S3 또는 DynamoDB를 위한 트래픽을 게이트웨이 엔드포인트로 보냅니다(0.0.0.0/0 -> igw로 생각).

인터페이스 엔드포인트를 보호하려면 보안 그룹을 사용하십시오. 그러나 게이트웨이 엔드포인트를 보호하려면 VPC 엔드포인트 정책을 사용하십시오.

요약 : VPC 엔드포인트 는 비공개 터널을 통해 VPC를 AWS 서비스와 연결합니다.

AWS 프라이빗링크:

AWS PrivateLink는 데이터가 공용 인터넷에 노출되지 않도록 하여 클라우드 기반 애플리케이션과 공유하는 데이터의 보안을 단순화합니다. AWS PrivateLink는 Amazon 네트워크에서 안전하게 서로 다른 VPC, AWS 서비스 및 온프레미스 애플리케이션 간의 비공개 연결을 제공합니다.

Direct Connect가 온프레미스 환경을 AWS에 연결한다는 점을 제외하고 AWS 클라우드에 대한 프라이빗 연결을 설정한다는 점에서 AWS Direct Connect 서비스와 유사합니다. 반면 PrivateLink는 이미 AWS에 있는 VPC 환경의 트래픽을 보호합니다.

이는 서로 다른 AWS 서비스가 종종 인터넷을 통해 서로 통신하기 때문에 유용합니다. 이러한 동작을 원하지 않고 AWS 서비스가 AWS 네트워크 내에서만 통신하도록 하려면 AWS PrivateLink를 사용하십시오. PrivateLink는 인터넷을 통과하지 않음으로써 무차별 대입 공격 및 분산 서비스 거부 공격과 같은 위협 벡터에 대한 노출을 줄입니다.

PrivateLink를 사용하면 다른 사람들이 자신의 VPC에서 연결할 수 있는 "엔드포인트"를 게시할 수 있습니다. 일반 VPC 엔드포인트와 유사하지만 AWS 서비스에 연결하는 대신 사용자가 엔드포인트에 연결할 수 있습니다.

또한 서비스가 사설 네트워크에서 직접 호스팅되는 것처럼 작동하도록 사설 IP 연결 및 보안 그룹을 사용하려고 합니다.

AWS PrivateLink는 AWS 네트워크 내에서 서로 통신하는 애플리케이션/서비스에 적용됩니다. VPC가 AWS 네트워크 내에서 서로 통신하려면 VPC 피어링을 사용하십시오.

요약: AWS PrivateLink는 비공개 터널을 통해 AWS 서비스를 다른 AWS 서비스와 연결합니다.

VPC 피어링:

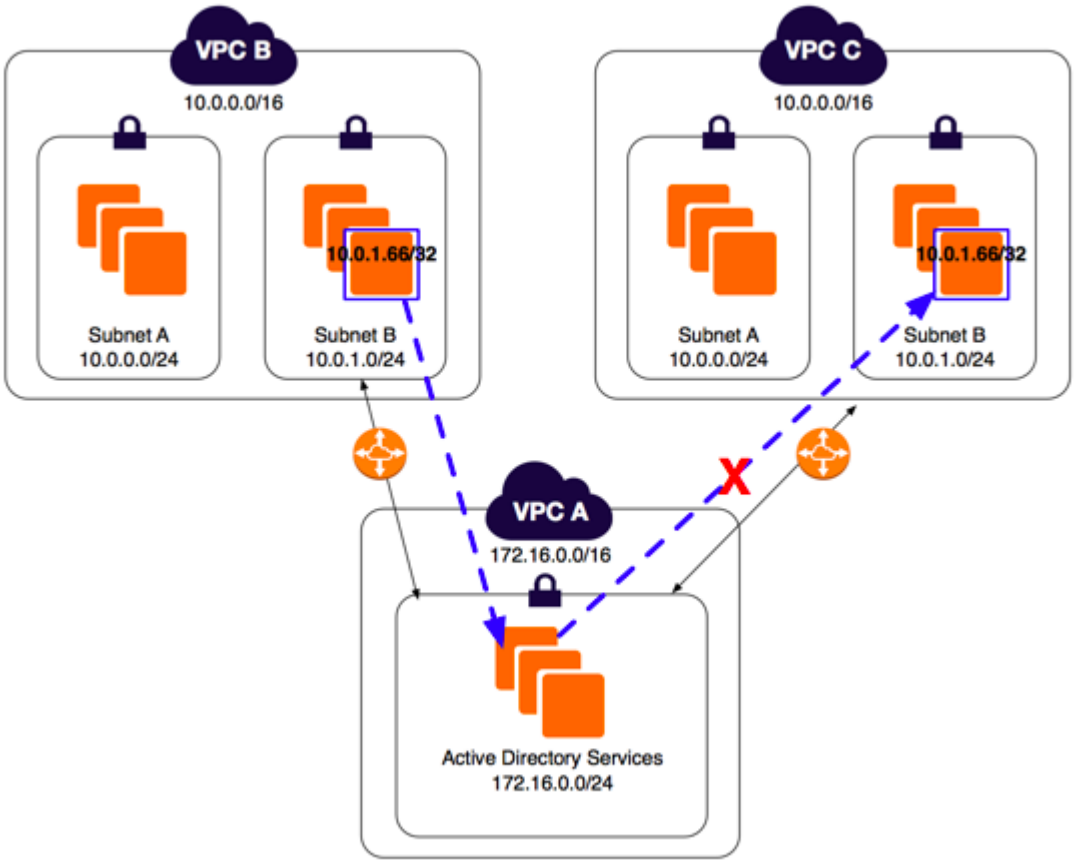
VPC 피어링을 사용하면 둘 다에 속하는 프라이빗 IP를 사용하여 직접 네트워크 경로를 통해 한 VPC를 다른 VPC와 연결할 수 있습니다. VPC 피어링을 사용하면 서로 다른 VPC의 인스턴스가 동일한 네트워크에 있는 것처럼 작동합니다.

동일한 지역에 있는지 여부와 완전히 다른 AWS 계정에 있는 VPC와 상관없이 자체 VPC 간에 VPC 피어링 연결을 생성할 수 있습니다.

VPC 피어링은 일반적으로 다른 것과 피어링하는 하나의 중앙 VPC가 있는 방식으로 수행됩니다. 중앙 VPC만 다른 VPC와 통신할 수 있습니다.

비중앙 VPC에 대해서는 전이적 피어링을 수행할 수 없습니다. 비중앙 VPC는 중앙 VPC를 통해 다른 비중앙 VPC로 이동할 수 없습니다. 서로 대화해야 하는 경우 비중앙 노드 간에 새 포털을 설정해야 합니다.

다음 다이어그램은 위의 아이디어를 강조 표시합니다. VPC B는 VPC A와 VPC 피어링이 활성화된 상태에서 자유롭게 통신할 수 있습니다. 그러나 VPC B는 VPC C와 대화를 계속할 수 없습니다. VPC A만 VPC C와 통신할 수 있습니다.



지원되지 않는 VPC 피어링 구성을 아는 것이 좋습니다.

- 겹치는 CIDR 블록
- 전이적 피어링
- 게이트웨이 또는 연결 장치를 통한 Edge to Edge 라우팅(VPN 연결, 인터넷 게이트웨이, AWS Direct Connect 연결 등)

여러 지역에서 피어링할 수 있지만 여러 가용 영역에 걸쳐 하나의 서브넷을 확장할 수는 없습니다. 그러나 동일한 가용 영역에 여러 서브넷이 있을 수 있습니다.

요약 : VPC 피어링은 비공개 터널을 통해 VPC를 다른 VPC에 연결합니다.

두 VPC간의 네트워크 연결 / 개인적으로 라우팅 / 온프레미스 X.

VPC 흐름 로그: Flow Logs

VPC 흐름 로그는 VPC로 들어오고 나가는 모든 트래픽에 대한 IP 정보를 캡처하는 기능입니다. 흐름 로그 데이터는 이 데이터를 보고 검색하고 조작할 수 있는 S3 버킷 또는 CloudWatch로 전송됩니다.

이동을 통해 다양한 단계에서 트래픽 흐름을 캡처할 수 있습니다.

- VPC로 들어오고 나가는 트래픽(IGW에서와 같이)
- 서브넷으로 들어오고 나가는 트래픽
- EC2 인스턴스(eth0, eth1 등)의 네트워크 인터페이스로 들어오고 나가는 트래픽

VPS 흐름 로그는 패킷 내용이 아닌 패킷 메타데이터를 캡처합니다. 같은 것들:

- 소스 IP
- 목적지 IP
- 패킷 크기
- 패킷 외부에서 관찰할 수 있는 모든 것.

유효한 트래픽, 무효 트래픽 또는 둘 다를 기록하도록 흐름 로그를 구성할 수 있습니다.

흐름 로그가 있는 VPC와 비교하여 다른 VPC에서 가져온 흐름 로그를 가질 수 있습니다. 그러나 다른 VPC는 VPC 피어링을 통해 그리고 AWS Organizations를 통해 귀하의 계정에서 피어링되어야 합니다.

로그에 태그를 지정하여 로그를 사용자 정의할 수 있습니다.

흐름 로그를 만든 후에는 구성을 변경할 수 없습니다. 새로 만들어야 합니다.

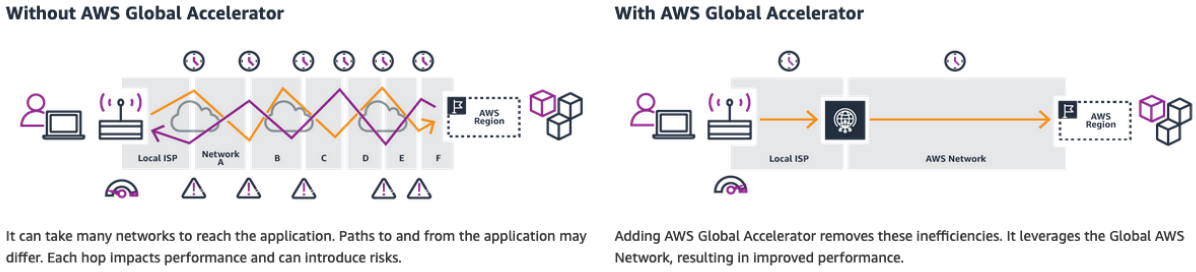
VPC 흐름 로그에서 모든 IP 트래픽이 모니터링되는 것은 아닙니다. 다음은 Flow Logs에서 무시되는 항목의 목록입니다.

- 인스턴스 메타데이터에 대한 쿼리 요청
- DHCP 트래픽
- AWS DNS 서버에 대한 쿼리 요청

AWS 글로벌 액셀러레이터:

AWS Global Accelerator는 연결을 가속화하여 사용자의 성능과 가용성을 개선합니다. Global Accelerator는 AWS 백본 위에 위치하며 전 세계적으로 최적의 엔드포인트로 트래픽을 전달합니다. 기본적으로 Global Accelerator는 사용할 수 있는 두 개의 고정 IP 주소를 제공합니다.

Global Accelerator는 AWS 리소스에 도달하기 위한 홉 수를 줄이는 데 도움이 됩니다. 사용자는 엣지 로케이션에 도착하면 모든 것이 AWS 글로벌 네트워크 내부에 유지됩니다. 일반적으로 응용 프로그램에 완전히 도달하려면 많은 네트워크가 필요하며 응용 프로그램과의 경로는 다를 수 있습니다. 각 홉에는 보안 또는 실패와 관련된 위험이 있습니다.



요약하면 Global Accelerator는 사용자와 애플리케이션 간의 빠르고 안정적인 파이프라인입니다.

원하는 곳으로 직접 안내하는 GPS(글로벌 액셀러레이터)가 있는 것과는 반대로 여행(웹 트래픽)을 하고 도시의 안전하지 않을 수 있는 지역(여러 네트워크를 방문하여 보안 위험을 증가시킬 수 있음)에서 길을 묻기 위해 멈추는 것과 같습니다. 불필요한 중지를 만들지 않고 (끝점) 이동합니다.

Cloudfront와 혼동될 수 있지만 CloudFront는 멀리 떨어진 오리진 서버에서 가져온 콘텐츠에 대한 캐시입니다.

CloudFront는 단순히 정적 콘텐츠를 가장 가까운 AWS POP(Point Of Presence) 위치에 캐싱하지만 Global Accelerator는 동일한 Amazon POP를 사용하여 초기 요청을 수락하고 서비스로 직접 라우팅합니다.

Route53의 지연 시간 기반 라우팅도 Global Accelerator와 유사하게 보일 수 있지만 Route 53은 단순히 사용자가 사용할 지역을 선택하는 데 도움이 됩니다. Route53은 실제로 빠른 네트워크 경로를 제공하는 것과 관련이 없습니다.

Global Accelerator는 빠른 지역 장애 조치도 제공합니다.

Simple Queuing Service(SQS)

SQS 단순화:

SQS는 대기열이 처리하기를 기다리는 동안 메시지를 저장하는 데 사용할 수 있는 메시지 대기열에 대한 액세스를 제공하는 웹 기반 서비스입니다. 시스템을 분리하고 AWS 리소스를 수평적으로 확장하는 데 도움이 됩니다.

SQS 주요 세부사항:

SQS의 요점은 시스템 간에 작업을 분리하는 것입니다. 이런 식으로 시스템의 다운스트림 서비스는 업스트림 서비스가 데이터를 공급할 때가 아니라 준비가 되었을 때 작업을 수행할 수 있습니다.

SQS 없이 실행되는 가상의 AWS 환경에서 애플리케이션 A 는 애플리케이션 B가 정보를 수신할 준비가 되었는지 여부에 관계없이 애플리케이션 B 데이터 를 전달 합니다. 그러나 SQS에는 데이터가 버퍼에 임시로 저장되는 중간 단계가 있습니다. 애플리케이션 B가 임시로 저장된 데이터를 가져올 때까지 대기합니다. SQS는 푸시 기반 서비스가 아니므로 SQS가 정보를 쿼리하는 다른 서비스와 함께 작동해야 합니다.

SQS 대기열에는 두 가지 유형이 있습니다. 표준 및 FIFO . 표준 대기열은 메시지 크기에 따라 순서 없이 수신될 수 있지만 그렇지 않으면 SQS 대기열이 최적화하기로 결정합니다. FIFO 대기열은 대기열에 들어간 메시지의 순서가 대기열에서 나가는 메시지의 순서와 동일함을 보장합니다.

표준 SQS 대기열은 메시지가 한 번 이상 배달되도록 보장하며 이 때문에 비동기식 및 고도로 분산된 아키텍처로 인해 메시지가 두 번 이상 배달될 수 있는 경우가 있습니다. 표준 대기열을 사용하면 초당 거의 무제한의 트랜잭션이 있습니다.

FIFO SQS 대기열은 정확히 한 번 처리를 보장하며 초당 300개의 트랜잭션으로 제한됩니다.

대기열의 메시지는 1분에서 14일까지 보관할 수 있으며 기본 보관 기간은 4일입니다.

SQS의 표시 시간 제한은 대기열에서 배달하도록 표시된 메시지에 독자가 완전히 수신할 시간 프레임을 제공하는 메커니즘입니다. 이것은 일시적으로 다른 독자에게 보이지 않도록 하여 수행됩니다. 제한 시간 내에 메시지가 완전히 처리되지 않으면 메시지가 다시 표시됩니다. 이것은 메시지를 복제할 수 있는 또 다른 방법입니다. 중복 가능성을 줄이려면 가시성 시간 초과를 늘리십시오.

가시성 제한 시간 최대값은 12시간입니다.

SQS 대기열의 메시지는 EC2 인스턴스가 메시지를 처리한 후에도 해당 메시지를 삭제할 때까지 계속 존재한다는 점을 항상 기억하십시오. 가시성 제한 시간이 만료되면 메시지가 다시 수신 및 처리되지 않도록 처리 후에 메시지를 삭제해야 합니다.

SQS 대기열은 메시지를 무제한으로 포함할 수 있습니다.

SQS 대기열의 개별 항목에 우선 순위를 설정할 수 없습니다. 메시징의 우선 순위가 중요한 경우 두 개의 별도 SQS 대기열을 만듭니다. 우선 순위 메시지에 대한 SQS 대기열은 EC2 인스턴스에서 먼저 폴링할 수 있으며 완료되면 두 번째 대기열의 메시지를 다음에 처리할 수 있습니다.

SQS 폴링:

폴링은 메시지나 작업에 대해 SQS를 쿼리하는 수단입니다. Amazon SQS는 대기열에서 메시지를 수신하기 위해 짧은 폴링과 긴 폴링을 제공합니다. 기본적으로 대기열은 짧은 폴링을 사용합니다.

SQS 롱 폴링 : 이 폴링 기술은 큐가 현재 가득 찼는지 비어 있는지에 관계없이 메시지가 있는 경우에만 큐에서 반환됩니다. 이런 식으로 판독기는 시간 초과가 설정되거나 메시지가 최종적으로 도착할 때까지 기다려야 합니다. SQS 긴 폴링은 메시지가 대기열에 도착할 때까지 응답을 반환하지 않으므로 시간이 지남에 따라 전체 비용이 절감됩니다. 성능 향상 ↑

SQS 숏 폴링 : 이 폴링 기술은 이미 대기열에 저장된 메시지나 빈 손으로 메시지를 즉시 반환합니다. 응답 즉시

그 메시지 찾지 못해도

ReceiveMessageWaitTimeSeconds는 Short 또는 Long 폴링을 사용하는지 여부를 결정하는 대기열 속성입니다. 기본적으로 그 값은 0이며 이는 짧은 폴링을 사용하고 있음을 의미합니다. 0보다 큰 값으로 설정하면 롱 폴링입니다.

대기열을 폴링할 때마다 요금이 발생합니다. 따라서 사용 사례에 맞는 폴링 전략을 신중하게 결정하는 것이 중요합니다.

Simple Workflow Service(SWF)

SWF 단순화:

SWF는 분산된 응용 프로그램 구성 요소 간의 작업을 쉽게 조정할 수 있게 해주는 웹 서비스입니다. SWF에는 미디어 처리, 웹 앱 백엔드, 비즈니스 프로세스 워크플로 및 분석 파이프라인을 비롯한 다양한 사용 사례가 있습니다.

SWF 키 세부 정보:

SWF는 응용 프로그램과 사람 간의 작업을 조정하는 방법입니다. 디지털 워크플로와 인간 중심 워크플로를 결합한 서비스입니다.

인간 중심 워크플로의 예는 Amazon 창고 작업자가 Amazon 주문의 일부로 아이템을 찾아 배송하는 프로세스입니다.

SWF는 작업 지향 API를 제공하며 작업이 한 번만 할당되고 절대 중복되지 않도록 합니다. 다시 한 번 Amazon 창고 작업자를 예로 들면 이것이 의미가 있습니다. 아마존은 돈을 잃을 것이므로 같은 품목을 두 번 보내고 싶지 않을 것입니다.

SWF 파이프라인은 작업을 완료하는 데 도움이 되는 세 가지 작업자 응용 프로그램로 구성됩니다.

- SWF 액터는 워크플로의 시작을 트리거하는 작업자입니다.
- SWF 결정자는 워크플로가 시작된 후 흐름을 제어하는 작업자입니다.
- SWF 활동 작업자는 실제로 작업을 완료할 때까지 수행하는 작업자입니다.

SWF를 사용하면 SQS의 최대 보존 기간이 14일인 것에 비해 워크플로 실행이 최대 1년 동안 지속될 수 있습니다.

Simple Notification Service(SNS)

SNS 단순화:

Simple Notification Service는 특정 주제에 대한 정보를 받고자 하는 가입자에게 맞춤형 메시지를 게시할 수 있는 확장성이 뛰어나고 유연하며 비용 효율적인 방법을 제공하는 푸시 기반 메시징 서비스입니다.

SNS 키 세부 정보:

SNS는 주로 알람이나 알림을 보내는 데 사용됩니다.

SNS는 처리량이 많은 푸시 기반 다대다 메시징에 대한 주제를 제공합니다.

Amazon SNS 주제를 사용하여 게시자 시스템은 Amazon SQS 대기열, AWS Lambda 함수 및 HTTP/S 웹훅을 포함하여 병렬 처리를 위해 많은 구독자 엔드포인트로 메시지를 팬아웃할 수 있습니다. 또한 SNS는 모바일 푸시, SMS 및 이메일을 사용하여 최종 사용자에게 알림을 보내는 데 사용할 수 있습니다.

이러한 푸시 알림을 Apple, Google, Fire OS 및 Windows 장치로 보낼 수 있습니다.

SNS를 사용하면 주제를 사용하여 여러 수신자를 그룹화할 수 있습니다. 주제는 수신자가 동일한 알림의 동일한 사본을 동적으로 구독할 수 있도록 하는 액세스 지점입니다.

하나의 주제는 여러 엔드포인트 유형에 대한 전달을 지원할 수 있습니다. 주제에 게시하면 SNS는 해당 메시지의 복사본을 적절한 형식으로 지정하여 어떤 종류의 장치로 보내든 보냅니다.

메시지 손실을 방지하기 위해 메시지는 여러 AZ에 중복 저장됩니다.

메시지의 즉각적인 푸시로 인해 SNS와 관련된 장단기 폴링이 없습니다.

SNS는 여러 전송 프로토콜을 통해 유연한 메시지 전달을 제공하며 간단한 API를 가지고 있습니다.

Kinesis

Kinesis 단순화:

Amazon Kinesis를 사용하면 실시간 스트리밍 데이터를 쉽게 수집, 처리 및 분석할 수 있으므로 적시에 통찰력을 얻고 새로운 정보에 빠르게 대응할 수 있습니다. Amazon Kinesis를 사용하면 비디오, 오디오, 애플리케이션 로그, 웹 사이트 클릭스트림 및 기계 학습, 분석 및 기타 애플리케이션을 위한 IoT 원격 측정 데이터와 같은 실시간 데이터를 수집할 수 있습니다. Amazon Kinesis를 사용하면 데이터가 도착하는 즉시 처리 및 분석하고 처리가 시작되기 전에 모든 데이터가 수집될 때까지 기다릴 필요 없이 즉시 응답할 수 있습니다.

Kinesis 키 세부 정보:

Amazon Kinesis를 사용하면 AWS에 들어오는 대용량 데이터를 쉽게 로드하고 분석할 수 있습니다.

Kinesis는 데이터를 지속적으로 AWS로 전송하는 디바이스에서 실시간 데이터 스트림(지속적으로 생성되는 데이터)을 처리하는 데 사용되어 해당 데이터를 수집 및 분석할 수 있습니다.

데이터 처리량에 맞게 자동으로 확장되며 지속적인 관리가 필요하지 않은 완전 관리형 서비스입니다. 또한 데이터를 로드하기 전에 일괄 처리, 압축 및 암호화하여 대상에서 사용되는 스토리지의 양을 최소화하고 보안을 강화할 수 있습니다.

Kinesis에는 세 가지 유형이 있습니다. 보존기간 → 보관기간

- Kinesis Streams 보관기간 - 데이터가 녹아질 시점부터 현재까지 액세스X

  - Kinesis Streams는 데이터 생산자가 데이터를 Kinesis Streams로 스트리밍하는 곳에서 작동합니다. Kinesis Streams는 입력된 데이터를 하루부터 최대 7일까지 보유할 수 있습니다. Kinesis Streams 내부에 있는 데이터는 샤드에 포함됩니다.

  - Kinesis Streams는 웹사이트 클릭스트림, 금융 거래, 소셜 미디어 피드, IT 로그 및 위치 추적 이벤트와 같은 수십만 소스에서 시간당 테라바이트의 데이터를 지속적으로 캡처하고 저장할 수 있습니다. 예: Amazon과 같은 대규모 온라인 상점의 구매 요청, 주가, Netflix 콘텐츠, Twitch 콘텐츠, 온라인 게임 데이터, Uber 위치 및 방향 등

- Kinesis Firehose

  - Amazon Kinesis Firehose는 스트리밍 데이터를 데이터 스토어 및 분석 도구에 로드하는 가장 쉬운 방법입니다. 데이터가 Kinesis Firehose로 스트리밍되면 저장할 영구 스토리지가 없습니다. 데이터는 들어오는 대로 분석해야 하므로

Kinesis Firehose 내부에 Lambda 함수를 포함하는 것은 선택 사항입니다. 일단 처리되면 데이터를 다른 곳으로 보냅니다.

- Kinesis Firehose는 스트리밍 데이터를 캡처, 변환 및 Amazon S3, Amazon Redshift, Amazon Elasticsearch Service 및 Splunk로 로드할 수 있으므로 현재 이미 사용 중인 기존 비즈니스 인텔리전스 도구 및 대시보드로 거의 실시간 분석이 가능합니다.

- Kinesis Analytics
  - Kinesis Analytics는 Kinesis Streams 및 Kinesis Firehose와 함께 작동하며 즉석에서 데이터를 분석할 수 있습니다. Kinesis Analytics 내의 데이터는 처리가 완료되면 다른 곳으로도 전송됩니다. Kinesis 서비스 자체 내부의 데이터를 분석합니다.

파티션 키는 Kinesis와 함께 사용되므로 샤드별로 데이터를 구성할 수 있습니다. 이렇게 하면 특정 장치의 입력에 대상을 특정 샤드로 제한하는 키를 할당할 수 있습니다.

파티션 키는 샤드 내에서 순서를 유지하려는 경우에 유용합니다.

Kinesis Streams에서 읽는 소비자 또는 EC2 인스턴스는 샤드 내부로 이동하여 그 안에 무엇이 있는지 분석할 수 있습니다. 데이터 분석 또는 구문 분석이 완료되면 소비자는 데이터를 DB 또는 S3와 같은 저장을 위해 여러 위치에 전달할 수 있습니다.

Kinesis 스트림의 총 용량은 구성 샤드 내 데이터의 합계입니다.

샤드 테이블에 할당된 쓰기 용량은 언제든지 늘릴 수 있습니다.

함수 호출시 { 동기 - 이벤트 처리 / 응답 반환 기다리기
비동기 - 이벤트 대기열 저장 / 즉시 응답 반환 처리 후

Lambda

단순화된 람다:

AWS Lambda를 사용하면 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있습니다. 사용한 컴퓨팅 시간에 대해서만 비용을 지불합니다. Lambda를 사용하면 거의 모든 유형의 애플리케이션 또는 백엔드 서비스에 대한 코드를 실행할 수 있으며 모두 관리가 필요 없습니다. 코드를 업로드하면 Lambda가 고가용성으로 코드를 실행하고 확장하는 데 필요한 모든 것을 처리합니다. 다른 AWS 서비스에서 자동으로 트리거되거나 웹 또는 모바일 앱에서 직접 호출되도록 코드를 설정할 수 있습니다.

람다 키 세부 정보:

Lambda는 코드를 함수로 업로드하고 AWS가 함수가 성공적으로 실행될 수 있도록 함수 아래에 필요한 세부 정보를 프로비저닝하는 컴퓨팅 서비스입니다.

AWS Lambda는 궁극적인 추상화 계층입니다. 코드만 걱정하면 AWS가 나머지 모든 작업을 수행합니다.

Lambda는 Go, Python, C#, PowerShell, Node.js 및 Java를 지원합니다.

각 Lambda 함수는 하나의 요청에 매핑됩니다. Lambda는 자동으로 수평으로 확장됩니다.

Lambda는 요청 수에 따라 가격이 책정되며 처음 100만 개는 무료입니다. 이후 100만 달러는 0.20 달러입니다.

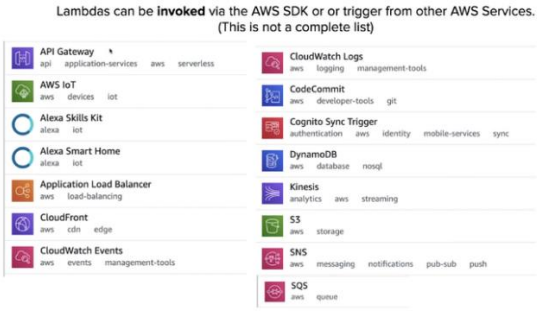
또한 Lambda는 코드의 런타임에 따라 가격이 책정되며, 가장 가까운 100MB로 반올림되며 코드가 할당하는 메모리 양입니다.

Lambda는 전 세계적으로 작동합니다.

Lambda 함수는 다른 Lambda 함수를 트리거할 수 있습니다.

Lambda를 AWS 에코시스템의 변경 사항에 따라 실행되는 이벤트 기반 서비스로 사용할 수 있습니다.

AWS SDK 또는 API Gateway를 통한 API 호출을 통해 HTTP 이벤트에 대한 응답으로 Lambda를 핸들러로 사용할 수도 있습니다.



환경 변수를 사용하는 Lambda 함수를 생성하거나 업데이트할 때 AWS Lambda는 AWS Key Management Service를 사용하여 이를 암호화합니다. Lambda 함수가 호출되면 해당 값이 해독되어 Lambda 코드에서 사용할 수 있습니다.

한 지역에서 환경 변수를 사용하는 Lambda 함수를 처음 생성하거나 업데이트할 때 AWS KMS 내에서 기본 서비스 키가 자동으로 생성됩니다. 이 키는 환경 변수를 암호화하는 데 사용됩니다. 그러나 Lambda 함수가 생성된 후 암호화 도우미를 사용하고 KMS를 사용하여 환경 변수를 암호화하려면 자체 AWS KMS 키를 생성하고 기본 키 대신 선택해야 합니다.

Lambda 함수가 프라이빗 VPC 내부의 리소스에 액세스할 수 있도록 하려면 VPC 서브넷 ID 및 보안 그룹 ID를 포함하는 추가 VPC별 구성 정보를 제공해야 합니다. AWS Lambda는 이 정보를 사용하여 함수가 프라이빗 VPC 내의 다른 리소스에 안전하게 연결할 수 있도록 하는 탄력적 네트워크 인터페이스(ENI)를 설정합니다.

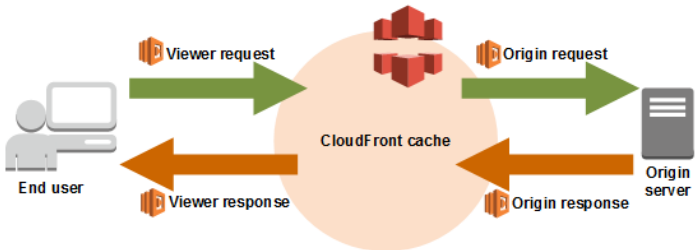
AWS X-Ray를 사용하면 예기치 않은 동작이 발생할 경우 Lambda 함수를 디버그할 수 있습니다.

람다@에지:

Lambda@Edge를 사용하여 Lambda 함수가 CloudFront에서 제공하는 콘텐츠를 사용자 지정하도록 허용할 수 있습니다.

CloudFront 엣지 로케이션에 컴퓨팅 용량을 추가하고 애플리케이션 뷰어에 더 가까운 AWS 로케이션에서 기능을 실행할 수 있습니다. 함수는 서버를 프로비저닝하거나 관리하지 않고 CloudFront 이벤트에 대한 응답으로 실행됩니다. Lambda 함수를 사용하여 다음 시점에서 CloudFront 요청 및 응답을 변경할 수 있습니다.

- CloudFront가 뷰어로부터 요청을 받은 후(뷰어 요청)
- CloudFront가 요청을 오리진으로 전달하기 전(오리진 요청)
- CloudFront에서 오리진으로부터 응답을 받은 후(오리진 응답)
- CloudFront가 최종 사용자에게 응답을 전달하기 전(시청자 응답)



Lambda@Edge를 사용하여 오리진 인프라를 단순화하고 줄일 수 있습니다.

API 게이트웨이

API 게이트웨이 간소화:

API Gateway는 전체 API를 쉽게 구축, 게시, 관리 및 보호할 수 있는 개발자를 위한 완전 관리형 서비스입니다. AWS Management Console에서 몇 번의 클릭으로 애플리케이션이 데이터, 비즈니스 로직 또는 백엔드 서비스의 기능(예: EC2)에서 실행되는 AWS Lambda 또는 모든 웹 애플리케이션에서.

API 게이트웨이 키 세부 정보:

Amazon API Gateway는 트래픽 관리, 권한 부여 및 액세스 제어, 모니터링, API 버전 관리를 포함하여 최대 수십만 건의 동시 API 호출을 수락하고 처리하는 것과 관련된 모든 작업을 처리합니다.

Amazon API Gateway에는 최소 비용이나 시작 비용이 없습니다. 수신한 API 호출과 전송된 데이터 양에 대해서만 비용을 지불하면 됩니다.

API Gateway는 API에 대해 다음을 수행합니다.

- RESTful 기능을 위해 HTTP(S) 끝점을 노출합니다.
- 서버리스 기능을 사용하여 Lambda 및 DynamoDB에 연결
- 각 API 엔드포인트를 다른 대상으로 보낼 수 있습니다.
- 저렴하고 효율적으로 운영
- 쉽고 간편하게 확장
- 공격을 방지하기 위해 요청을 제한할 수 있음
- API 키를 통한 사용량 추적 및 제어
- 버전 제어 가능
- 모니터링 및 관찰을 위해 CloudWatch에 연결 가능

API Gateway는 AWS Lambda와 함께 작동할 수 있으므로 서버를 유지 관리할 필요 없이 API와 코드를 실행할 수 있습니다.

Amazon API Gateway는 전역 및 서비스 호출을 포함한 여러 수준에서 조절을 제공합니다.

- 소프트웨어에서 조절 프로세스 또는 조절 컨트롤러라고도 하는 조절 컨트롤러는 응용 프로그램 처리가 수행되는 속도를 정적으로 또는 동적으로 조절하는 역할을 하는 프로세스입니다.
- 스로틀 제한은 표준 속도 및 버스트에 대해 설정할 수 있습니다. 예를 들어 API 소유자는 REST API에서 특정 메서드에 대해 초당 1,000개 요청의 비율 제한을 설정할 수 있으며 몇 초 동안 초당 2,000개의 요청 버스트를 처리하도록 Amazon API Gateway를 구성할 수도 있습니다.

- Amazon API Gateway는 초당 요청 수를 추적합니다. 제한을 초과하는 모든 요청은 429 HTTP 응답을 받습니다. Amazon API Gateway에서 생성된 클라이언트 SDK는 이 응답을 받으면 자동으로 호출을 재시도합니다.

Amazon API Gateway 캐시를 프로비저닝하고 크기를 기가바이트 단위로 지정하여 API 호출에 캐싱을 추가할 수 있습니다. 캐시는 API의 특정 단계에 대해 프로비저닝됩니다. 이렇게 하면 성능이 향상되고 백엔드로 전송되는 트래픽이 줄어듭니다. 캐시 설정을 사용하면 캐시 키가 구축되는 방식과 각 방법에 대해 저장된 데이터의 TTL(Time-to-Live)을 제어할 수 있습니다. Amazon API Gateway는 또한 각 단계의 캐시를 무효화하는 데 도움이 되는 관리 API를 노출합니다.

API 캐싱을 활성화하여 엔드포인트의 지연 시간을 개선하고 I/O를 줄일 수 있습니다.

특정 API 단계(버전 제어 버전)에 대해 캐싱할 때 특정 TTL에 대한 응답을 몇 초 만에 캐싱합니다.

API Gateway는 AWS Certificate Manager를 지원하며 무료 TLS/SSL 인증서를 사용할 수 있습니다.

API Gateway에는 두 가지 종류의 API 호출이 있습니다.

- API Gateway API를 호출하여 REST API를 생성, 수정, 삭제 또는 배포합니다. 이는 CloudTrail에 기록됩니다.
- 사용자 지정 기능을 제공하기 위해 개발자가 설정한 API 호출: CloudTrail에 기록되지 않습니다.

교차 출처 리소스 공유:

컴퓨팅에서 동일 출처 정책은 웹 브라우저가 한 페이지에 포함된 스크립트가 다른 페이지의 데이터에 액세스할 수 있도록 허용하지만 두 페이지의 출처가 동일한 경우에만 중요한 개념입니다.

이 동작은 브라우저에서 적용되지만 cURL 및 PostMan과 같은 도구에서는 무시됩니다.

CORS(교차 출처 자원 공유)는 출처의 서버가 동일 출처 정책을 완화할 수 있는 한 가지 방법입니다. CORS를 사용하면 글꼴과 같은 제한된 리소스를 공유하여 첫 번째 리소스가 공유된 원래 도메인 외부의 다른 도메인에서 요청할 수 있습니다.

CORS는 한 도메인에 로드된 클라이언트 웹 응용 프로그램이 다른 도메인의 리소스와 상호 작용하는 방법을 정의합니다. CORS 지원을 통해 Amazon S3로 풍부한 클라이언트 측 웹 애플리케이션을 구축하고 Amazon S3 리소스에 대한 교차 출처 액세스를 선택적으로 허용할 수 있습니다.

원격 리소스에서 원본 정책을 읽을 수 없다는 오류가 발생한 경우 API Gateway에서 CORS를 활성화해야 합니다.

CORS는 클라이언트(웹 브라우저) 측에서 시행됩니다.

이 문제의 일반적인 예는 API Gateway에서 여러 도메인에 대해 Javascript/AJAX가 있는 사이트를 사용하는 경우입니다. CORS가 활성화되어 있는지 확인해야 합니다.

CORS는 XSS 공격을 방지하지 않지만 CSRF 공격은 방지합니다. 엔드포인트에서 제공하는 데이터를

사용할 수 있는 사용자를 제어합니다. 따라서 일기예보를 확인하는 API에 대한 콜백이 있는 날씨 웹사이트가 있는 경우 누군가가 웹사이트를 탐색할 때 API에 JavaScript 호출을 제공하는 웹사이트를 작성하지 못하도록 막을 수 있습니다.

누군가 악의적인 호출을 시도하면 브라우저가 CORS 헤더를 읽고 요청이 발생하지 않도록 하여 공격으로부터 사용자를 보호합니다.

CloudFromation

간소화된 CloudFormation:

CloudFormation은 전체 클라우드 기반 환경을 프로비저닝하기 위한 자동화된 도구입니다. 이는 애플리케이션 설정(백엔드에 Z 유형 DB가 있는 Y 유형의 X개 웹 서버 등) 내부에 원하는 항목에 대한 지침을 코드화하는 Terraform과 유사합니다. 마크업에서 원하는 것을 설명하고 AWS가 관련된 실제 프로비저닝 작업을 수행하도록 하는 것이 훨씬 더 쉽습니다.

CloudFormation 키 세부 정보:

CloudFormation의 주요 사용 사례는 복잡하고 강력한 기능이 많기 때문에 고급 설정 및 프로덕션 환경입니다.

CloudFormation 템플릿을 사용하여 인프라를 생성, 업데이트 및 삭제할 수 있습니다.

템플릿은 YAML 또는 JSON으로 작성됩니다.

전체 CloudFormation 설정을 스택이라고 합니다.

템플릿이 생성되면 AWS에서 해당 스택을 만듭니다. 이것은 해당 템플릿의 살아 있고 활동적인 표현입니다. 하나의 템플릿으로 무한한 스택을 생성할 수 있습니다.

리소스 필드는 CloudFormation 템플릿을 생성할 때 유일한 필수 필드입니다 .

롤백 트리거를 사용하면 스택이 빌드될 때 스택 생성을 모니터링할 수 있습니다. 오류가 발생하면 이름에서 알 수 있듯이 롤백을 트리거할 수 있습니다.

AWS Quick Start는 AWS 엔지니어가 설계한 많은 고품질 CloudFormation 스택으로 구성됩니다.

EC2 인스턴스를 가동하는 예제 템플릿:

```
Resources:
  Instance: ## Logical Resource
    Type: 'AWS::EC2::Instance' ## This is what will be created
    Properties: ## Configure the resources in a particular way
      ImageId: !Ref LatestAmiId
      Instance Type: !Ref Instance Type
      KeyName: !Ref Keyname
```

스택의 모든 논리적 리소스에 대해 CloudFormation은 AWS 계정에 해당 물리적 리소스를 만듭니다. 논리적 리소스와 물리적 리소스를 동기화 상태로 유지하는 것이 CloudFormation의 역할입니다.

템플릿을 업데이트한 다음 동일한 스택을 업데이트하는 데 사용할 수 있습니다.

ElasticBeanstalk

ElasticBeanstalk 단순화:

ElasticBeanstalk는 기존 애플리케이션을 클라우드에 배포하여 프로비저닝 프로세스를 스크립팅하는 또 다른 방법입니다. ElasticBeanstalk는 클라우드에 대해 거의 알지 못하고 코드를 배포하는 가장 간단한 방법을 원하는 개발자를 대상으로 합니다.

ElasticBeanstalk 키 세부 정보:

애플리케이션을 업로드하기만 하면 ElasticBeanstalk가 기본 인프라를 처리합니다.

ElasticBeanstalk에는 용량 프로비저닝이 있으므로 처음부터 자동 확장과 함께 사용할 수 있습니다. ElasticBeanstalk는 이미 업데이트된 버전으로 복제본을 준비하여 애플리케이션에 업데이트를 적용합니다. 그런 다음 이 복제본을 원본으로 교체합니다. 이것은 업데이트된 애플리케이션이 실패할 경우를 대비한 예방 조치로 수행됩니다. 앱이 실패할 경우 ElasticBeanstalk는 이전 버전의 원본 복사본으로 다시 전환되며 애플리케이션을 사용하는 사용자는 다운타임을 겪지 않습니다.

Elastic Beanstalk는 컨테이너에서 웹 애플리케이션 배포를 지원하므로 ElasticBeanstalk를 사용하여 Docker를 호스팅할 수도 있습니다. Docker 컨테이너를 사용하면 고유한 런타임 환경, 고유한 플랫폼, 프로그래밍 언어 및 다른 플랫폼에서 지원하지 않는 애플리케이션 종속성(예: 패키지 관리자 또는 도구)을 정의할 수 있습니다. ElasticBeanstalk를 사용하면 Docker 컨테이너가 이미 자체 포함되어 있고 실행에 필요한 모든 구성 정보와 소프트웨어가 포함되어 있으므로 Docker를 쉽게 배포할 수 있습니다.

AWS Organizations

간소화된 AWS Organizations:

AWS Organizations는 여러 AWS 계정을 생성하고 중앙에서 관리하는 조직으로 통합할 수 있는 계정 관리 서비스입니다.

AWS Organizations 주요 세부 정보:

모범 사례는 루트 계정을 사용하여 리소스를 배포하는 데 사용되는 별도의 계정으로만 청구를 관리하는 것입니다.

AWS Organizations의 요점은 루트 계정 아래에 있는 별도의 계정에 권한을 배포하고 해당 정책을 적용하는 것입니다. AWS Organizations는 AWS에서 워크로드를 확장하고 확장함에 따라 환경을 중앙에서 관리할 수 있도록 도와줍니다.

조직 단위(OU)를 사용하여 유사한 계정을 그룹화하여 단일 단위로 관리할 수 있습니다. 이렇게 하면 계정 관리가 크게 간소화됩니다.

정책 기반 제어를 OU에 연결할 수 있으며 OU 내의 모든 계정은 자동으로 정책을 상속합니다. 따라서 회사의 개발자가 모두 자신의 샌드박스 AWS 계정을 가지고 있는 경우 단일 단위로 취급될 수 있으며 동일한 정책에 의해 제한될 수 있습니다.

AWS Organizations를 사용하면 조직 단위 또는 보다 구체적으로 개별 계정에서 서비스 제어 정책(SCP)을 사용하여 서비스를 활성화하거나 비활성화할 수 있습니다.

AWS Organizations에서 SCP를 사용하여 모든 IAM 보안 주체(사용자 및 역할)가 이를 준수하도록 액세스 제어를 설정합니다. SCP를 사용하면 Conditions, Resources 및 NotAction 을 지정 하여 조직 또는 조직 단위의 계정 전체에 대한 액세스를 거부할 수 있습니다. 예를 들어 SCP를 사용하여 특정 AWS 리전에 대한 액세스를 제한하거나 중앙 관리자에게 사용되는 IAM 역할과 같은 공통 리소스 삭제를 방지할 수 있습니다.

알아두면 좋은 질문

다음 섹션에는 시험에 나올 수 있는 서비스, 기능 및 기술이 포함되어 있습니다. 또한 AWS를 사용하는 엔지니어로서 알고 있으면 매우 유용합니다. 시험에 다음 항목이 나타날 경우 세부적으로 시험하지 않습니다. 이름 뒤에 숨겨진 의미를 알아야 합니다. 자신의 경력에 도움이 되도록 각 항목을 깊이 있게 배우는 것은 좋은 생각이지만 시험에 꼭 필요한 것은 아닙니다.

Amazon Cognito란 무엇입니까?

Amazon Cognito에 대해 논의하기 전에 먼저 Web Identity Federation이 무엇인지 이해하는 것이 중요합니다. Web Identity Federation을 사용하면 Facebook, Google, Amazon 등과 같은 웹 기반 자격 증명 공급자에 성공적으로 인증을 받은 사용자에게 AWS 리소스에 대한 액세스 권한을 부여할 수 있습니다. 이러한 서비스에 성공적으로 로그인하면 사용자에게 다음의 인증 코드가 제공됩니다. 임시 AWS 자격 증명을 얻는 데 사용할 수 있는 자격 증명 공급자.

Amazon Cognito는 Web Identity Federation을 제공하는 Amazon 서비스입니다. 애플리케이션에서 사용자에게 Facebook에 로그인하거나 Google에 로그인하도록 지시하는 코드를 작성할 필요가 없습니다. Cognito는 즉시 이를 수행합니다.

ID 제공자(예: Facebook)에 인증되면 제공자는 인증 토큰을 제공합니다. 그런 다음 이 인증 토큰은 AWS 환경에 대한 제한된 액세스로 응답하는 cognito에 제공됩니다. IAM 역할에서 이 액세스 권한을 얼마나 제한할지 지정합니다.

Cognito의 역할은 앱과 합법적인 인증자 간의 중개자입니다.

Cognito 사용자 풀 은 애플리케이션에서 등록 및 로그인 기능에 사용되는 사용자 디렉토리입니다. 인증에 성공하면 JSON 웹 토큰이 생성됩니다. 사용자 풀은 사용자 기반임을 기억하십시오. 등록, 복구 및 인증을 처리합니다.

Cognito 자격 증명 풀 은 사용자가 S3 또는 DynamoDB와 같은 직접 AWS 서비스에 임시로 액세스할 수 있도록 하는 데 사용됩니다. 자격 증명 풀이 실제로 들어가서 IAM 역할을 부여합니다.

SAML 기반 인증을 사용하여 IAM이 아닌 사용자의 AWS Management 콘솔 로그인을 허용할 수 있습니다.

특히 SAML(Security Assertion Markup Language)을 구현하는 Microsoft Active Directory도 사용할 수 있습니다.

Amazon Cognito를 사용하여 사용자가 AWS 리소스에 액세스할 수 있도록 권한이 제한된 임시 자격 증명을 애플리케이션에 전달할 수 있습니다.

Amazon Cognito 자격 증명 풀은 인증된 자격 증명과 인증되지 않은 자격 증명을 모두 지원합니다.

인증되지 않은 사용자를 허용하는 경우 또는 사용자를 인증하는 경우 자격 증명 공급자에 로그인 토큰을 설정한 후 즉시 최종 사용자의 고유 Amazon Cognito 식별자(ID)를 검색할 수 있습니다.

모바일 및 데스크톱 앱에 인증을 쉽게 추가해야 하는 경우 Amazon Cognito를 생각해 보십시오.

AWS 리소스 액세스 관리자란 무엇입니까?

AWS Resource Access Manager(RAM)는 AWS 계정이나 AWS 조직 내에서 AWS 리소스를 쉽고 안전하게 공유할 수 있는 서비스입니다. AWS Transit Gateway, 서브넷, AWS License Manager 구성 및 Amazon Route 53 Resolver 규칙 리소스를 RAM과 공유할 수 있습니다.

많은 조직에서 여러 계정을 사용하여 관리 또는 청구 격리를 생성하고 AWS Organizations 서비스의 일부로 오류의 영향을 제한합니다.

RAM을 사용하면 여러 계정에서 중복 리소스를 생성할 필요가 없으므로 소유하고 있는 모든 단일 계정에서 해당 리소스를 관리하는 운영 오버헤드가 줄어듭니다.

다중 계정 환경에서 중앙에서 리소스를 생성하고 RAM을 사용하여 리소스 공유 생성, 리소스 지정, 계정 지정이라는 간단한 세 단계를 통해 계정 간에 해당 리소스를 공유할 수 있습니다.

추가 비용 없이 RAM을 사용할 수 있습니다.

Athena는 무엇입니까?

Athena는 표준 SQL 명령을 사용하여 S3에서 데이터를 쿼리하고 상호 작용할 수 있는 대화형 쿼리 서비스입니다. 이는 일반 개발자를 위한 프로그래밍 방식 쿼리에 유용합니다. 서버리스이며 프로비저닝이 필요하지 않으며 쿼리 및 스캔한 TB당 비용을 지불합니다. 기본적으로 Athena를 사용하여 S3를 SQL 지원 데이터베이스로 전환합니다.

사용 사례 예시:

- ELK 스택에 대한 대안 또는 보완으로 S3 버킷에 덤프되는 쿼리 로그
- 정기적으로 S3에 입력되는 데이터를 기반으로 비즈니스 보고서를 실행하도록 쿼리 설정
- 클릭 스트림 데이터에 대한 쿼리를 실행하여 고객 행동에 대한 추가 통찰력 확보

AWS Macie란 무엇입니까?

Macie를 이해하려면 PII 또는 개인 식별 정보를 이해하는 것이 중요합니다.

- 악용될 수 있는 개인의 신원을 확인하는 데 사용되는 개인 데이터
- 예: 주민등록번호, 전화번호, 집 주소, 이메일 주소, 생년월일, 여권 번호 등

Amazon Macie는 Amazon S3에 저장된 민감한 데이터를 자동으로 검색, 분류 및 보호하여 데이터 손실을 방지하는 ML 기반 보안 서비스입니다. Amazon Macie는 기계 학습을 사용하여 개인 식별 정보(PII) 또는 지적 재산과 같은 민감한 데이터를 인식하고 비즈니스 가치를 할당하며 이 데이터가 저장되는 위치와 조직에서 데이터가 사용되는 방식에 대한 가시성을 제공합니다.

Macie 대시보드, 경고 또는 보고를 통해 탐지 내용을 알 수 있습니다.

Macie는 CloudTrail 로그를 분석하여 누가 민감한 데이터와 상호 작용했는지 확인할 수도 있습니다.

Macie는 비정상적인 데이터 액세스 활동을 지속적으로 모니터링하고 무단 액세스 또는 부주의한 데이터 누출 위험을 감지하면 경고를 전달합니다.

Macie는 민감한 데이터에 실수로 설정되는 전역 액세스 권한을 감지하고, 소스 코드 내부의 API 키 업로드를 감지하고, 민감한 고객 데이터가 규정 준수 표준을 충족하는 방식으로 저장 및 액세스되고 있는지 확인할 수 있습니다.

AWS KMS란 무엇입니까?

AWS Key Management Service(AWS KMS)는 데이터를 암호화하는 데 사용되는 암호화 키를 쉽게 생성하고 제어할 수 있는 관리형 서비스입니다. AWS KMS에서 생성하는 마스터 키는 FIPS 140-2 인증 암호화 모듈로 보호됩니다.

AWS KMS는 사용자가 관리하는 암호화 키로 데이터를 암호화하는 대부분의 다른 AWS 서비스와 통합됩니다. 또한 AWS KMS는 AWS CloudTrail과 통합되어 감사, 규제 및 규정 준수 요구 사항을 충족하는 데 도움이 되는 암호화 키 사용 로그를 제공합니다.

디스크에 저장하기 전에 KMS API를 사용하여 모든 데이터를 암호화하도록 애플리케이션을 구성할 수 있습니다.

AWS Secrets Manager란 무엇입니까?

AWS Secrets Manager는 보안 암호를 더 쉽게 관리할 수 있도록 해주는 AWS 서비스입니다.

비밀은 데이터베이스 자격 증명, 암호, 타사 API 키 및 임의의 텍스트일 수 있습니다. Secrets Manager 콘솔, Secrets Manager 명령줄 인터페이스(CLI) 또는 Secrets Manager API 및 SDK를 사용하여 이러한 비밀에 대한 액세스를 중앙에서 저장하고 제어할 수 있습니다.

과거에는 데이터베이스에서 정보를 검색하는 사용자 지정 응용 프로그램을 만들 때 일반적으로 응용 프로그램에 직접 데이터베이스에 액세스하기 위한 자격 증명(비밀)을 포함해야 했습니다. 자격 증명을 교체할 때가 되었을 때 새 자격 증명을 만드는 것 이상의 작업을 수행해야 했습니다. 새 자격 증명을 사용하도록 애플리케이션을 업데이트하는 데 시간을 투자해야 했습니다. 그런 다음 업데이트된 애플리케이션을 배포해야 했습니다. 자격 증명을 공유하는 여러 응용 프로그램이 있고 그 중 하나를 업데이트하지 못한 경우 응용 프로그램이 중단됩니다.

이러한 위험 때문에 많은 고객이 자격 증명을 정기적으로 교체하지 않기로 선택하여 하나의 위험을 다른 위험(기능 대 보안)으로 효과적으로 대체합니다.

Secrets Manager를 사용하면 코드의 하드 코딩된 자격 증명(암호 포함)을 Secrets Manager에 대한 API 호출로 교체하여 프로그래밍 방식으로 암호를 검색할 수 있습니다.

이것은 비밀이 단순히 존재하지 않기 때문에 코드를 검사하는 누군가에 의해 비밀이 손상되는 것을 방지하는 데 도움이 됩니다.

또한 지정한 일정에 따라 암호를 자동으로 교체하도록 Secrets Manager를 구성할 수 있습니다. 이를 통해 장기 비밀을 단기 비밀로 교체할 수 있으므로 손상 위험을 크게 줄이는 데 도움이 됩니다.

AWS STS란 무엇입니까?

AWS Security Token Service(AWS STS)는 AWS 리소스에 대한 액세스를 제어할 수 있는 임시 보안 자격 증명을 생성하고 신뢰할 수 있는 사용자에게 제공하는 데 사용할 수 있는 서비스입니다.

임시 보안 자격 증명은 IAM 사용자가 사용할 수 있는 장기 액세스 키 자격 증명과 거의 동일하게 작동합니다.

임시 보안 자격 증명은 이름에서 알 수 있듯이 단기적입니다. 몇 분에서 몇 시간 동안 지속되도록 구성할 수 있습니다. 자격 증명이 만료되면 AWS는 더 이상 자격 증명을 인식하지 않거나 자격 증명으로 수행된 API 요청에서 모든 종류의 액세스를 허용하지 않습니다.

OpsWorks란?

AWS OpsWorks는 Chef 및 Puppet의 관리형 인스턴스를 제공하는 구성 관리 서비스입니다. Chef 및 Puppet은 코드를 사용하여 서버 구성을 자동화할 수 있는 자동화 플랫폼입니다.

OpsWorks를 사용하면 Chef 및 Puppet을 사용하여 Amazon EC2 인스턴스 또는 온프레미스 컴퓨팅 환경에서 서버를 구성, 배포 및 관리하는 방법을 자동화할 수 있습니다.

OpsWorks에는 Chef Automate용 AWS Opsworks, Puppet Enterprise용 AWS OpsWorks 및 AWS OpsWorks Stacks의 세 가지 제품이 있습니다.

AWS OpsWorks Stacks를 사용하면 AWS 및 온프레미스에서 애플리케이션과 서버를 관리할 수 있습니다. OpsWorks Stacks를 사용하면 로드 밸런싱, 데이터베이스 및 애플리케이션 서버와 같은 다양한 계층을 포함하는 스택으로 애플리케이션을 모델링할 수 있습니다.

OpsWorks Stacks는 각 계층에서 Amazon EC2 인스턴스를 배포 및 구성하거나 Amazon RDS 데이터베이스와 같은 다른 리소스에 연결할 수 있을 만큼 충분히 복잡합니다.

Elastic Transcoder란 무엇입니까?

클라우드의 미디어 트랜스코더. 기본적으로 미디어 파일을 원래 형식에서 휴대폰, 태블릿, PC 등에 대해 지정된 미디어 형식으로 변환하는 서비스입니다.

다양한 미디어 유형에 대한 기본 제공 지원으로 인해 결과 품질이 좋을 것이라고 신뢰할 수 있습니다.

Elastic Transcoder를 사용하면 트랜스코딩 작업과 완료된 작업의 해상도에 대해 분당 비용을 지불합니다.

AWS 디렉터리 서비스란 무엇입니까?

AWS Directory Service는 Amazon Cloud Directory 및 Microsoft Active Directory(AD)를 다른 AWS 서비스와 함께 사용할 수 있는 다양한 방법을 제공합니다.

디렉터리는 사용자, 그룹 및 장치에 대한 정보를 저장하고 관리자는 이를 사용하여 정보 및 리소스에 대한 액세스를 관리합니다.

AWS Directory Service는 클라우드에서 기존 Microsoft AD 또는 LDAP(Lightweight Directory Access Protocol) 인식 애플리케이션을 사용하려는 고객에게 다양한 디렉터리 선택을 제공합니다. 또한 사용자, 그룹, 장치 및 액세스를 관리하기 위해 디렉토리가 필요한 개발자에게 동일한 선택을 제공합니다.

IoT 코어란?

AWS IoT Core는 연결된 디바이스가 클라우드 애플리케이션 및 기타 디바이스와 쉽고 안전하게 상호 작용할 수 있도록 하는 관리형 클라우드 서비스입니다.

AWS IoT Core는 다양한 종류의 연결된 디바이스와 위치에서 보안 통신 및 데이터 처리를 제공하므로 IoT 애플리케이션을 쉽게 구축할 수 있습니다.

AWS WorkSpaces란 무엇입니까?

Amazon WorkSpaces는 안전한 관리형 DaaS(Desktop-as-a-Service) 솔루션입니다. Amazon WorkSpaces를 사용하여 단 몇 분 만에 Windows 또는 Linux 데스크톱을 프로비저닝하고 전 세계 작업자에게 수천 대의 데스크톱을 제공하도록 신속하게 확장할 수 있습니다.

Amazon WorkSpaces는 하드웨어 인벤토리, OS 버전 및 패치, 그리고 데스크톱 제공 전략을 단순화하는 데 도움이 되는 가상 데스크톱 인프라(VDI) 관리의 복잡성을 제거하는 데 도움이 됩니다.

Amazon WorkSpaces를 사용하면 지원되는 모든 디바이스에서 언제 어디서나 액세스할 수 있는 빠르고 응답성이 뛰어난 데스크톱을 사용자가 선택할 수 있습니다.

AWS Fargate란 무엇입니까?

AWS Fargate는 컨테이너용 서버리스 컴퓨팅 엔진입니다.

Fargate 시작 유형을 사용하면 백엔드 인프라를 프로비저닝하고 관리할 필요 없이 컨테이너화된 애플리케이션을 실행할 수 있습니다. 작업 정의를 등록하기만 하면 Fargate가 컨테이너를 시작합니다.

Amazon Elastic Container Service(ECS) 및 Amazon Elastic Kubernetes Service(EKS)와 함께 작동합니다.

Fargate를 사용하면 애플리케이션 구축에 집중할 수 있습니다. 서버를 프로비저닝 및 관리할 필요가 없고 애플리케이션별로 리소스를 지정하고 비용을 지불할 수 있으며 설계에 따른 애플리케이션 격리를 통해 보안이 향상됩니다.

Amazon Elastic Container Service란 무엇입니까?

Amazon Elastic Container Service(Amazon ECS)는 완전 관리형 컨테이너 오케스트레이션 서비스입니다.

Amazon ECS를 사용하면 자체 클러스터 관리 인프라를 설치, 운영 및 확장할 필요가 없습니다. 간단한 API 호출로 컨테이너 지원 애플리케이션을 시작 및 중지하고, 클러스터의 전체 상태를 쿼리하고, 보안 그룹, Elastic Load Balancing, EBS 볼륨 및 IAM 역할과 같은 친숙한 기능에 액세스할 수 있습니다.

Amazon ECS를 사용하여 리소스 요구 사항 및 가용성 요구 사항에 따라 클러스터 전체에 컨테이너 배치를 예약할 수 있습니다. 또한 고유한 스케줄러 또는 타사 스케줄러를 통합하여 비즈니스 또는 애플리케이션 특정 요구 사항을 충족할 수 있습니다.

컨테이너용 서버리스 컴퓨팅인 AWS Fargate를 사용하여 ECS 클러스터를 실행하도록 선택할 수 있습니다. Fargate를 사용하면 서버를 프로비저닝 및 관리할 필요가 없고, 애플리케이션별로 리소스를 지정하고 비용을 지불할 수 있으며, 설계에 따른 애플리케이션 격리를 통해 보안이 향상됩니다.

Amazon Elastic Kubernetes Service란 무엇입니까?

Amazon Elastic Kubernetes Service(Amazon EKS)는 완전 관리형 Kubernetes 서비스입니다. EKS는 업스트림 Kubernetes를 실행하고 Kubernetes 인증을 받았으므로 커뮤니티에서 제공하는 오픈 소스 도구의 모든 이점을 활용할 수 있습니다. 또한 코드를 리팩토링할 필요 없이 표준 Kubernetes 애플리케이션을 EKS로 쉽게 마이그레이션할 수 있습니다.

Kubernetes는 컨테이너화된 애플리케이션을 대규모로 배포하고 관리할 수 있는 오픈 소스 소프트웨어입니다. Kubernetes는 관리 및 검색 가능성을 위해 컨테이너를 논리적 그룹으로 그룹화한 다음 EC2 인스턴스 클러스터에서 시작합니다. Kubernetes를 사용하면 온프레미스 및 클라우드에서 동일한 도구 세트를 사용하여 마이크로서비스, 일괄 처리 작업자 및 PaaS(Platform as a Service)를 포함한 컨테이너화된 애플리케이션을 실행할 수 있습니다.

Amazon EKS는 고가용성 및 내결함성을 위해 여러 AWS 가용 영역에서 API 서버 및 백엔드 지속성 계층을 포함한 Kubernetes 제어 평면을 프로비저닝하고 확장합니다. Amazon EKS는 비정상 제어 플레인 노드를 자동으로 감지 및 교체하고 제어 플레인에 대한 패치를 제공합니다.

Amazon EKS가 없으면 Kubernetes 제어 평면과 작업자 노드 클러스터를 모두 직접 실행해야 합니다. Amazon EKS를 사용하면 EKS 콘솔, CLI 또는 API에서 단일 명령을 사용하여 작업자 노드를 프로비저닝하고 AWS는 가용성이 높고 안전한 구성에서 Kubernetes 제어 플레인의 프로비저닝, 확장 및 관리를 처리합니다. 이를 통해 Kubernetes 실행에 대한 상당한 운영 부담을 제거하고 AWS 인프라를 관리하는 대신 애플리케이션 구축에 집중할 수 있습니다.

컨테이너용 서버리스 컴퓨팅인 AWS Fargate를 사용하여 EKS를 실행할 수 있습니다. Fargate를 사용하면 서버를 프로비저닝 및 관리할 필요가 없고, 애플리케이션별로 리소스를 지정하고 비용을 지불할 수 있으며, 설계에 따른 애플리케이션 격리를 통해 보안이 향상됩니다.

Amazon EKS는 많은 AWS 서비스와 통합되어 애플리케이션에 확장성과 보안을 제공합니다. 이러한 **서비스에는 로드 분산을 위한 Elastic Load Balancing, 인증을 위한 IAM, 격리를 위한 Amazon VPC 및 로깅을 위한 AWS CloudTrail이 포함됩니다.**

Pilot light(파일럿 라이트)는 무엇을 의미합니까?

파일럿 라이트라는 용어는 최소 버전의 환경이 항상 클라우드에서 실행되는 재해 복구 시나리오를 설명하는 데 자주 사용됩니다.

파일럿 라이트의 아이디어는 가스 히터에서 나온 비유입니다. 가스 히터에서 항상 켜져 있고 신속하게 전체 용광로를 점화하여 집을 데울 수 있는 작은 불꽃. 이 시나리오는 백업 및 복원 시나리오와 유사합니다.

예를 들어 AWS를 사용하면 AWS에서 시스템의 가장 중요한 핵심 요소를 구성하고 실행하여 파일럿 라이트를 유지할 수 있습니다. 복구 시기가 되면 항상 실행 중인 핵심 코어를 중심으로 본격적인 프로덕션 환경을 신속하게 프로비저닝할 수 있습니다.

Blue-Green deployments(블루-그린 배포)란 무엇입니까?

배포 자동화의 과제 중 하나는 테스트의 마지막 단계에서 실제 프로덕션으로 전환하는 것입니다. 가동 중지 시간을 최소화하려면 일반적으로 이 작업을 빠르게 수행해야 합니다.

Blue-Green 배포 접근 방식은 가능한 한 동일한 두 개의 프로덕션 환경을 갖도록 하여 이를 수행합니다. 예를 들어 파란색이라고 하면 언제든지 그 중 하나가 라이브입니다. 소프트웨어의 새 릴리스를 준비하면서 친환경 환경에서 테스트의 마지막 단계를 수행합니다. 소프트웨어가 녹색 환경에서 작동하면 들어오는 모든 요청이 녹색 환경으로 이동하도록 라우터를 전환합니다. 파란색 환경은 이제 유휴 상태입니다.

블루-그린 배포는 또한 롤백을 위한 빠른 방법을 제공합니다. 문제가 발생하면 라우터를 블루 환경으로 다시 전환합니다.

CloudFormation 및 CodeDeploy(Jenkins의 AWS 버전)는 모두 이 배포 기술을 지원합니다.

Amazon 데이터 수명 주기 관리자란 무엇입니까?

Amazon Data Lifecycle Manager(Amazon DLM)를 사용하여 Amazon EBS 볼륨을 백업하기 위해 생성된 스냅샷의 생성, 보존 및 삭제를 자동화할 수 있습니다.

스냅샷 관리를 자동화하면 다음과 같은 이점이 있습니다.

- 정기적인 백업 일정을 시행하여 소중한 데이터를 보호하십시오.
- 감사자 또는 내부 규정 준수의 요구에 따라 백업을 유지합니다.
- 오래된 백업을 삭제하여 스토리지 비용을 줄입니다.

Amazon DLM을 사용하면 더 이상 EBS 스냅샷을 찍는 것을 기억할 필요가 없으므로 엔지니어의 인지 부하가 줄어듭니다.

Route Origin Authorization이란 무엇입니까?

온프레미스 네트워크에서 AWS 계정으로 퍼블릭 IPv4 주소 범위의 일부 또는 전체를 가져올 수 있습니다. 계속해서 주소 범위를 소유하지만 AWS는 이를 인터넷에 광고합니다. 주소 범위를 AWS로 가져오면 계정에 주소 풀로 나타납니다.

그런 다음 주소 풀에서 탄력적 IP 주소를 생성하고 이를 EC2 인스턴스, NAT 게이트웨이 및 Network Load Balancer와 같은 AWS 리소스와 함께 사용할 수 있습니다. 이를 "BYOIP(Bring Your Own IP Addresses)"라고도 합니다.

자신만 AWS 계정으로 주소 범위를 가져올 수 있도록 하려면 Amazon이 주소 범위를 알리고 주소 범위를 소유하고 있다는 증거를 제공하도록 승인해야 합니다.

ROA의 이점은 파트너와 고객이 IP 주소 화이트리스트를 변경할 필요 없이 기존 애플리케이션을 AWS로 마이그레이션할 수 있다는 것입니다.

Amazon MQ란 무엇입니까?

Amazon MQ는 클라우드에서 메시지 브로커를 쉽게 설정하고 운영할 수 있는 관리형 메시지 브로커 서비스입니다.

이 서비스는 Amazon SQS와 다른 점인 온프레미스에서 클라우드로 서비스와 앱을 마이그레이션할 때 사용됩니다.

Amazon MQ는 Amazon EFS에서 지원하는 내구성 최적화 브로커를 지원하여 고가용성 및 메시지 내구성을 지원하고 Amazon EBS가 지원하는 처리량 최적화 브로커를 지원하여 짧은 지연 시간과 높은 처리량이 필요한 대용량 애플리케이션을 지원합니다.

애플리케이션에서 메시징 코드를 다시 작성할 필요가 없기 때문에 모든 메시지 브로커에서 Amazon MQ로 쉽게 이동할 수 있습니다.

Amazon MQ는 온프레미스 또는 클라우드에서 직접 메시지 브로커를 관리하고 애플리케이션에서 메시징 코드를 다시 작성하지 않고 완전 관리형 클라우드 서비스로 전환하려는 엔터프라이즈 IT 전문가, 개발자 및 설계자에게 적합합니다.

AWS Config란 무엇입니까?

AWS Config는 AWS 리소스의 구성을 평가, 감사 및 평가할 수 있는 서비스입니다. Config는 AWS 리소스 구성을 지속적으로 모니터링하고 기록하며, 이를 통해 원하는 구성에 대해 기록된 구성 평가를 자동화할 수 있습니다.

Config를 사용하면 AWS 리소스 간의 관계 및 구성 변경 사항을 검토하고, 자세한 리소스 구성 기록을 살펴보고, 내부 지침에 지정된 구성에 대한 전반적인 규정 준수를 결정할 수 있습니다. 이를 통해 규정 준수 감사, 보안 분석, 변경 관리 및 운영 문제 해결을 단순화할 수 있습니다.

AWS Config를 사용하면 다음을 수행할 수 있습니다. ·

- 원하는 설정에 대한 AWS 리소스 구성을 평가합니다. ·
- AWS 계정과 연결된 지원되는 리소스의 현재 구성에 대한 스냅샷을 가져옵니다. ·
- 계정에 있는 하나 이상의 리소스 구성을 검색합니다. ·
- 하나 이상의 리소스에 대한 기록 구성을 검색합니다. ·
- 리소스가 생성, 수정 또는 삭제될 때마다 알림을 받습니다.
- 리소스 간의 관계를 봅니다. 예를 들어 특정 보안 그룹을 사용하는 모든 리소스를 찾고 싶을 수 있습니다.