

AWS를 통한 비용 최적화

아키텍처, 도구, 모범 사례

2016년 2월



© 2016, Amazon Web Services, Inc. 또는 계열사. All rights reserved.

고지 사항

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

목차

요약	3
소개	4
비용 최적화	4
설계 원칙	5
공급과 수요의 균형	6
컴퓨팅	6
스토리지	11
애플리케이션 서비스	13
비용 효율적인 리소스	18
세분화된 접근 방식	18
비용 인지	19
AWS 가격표 API	20
세부 결제 보고서	20
시간에 따른 최적화	21
결론	21
기고자	22
부록: 비용 효율적인 스토리지의 예	22
참고 문헌	23

요약

본 문서는 [Well-Architected 프레임워크](#)의 비용 최적화 기반을 중점적으로 다룹니다. 또한 고객이 모범 사례를 적용해 AWS 환경을 설계, 제공 및 유지 관리할 수 있도록 지침을 제공합니다.

소개

AWS는 고객이 클라우드에서 안정적이고, 안전하고, 효율적이며, 경제적인 시스템을 설계할 수 있도록 하려면 아키텍처 모범 사례를 고객과 공유해야 한다는 점을 잘 알고 있습니다. 이러한 노력의 일환으로 개발된 [AWS Well-Architected 프레임워크](#)를 이용하면 AWS에서 시스템을 구축하면서 내리게 되는 결정의 장점과 단점을 이해할 수 있습니다. Well-Architected 시스템이 사업의 성공 가능성을 대폭 높일 것으로 기대합니다.

프레임워크의 네 가지 기반:

- 보안
- 안정성
- 성능 효율성
- 비용 최적화

비용 최적화 기반을 활용하면 불필요한 비용 또는 최선이 아닌 리소스 사용을 피하거나 제거할 수 있고, 이러한 절감을 통해 비즈니스를 차별화할 수 있습니다. 비용 최적화 시스템을 사용하면 비용을 최대한 절감할 수 있는 동시에 비즈니스 목표를 달성하고, 프레임워크의 다른 기반에 필요한 요구 사항을 초과 충족할 수 있습니다. 이 문서는 적절한 아키텍처를 선택하고 AWS 리소스를 최대한 효율적으로 활용하는 데 필요한 구체적인 모범 사례 지침을 제공합니다.

비용 최적화

비즈니스 문제는 운영적인 성격을 띠고(지원 팀 대기), 낭비를 유발하며(하드웨어 리소스 오버프로비저닝), 실패할 가능성이 높고(긴밀하게 결합된 배포), 큰 비용이 발생할 수 있습니다(데이터 센터 운영, 보안 직원 배치 등). AWS가 이러한 문제에 대한 해결책을 제시하지만, 비용 최적화를 염두에 두고 서비스를 이해하고 구현하는 것이 중요합니다.

AWS는 비용 최적화를 다음 네 가지 영역으로 분류합니다.

- 공급과 수요의 균형
- 비용 효율적인 리소스
- 비용 인지
- 시간에 따른 최적화

보안 개선이나 출시 시기 단축과 더불어, 비용 및 투자수익률(ROI)이 구매 결정에서 주로 고려해야 하는 요소입니다. AWS는 [AWS Trusted Advisor](#), [상세 결제 보고서](#), [총 소유 비용\(TCO\) 계산기](#) 등, 청구서를 더 잘 이해하고 소유 비용을 계산하며 개선이 필요한 영역을 파악하는 데 활용할 수 있는 여러 가지 도구를 지원합니다. 의사 결정 프로세스에 도움이 될 정량적 지원의 역할을 하는 AWS 사례 연구 및 참조 자료도 있습니다.

설계 원칙

AWS는 비용 최적화의 네 개 영역에서 모범 사례를 논할 때 다음과 같은 설계 원칙을 염두에 둡니다.

- **예측—조달 모델에서 소비 모델로 전환:** 기록 데이터나 예측 추정치를 근거로 데이터 센터와 서버에 과도하게 투자하는 대신 실제로 소비하는 컴퓨팅 리소스에 대해서만 지불하고, 정교한 예측이 아닌 실제 수요에 따라 사용량을 증감합니다. 예를 들어 개발 및 테스트 환경은 대체로 주중에 하루 여덟 시간 동안만 사용됩니다. 이러한 리소스를 사용하지 않는 동안 중지하면 비용 절감 효과는 최대 75%에 달합니다(40시간 vs. 168시간).
- **규모의 경제에 따른 이점:** AWS 클라우드를 사용하는 고객이 수십만 명에 달하므로 보다 저렴한 종량 요금제를 이용할 수 있습니다.
- **데이터 센터 운영에 필요한 비용 제거:** 클라우드 컴퓨팅 서버를 랙에 설치하고, 쌓아 올리고, 서버에 전원을 공급하는 등의 과중한 업무를 공급업체에서 처리하므로 IT 인프라에 신경 쓸 필요 없이 고객과 핵심 비즈니스에만 집중할 수 있습니다.

- **비용을 투명하게 부과:** 클라우드를 사용하면 시스템의 비용을 손쉽게 확인하고, 비즈니스 사용자별로 IT 비용을 부과할 수 있습니다. 이를 통해 ROI를 측정하고, 개별 비즈니스 사용자들이 리소스를 최적화하고 비용을 절감하도록 장려할 수 있습니다.
- **관리형 서비스를 사용하여 소유 비용 절감:** 클라우드에서 관리형 서비스는 이메일 전송이나 데이터베이스 관리 같은 작업을 위해 서버를 유지관리하는 운영상의 부담을 없애줍니다. 그리고 관리형 서비스는 클라우드 규모에서 운영되기 때문에 트랜잭션 또는 서비스당 비용이 저렴합니다.
- **설계 결정에 대한 지속적인 재평가:** 하드웨어와 소프트웨어에 거액의 자본을 투자해야 했던 기존의 IT 인프라 접근 방식과 달리, AWS는 대부분의 서비스에 종량 요금제를 실시합니다. 즉, 프로젝트 수명 주기를 시작하는 설계 단계에 결정을 내릴 필요가 없다는 뜻입니다. 따라서 오버프로비저닝이 발생하거나 예기치 못한 수요에 대응하지 못할 위험이 줄어듭니다. 설계 결정을 지속적으로 재평가할 수 있습니다. 또한 새로운 AWS 제품 사용을 탐색하여 효율성이 더욱 높아지고 있는지 확인할 수도 있습니다.

공급과 수요의 균형

기존의 IT 인프라 모델과 달리 AWS는 처음부터 탄력적인 온디맨드 방식을 지원합니다. AWS는 필요에 따라 프로그래밍 방식으로 수직 확장 및 축소하거나, 자동으로 스토리지 객체를 아카이브하거나 만료시키는 수명 주기 규칙을 구현할 수 있는 메커니즘을 제공합니다. 비용 최적화된 아키텍처를 구현하기 원한다면 이러한 기능과 서비스 사용을 고려하십시오.

컴퓨팅

AWS 에코시스템에서 컴퓨팅은 Amazon Elastic Compute Cloud(Amazon EC2)로 시작됩니다. 가장 기본적인 수준에서 EC2는 클라우드 내 가상 머신입니다.

다음 그림에서 보듯이 Amazon EC2는 2006년부터 발전하고 있으며 Auto Scaling, Amazon EC2 Container Service(효율성 증대), Elastic Load Balancing(복원성), AWS Lambda(서버리스 컴퓨팅) 등 다른 서비스의 등장으로 이어졌습니다.



Amazon EC2

가장 적합한 인스턴스 유형으로 애플리케이션이나 워크로드의 요구 사항을 충족할 수 있어야만 비용 최적화에 성공할 수 있습니다. Amazon EC2 [인스턴스 유형 및 크기](#)¹는 다양한 조합의 CPU, 메모리, 스토리지, 네트워크 용량으로 제공되므로 애플리케이션에 맞는 리소스 구성을 유연하게 선택할 수 있습니다. 예를 들어 C4 인스턴스는 대용량 컴퓨팅 리소스를 요하는 워크로드에 걸맞습니다. M4 인스턴스는 다중 역할 인스턴스 유형입니다. G2 인스턴스는 전용 그래픽 프로세서가 필요한 워크로드를 위한 것입니다. 적합한 인스턴스 유형을 선택하면 비용 효율적인 방식으로 성능을 최적화할 수 있습니다.

현재 워크로드의 요구 사항을 이해한 후, 각각 30% 또는 최대 90%의 EC2 비용 절감 효과를 보장하는 예약 인스턴스와 스팟 인스턴스의 장점을 고려하시기 바랍니다.

벤치마킹

AWS는 워크로드에 가장 근접한 인스턴스 패밀리에서 인스턴스 유형을 선택한 후 벤치마킹 과정을 시작할 것을 권장합니다. 인스턴스 유형을 변경할 수 있기 때문에 각 유형별로 벤치마킹을 수행할 수 있습니다. AWS가 새로운 인스턴스 유형을 출시하는 경우에도 마찬가지입니다.

¹ 인스턴스 유형의 실행 비용(시간당)은 각기 다릅니다. 일반적으로 인스턴스 유형이 클수록 비용이 높아집니다.

예약 인스턴스

인스턴스 유형을 정한 후에는 예약 인스턴스를 구입할 수 있는 옵션이 주어집니다. 이는 특정 AWS 리전에서 사전 약정으로 용량을 구입하는 것으로, 실행 비용을 크게 감축할 수 있습니다.

예약 인스턴스는 요금 항목 중 하나입니다. 이미 선택하여 해당 인스턴스 유형을 구입한 가용 영역²(AZ)에서 사용할 수 있는 용량이 주어지며 시간당 요금을 크게 줄일 수 있습니다. 예약 인스턴스는 연중무휴 24시간 리소스로 사용될 뿐만 아니라, 워크로드가 시간에 종속된 경우에는 예약 인스턴스를 결합할 수도 있습니다.

예를 들어 **m4.large** 같은 다용도 인스턴스 유형의 예약 인스턴스가 있다고 가정해봅시다. 이 인스턴스는 일일 총 9시간(오전 8:00 - 오후 5:00)의 근무 시간 동안만 실행하면 됩니다. 하지만 이 가용 영역에서 동일한 인스턴스 유형을 사용할 수 있는 또 하나의 워크로드는 근무 시간 이후에(오후 5:00 - 오전 8:00) 실행됩니다. 동일한 인스턴스 유형(**m4.large**)을 선택하고, 주간 인스턴스가 중단된 후 해당 인스턴스에서 야간 워크로드를 시작할 수 있습니다.

첫 번째 인스턴스가 중단된 후 예약 인스턴스의 시간당 요금이 근무 시간 이후 인스턴스에 적용되므로 전체 비용 효율성이 극대화됩니다.

워크로드와 인스턴스 유형은 시간 경과에 따라 변화하기 때문에 인스턴스 선택을 지속적으로 재평가하는 것이 중요합니다. 예약 인스턴스는 현재 1년 또는 3년 약정³으로 제공되며 예약 인스턴스 약정이 만료되기 전에 요구 사항이 변경될 수 있습니다. 이 경우, EC2 Container Service(Amazon ECS)를 이용해 인스턴스 사용량을 늘리거나 [AWS 예약 인스턴스 Marketplace](#)에서 예약 인스턴스를 판매할 수 있습니다.

² <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regionsavailability-zones.html#concepts-regions-availability-zones>

³ <https://aws.amazon.com/ec2/purchasing-options/reserved-instances/>

Auto Scaling

Auto Scaling은 사용자가 정의하는 조건에 따라 [Amazon EC2](#) 용량을 자동으로 확장하거나 축소합니다. 예를 들어 기존에는 IT 배포 시 피크 사용량을 기준으로 하드웨어 요구 사항을 결정해야 했습니다. 예상되는 가장 로드 기간에 따라 컴퓨팅 리소스를 결정해야 하고 이후에 수직 축소가 불가능하다면 로드가 최저 수준인 동안에는 컴퓨팅 리소스를 낭비하게 됩니다. **Auto Scaling**은 리소스 수요 변화에 따라 인스턴스 수를 늘리거나 줄여 효율성과 비용을 최적화합니다.

워크로드가 영업 시간(오전 9:00 - 오후 5:00) 중에 실행된다면, **Auto Scaling**이 예상 로드와 적합한 인스턴스를 시작하도록 구성할 수 있습니다. 영업 시간 이후에는 **Auto Scaling**이 인스턴스 수를 줄여 시간당 실행 비용에 대한 지출을 최소화할 수 있습니다.

스팟 인스턴스

스팟 인스턴스를 사용하면 고객이 예비 **Amazon EC2** 컴퓨팅 용량에 입찰할 수 있습니다. 스팟 인스턴스는 온디맨드 가격에서 할인되는 경우가 많기 때문에 애플리케이션 운영 비용을 크게 줄이는 동시에 애플리케이션의 컴퓨팅 용량과 처리량을 늘릴 수 있습니다.

스팟 시장을 활용하면⁴ 영구적인 상태가 필요하지 않은 워크로드의 운영 비용을 줄일 수 있습니다. 이 유형의 모델에 포함되는 워크로드에는 **Amazon Elastic MapReduce(Amazon EMR)** 클러스터가 있습니다. 스팟 인스턴스를 이용해 추가 코어 노드로 온디맨드 클러스터를 보강하고, **Amazon EMR** 하둡 작업을 완료하는데 필요한 시간을 줄일 수 있습니다.

입찰 가격이 최고 입찰가를 초과하면 스팟 인스턴스로 실행 중인 코어 노드의 종료 시간이 예약되었다는 알림을 받게 됩니다. 이 경우, 하둡이 클러스터 내에 남은 가용 인스턴스에서 실행 중인 작업을 자동으로 다시 시작합니다.

스팟 인스턴스를 비영구적으로 사용한 또 하나의 사례는 **BatchProcessing** 작업입니다. 이러한 작업은 흔히 중단 시에도 처리를 계속 진행합니다(이러한 로직이 포함된 경우). 적정 입찰 가격의 스팟 인스턴스를 사용할 수 있는 경우 야간 처리 작업이 시작될 수 있습니다. 이러한 인스턴스가 종료될 경우, 스팟 인스턴스를 다시 사용할 수 있게 되었을 때 작업이 재개될 수 있습니다.

⁴ <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/how-spot-instanceswork.html>

Amazon ECS

Amazon EC2 Container Service(Amazon ECS)는 Docker 컨테이너를 호스팅할 EC2 인스턴스에 플랫폼을 제공합니다. 그러면 워크로드가 좀 더 세분화되고 EC2 인스턴스의 효율성이 높아집니다. 이는 다수의 EC2 인스턴스에서 애플리케이션이나 인프라 구성 요소를 실행하는 대신, 여러 컨테이너를 호스팅할 수 있는 대형 인스턴스 유형을 사용하는 경우에 특히 유용합니다.

비용 최적화의 관점에서 볼 때, 잘 활용되지 않는 예약 인스턴스에서도 컨테이너와 함께 Amazon ECS를 사용할 수 있습니다.

m4.xlarge 예약 인스턴스를 사용하지만 CPU 리소스의 5%만 소비한다고 가정해봅시다. 이 경우 인스턴스에 여러 개의 컨테이너를 추가하고 활용률을 더 효율적인 수준으로 끌어올릴 수 있기 때문에 Amazon ECS의 사용 사례로 적합합니다. Amazon ECS는 일정에 따라 인스턴스에 컨테이너를 배치하는 데 사용하기도 합니다. 예약 인스턴스 예에서 확장을 원할 경우, 원래 인스턴스의 전원을 차단하기보다는 컨테이너가 오후 6:00부터 오전 8:00까지 인스턴스에 배치되도록 예약하여 예약 인스턴스 시간당 요금으로 사용할 수 있습니다.

AWS Lambda

항상 가동되는 인스턴스를 실행할 경우, 특히 서비스가 어떤 작업을 대기하는 동안 낭비될 수 있습니다. 예컨대 이와 같은 예전 방식으로 사진 처리 애플리케이션을 만들면 리스너 또는 폴링 서비스를 통해 수신 작업을 찾아낸 후 뭔가 업로드되면 처리 코드를 시작할 것입니다. 서버(인스턴스)를 계속 실행하고 애플리케이션 코드에 컴퓨팅 플랫폼을 제공하기 위해 다수의 인스턴스를 결정한 후 유지해야 합니다. Lambda를 사용하면 코드를 쓰고 이를 시작할 트리거 이벤트를 설정하기만 하면 됩니다.

Lambda는 서버리스 컴퓨팅입니다. Lambda 서비스는 코드에 필요한 컴퓨팅 플랫폼을 제공하고 제공된 파라미터를 기반으로 이벤트를 트리거합니다. 따라서 고유한 실행 인스턴스가 필요하지 않고 운영 체제 관리 오버헤드가 발생하지 않습니다.

Lambda는 업로드, 느슨하게 분리할 수 있는 서비스 요청 등의 이벤트에 응답하는 워크로드에 적절한 선택입니다. 또한 Lambda는 100밀리초 단위로 측정되기 때문에 정교합니다.

아키텍처 계획 및 지속적인 개정과 평가를 통해 비용 최적화를 달성할 수 있습니다. 가상화된 인스턴스에서 서버리스 컴퓨팅까지 AWS 서비스를 사용하면 설계 결정은 영구 불변한 것이 아닙니다. 뿐만 아니라, 새롭게 출시된 제품과 서비스를 평가하여 기존 배포를 지속적으로 최적화할 수 있습니다.

스토리지

스토리지 솔루션을 고려할 때 특히 중요한 요소는 이와 같습니다.

- 내구성
- 가용성
- 규제 및 거버넌스 요구 사항
- 보안
- 기능 요구 사항

Amazon EBS
Amazon EC2 내에서 사용하기 위한 블록 스토리지



Amazon S3
API를 통한 인터넷급 스토리지



이미지
비디오
파일
바이너리
스냅샷

Amazon Glacier

아카이브 및 백업용 스토리지



이미지
비디오
파일
바이너리
스냅샷

AWS 스토리지 솔루션은 다양한 기술 및 가격 요구 사항을 충족할 수 있습니다. 가격 최적화 관점에서 최저 컴퓨팅 관련 비용 스토리지 옵션은 로컬에서 연결된 휘발성 스토리지⁵입니다. 이는 EC2 인스턴스의 가동률에 포함된 스토리지입니다. 휘발성 스토리지의 로컬 처리 속성은 장기적인 영구 데이터 스토리지에 부적합합니다. 영구 스토리지를 위해 Amazon Elastic Block Store⁶(Amazon EBS)를 사용하면 여러 인스턴스에 마이그레이션할 수 있는 블록 수준의 디바이스에 EC2 인스턴스가 기록할 수 있습니다. 또한 이러한 볼륨을 백업하고 마이그레이션할 수 있는 스냅샷 기능도 제공합니다.

EC2 인스턴스에 사용할 수 있는 스토리지 유형은 두 가지입니다. 하나는 로그 파일 및 중요하지 않은 데이터에 적합한 비영구 또는 휘발성 스토리지이고, 다른 하나는 지속성을 요하는 애플리케이션과 워크로드에 사용할 영구 스토리지입니다.

각 옵션은 성능과 비용 면에서 서로 다른 장점과 단점이 있습니다. 휘발성 스토리지는 EC2 인스턴스 시간당 요금에 포함됩니다. Amazon EBS는 추가로 구입해야 합니다. 최초 평가 기간 동안 워크로드별로 이러한 장점과 단점을 고려하고, 워크로드를 일정 시간 실행한 후 다시 한 번 평가해야 합니다. 이렇게 하면 할당된 리소스를 제대로 활용 중인지 확인하고 필요에 따라 조정할 수 있습니다. 예를 들어 워크로드가 “버스팅” IOPS 수준에서 더 효율적으로 실행되는 경우 Amazon EBS 볼륨 유형⁷ 중 프로비저닝된 IOPS에서 GP2 인스턴스로 전환할 수 있습니다. AWS는 연결되지 않은 Amazon EBS 볼륨과 기타 중요 배포 정보를 강조해서 보여주는 [Trusted Advisor](#)를 통해 고객에게 볼륨 정보를 제공합니다. 이 경우 적절한 아카이브 전략을 적용하여, 연결되지 않은 Amazon EBS 볼륨을 삭제하면 스토리지 비용이 최소화됩니다.

⁵ <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html>

⁶ <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AmazonEBS.html>

⁷ <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html>

Amazon S3

세계 어디에서나 인터넷을 통해 액세스할 수 있고 99.99999999%의 내구성을 보장하는 Amazon S3는 사용 가능한 스토리지 양에 제한이 없으며, 쿼 액세스 스토리지에 더 이상 필요 없는 데이터를 삭제하거나 Amazon Glacier에 아카이브할 수 있는 메커니즘을 고객에게 제공합니다. Amazon S3 수명 주기 정책을 사용하면 Amazon S3에서 Amazon Glacier로 데이터를 마이그레이션하거나 완전히 삭제할 수 있습니다. Amazon S3는 다양한 수준의 내구성, 가용성, 가격대로 구성된 스토리지 클래스⁸도 제공합니다. 스토리지 메커니즘이 더 이상 필요하지 않을 때 다른 클래스가 유용할 수 있습니다.

99.99999999% 내구성.

Amazon Glacier

Amazon Glacier를 이용해 아카이브 요구 사항을 충족하고 비용 최적화를 개선할 수 있습니다. Amazon Glacier는 설계 면에서 콜드 스토리지입니다. 보관해야 하지만 자주 액세스할 필요 없는 데이터에 사용해야 합니다. S3에서 Amazon Glacier로 원활한 마이그레이션이 가능하지만, Amazon Glacier에서 사소하게 자주 검색하게 되면 비용이 많이 발생합니다. 이런 이유로, 검색 프로세스를 신중하게 고려해야 합니다.

애플리케이션 서비스

애플리케이션 서비스는 워크로드 배포와 관련된 획일적이고 과중한 관리 작업의 필요성을 없애, 고객이 중요 비즈니스에 집중할 수 있도록 도와주는 제품입니다. 예를 들어 Amazon Relational Database Service(Amazon RDS)와 Amazon DynamoDB는 관리형 데이터베이스 서비스입니다. DynamoDB 서비스는 NoSQL 솔루션을 제공하는 데 필요한 데이터베이스 인스턴스를 프로비저닝하고 유지합니다.

AWS 월간 청구서에서 절감 효과를 확인할 수는 없지만, 이러한 관리형 서비스 덕분에 DBA와 관리자는 핵심 비즈니스와 백업, 복구 등의 개발 작업에 집중할 수 있습니다.

⁸ <http://docs.aws.amazon.com/AmazonS3/latest/dev/storage-class-intro.html>

물론 EC2 인스턴스에서 바로 이러한 애플리케이션이나 데이터베이스를 실행할 수 있습니다. 하지만 EC2에서 데이터베이스나 하둡 프레임워크를 구축할 경우 팀에서 직접 배포를 구성하고 관리, 유지해야 한다는 점을 감안하면 Amazon EMR이나 DynamoDB 같은 애플리케이션 서비스가 시간과 비용 절약에 얼마나 큰 도움을 주는지 실감할 것입니다.

Amazon RDS, Amazon EMR 등의 데이터베이스 및 분석 제품, 그리고 Amazon Elastic Transcoder 및 Amazon Workspaces 등의 서비스는 데이터베이스 백업 자동화, 환경 초기화 및 지속적 관리, 자동 호스트 대체 같은 확실적인 관리 활동을 최소화하는 데 도움을 줍니다. 이러한 서비스를 이용함으로써 시간과 효율성을 얻으면 출시 기간을 단축하는 등 무형의 자산이 될 수 있습니다.

데이터 전송

데이터 송신과 수신은 AWS 리전으로 들어오고 나가는 이동을 뜻합니다. AWS는 AWS에서 인터넷으로의 데이터 전송에 대해 계층화된 요금을 적용합니다(많이 사용할수록 지불하는 단위 요금은 줄어듦). 데이터 전송 비용을 줄이기 위해, 인프라의 일부분에 AWS CloudFront⁹ 및 AWS Direct Connect¹⁰ 같은 AWS 서비스를 사용하도록 구성할 수 있습니다. 예를 들어 현재 배포가 온프레미스 데이터 센터와 AWS를 포함한 하이브리드 모델이라면 AWS Direct Connect를 활용하여 AWS 리소스와 데이터 센터 사이에 인터넷 대신 프라이빗 광섬유 링크를 구축할 수 있습니다. 온프레미스 데이터 센터로 전달되는 데이터 트래픽은 외부 데이터 전송으로 간주되며 월간 청구서에 요금이 청구됩니다. 고객은 AWS Direct Connect를 이용해 이 비용을 없앨 수 있습니다. 이 프라이빗 연결을 통한 데이터 트래픽에 발생하는 비용은 서비스에 포함되기 때문입니다. AWS Direct Connect 비용을 기준으로 외부 데이터 전송 관련 월간 요금을 비교하면 적절한 요금제를 찾는 데 도움이 됩니다.

마찬가지로, Amazon S3로 콘텐츠를 전송하는 경우 Amazon CloudFront를 사용하면 최종 사용자와 S3 버킷 간 지연 시간을 줄일 수 있습니다. 예를 들어 미국 서부(오레곤) 리전의 S3 버킷에서 인터넷으로 전송할 경우, 월간 10TB까지는 GB당 0.090 USD의 요금이 발생합니다. Amazon CloudFront 솔루션을 배포한 경우 전송 요금은 GB당 0.085 USD로 떨어지고 전체 지연 시간이 줄어듭니다.

⁹ <https://aws.amazon.com/cloudfront/details/>

¹⁰ <https://aws.amazon.com/directconnect/details/>

AWS 아키텍처를 지속적으로 평가하고 어디에서 데이터 전송 비용이 발생하는지 이해하면 전체 실행 비용을 줄이는 동시에 최종 사용자 경험을 개선하는 방향으로 배포를 수정할 수 있습니다.

운영 기술

자동화 도구는 IT 배포와 관련된 일부 관리 작업을 최소화하는 데 도움이 될 수 있습니다. 애플리케이션 서비스의 장점과 마찬가지로, 자동화된 방식 또는 DevOps 방식으로 AWS 인프라에 접근하면 수작업을 최소화하면서 확장성과 탄력성을 보장할 수 있습니다. AWS 환경 및 관련 지출에 대한 일정 수준의 통제도 가능합니다. 예를 들어 엔지니어나 개발자가 관리 및 감사가 가능한 기존의 프로세스와 도구(예: AWS Service Catalog¹¹ 같은 프로비저닝 포털)를 통해서만 AWS 리소스를 프로비저닝할 수 있다면 독립형 리소스를 단순히 켜는(그리고 주로 방치하는) 것만으로 발생하는 지출이나 낭비를 막을 수 있습니다.

AWS는 AWS 리소스 관리에 사용할 수 있는 도구를 지속적으로 개발하고 있습니다. AWS CloudFormation, AWS Elastic Beanstalk, AWS CodeCommit, AWS CodeDeploy, AWS CodePipeline 등의 서비스는 관리 오버헤드와 그에 따른 개발 시간을 줄이는 데 도움을 줄 수 있습니다. 팀에서 직접 랙 프로비저닝, 케이블 관리, 구매 주문서 작성에 시간을 할애할 필요 없이 제품 개발에만 집중할 수 있습니다. 이러한 서비스에 Auto Scaling, 스팟 인스턴스, 수명 주기 정책을 결합하면 효율성과 일관성이 개선됩니다.

태그 지정

태그는 AWS 리소스에 할당하는 레이블입니다. 각 태그는 사용자가 정의하는 키와 값으로 구성됩니다. AWS에서는 비용 할당 보고서에서 리소스 비용을 구성하기 위한 메커니즘으로 태그를 사용합니다. 예를 들어 EC2 인스턴스에 태그를 지정하면 다른 프로젝트, 부서 또는 소유자에게 속한 리소스를 식별하는 데 유용합니다. 태그에는 조정, 분석 또는 내부 차지백 모델에 도움이 될 만한 기타 유용한 정보도 포함되어 있습니다.

¹¹ <https://aws.amazon.com/servicecatalog/>

Amazon CloudWatch

Amazon CloudWatch를 이용하면 EC2 인스턴스로부터 CPU 사용률, 데이터 전송, 디스크 사용 활동에 관한 측정치를 모니터링할 수 있습니다. 뿐만 아니라 Amazon CloudWatch를 이용해 Amazon DynamoDB 테이블, Amazon EBS 볼륨, Amazon RDS 데이터베이스 인스턴스, Amazon EMR 작업 흐름, Elastic Load Balancing 로드 밸런서, Amazon SQS 대기열, Amazon Simple Notification Service(Amazon SNS) 주제에 관한 측정치도 모니터링할 수 있습니다. 이러한 서비스를 이용할 경우 Amazon CloudWatch가 포함됩니다.

한편, 단순한 API 요청을 통해 애플리케이션이 생성한 후 Amazon CloudWatch가 모니터링하는 사용자 지정 측정치를 만들 수 있습니다. 이러한 측정치는 AWS의 워크로드가 어떻게 작동하는지 파악하여 필요에 따라 리소스 할당을 조정하는 데 유용할 수 있습니다.

기존 시스템과 애플리케이션의 로그 파일을 통합할 수도 있습니다. 이 로그 파일을 Amazon CloudWatch 로그로 보내 거의 실시간으로 모니터링할 수 있습니다.

Amazon CloudWatch 경보

Amazon CloudWatch 측정치를 이용하면 저조하거나 과도한 사용률에 대한 경보와 알림을 구성할 수 있습니다. 측정치에 대한 경보를 설정하면, 서비스에서 알림을 전송하거나 기타 조치를 취하도록(Auto Scaling를 이용하여 인스턴스를 추가하거나 제거) 구성할 수 있습니다.

Amazon CloudWatch를 이용하여 예상되는 월간 AWS 요금을 바로 모니터링할 수도 있습니다. Amazon CloudWatch는 사용 중인 AWS 서비스에 대한 예상 요금 정보를 하루에도 여러 번 수신합니다. 임계값을 설정하고 이 값에 도달하면 이메일로 알림이 전송됩니다. 예를 들어 EC2 사용량 임계값을 월 1,000 USD로 설정했다면 사용량이 이 값을 초과할 경우 Amazon SNS를 통해 이메일 알림을 수신하게 됩니다.

AWS 계정 하나당 한달 간 10건의 Amazon CloudWatch 경보, 최대 1,000건의 Amazon SNS 이메일 알림을 수신하는데, 이는 AWS 사용량을 사전에 모니터링하는 경제적인 방법입니다.

비용 최적화 전략에 도움이 될 수 있는 또 한 가지 기능은 사용량 예산을 설정하여 비용(또는 “지출 데이터”)을 계획하는 기능입니다. 예산은 24시간마다 업데이트됩니다. 예산은 혼합되지 않은 비용, 구독을 추적하고 환불을 추적하지 않습니다.

다양한 유형의 비용에 대해 예산을 생성할 수 있습니다(어떤 서비스에 지출하는 금액 또는 특정 API 작업¹²을 호출하는 빈도를 파악하기 위해). 예산에 Amazon CloudWatch 경보를 적용하여 Amazon SNS에서 알림을 트리거할 수도 있습니다.

AWS Service Catalog

AWS 배포가 성숙함에 따라, 귀사의 최종 사용자가 서비스를 탐색하고 시작하는데 사용할 수 있는 맞춤형 셀프 서비스 포털인 AWS Service Catalog의 장점을 누릴 수 있습니다. IT 관리자는 개발 팀과 협력해 각 서비스에 대해 승인된 AWS 구성을 시작하는데 사용할 수 있는 Amazon CloudFormation 템플릿을 제공할 수 있습니다.

비용 최적화의 장점은 정책에 지정된 파라미터(인스턴스 유형, 스토리지 유형 등)를 환경에서 사용할 수 있다는 점입니다. 이처럼 통제된 접근 방식을 활용하면 설계 결정 및 최종 사용자에게 제공된 옵션에 따라 예산을 정의하고 비용을 예측할 수 있습니다. 기술 사용자와 협력하여 AWS Service Catalog에 포함시킬 적절한 템플릿을 정의하면 불필요한 리소스가 시작되고 켜져 있는 현상을 방지할 수 있습니다.

AWS Config

AWS 배포와 인프라를 추적하는 것은 비용을 통제하고 환경의 구조와 비용을 최적화하는 데 필수적입니다. AWS 서비스를 다양한 타사 구성 관리 도구와 통합할 수 있지만, AWS 리소스 인벤토리, 구성 내역, 구성 변경 알림을 제공하는 완전 관리형 서비스인 AWS Config도 있습니다.

¹² <http://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/budgetsmanaging-costs.html>

AWS Config 규칙은 특정 AWS 리소스나 전체 AWS 계정에 대해 바람직하다고 판단한 구성 설정을 나타냅니다. 내부 규정 준수 정책과 규제 표준을 충족하도록 AWS Config 규칙을 이행할 수 있습니다. 인프라에서 다른 리소스에 어떤 영향을 미치는지, 구성 변경이 규정된 규칙을 준수하는지 평가할 수 있습니다.

비용 효율적인 리소스

워크로드에 적합한 솔루션을 선택하는 것은 배포 비용을 절감하는 좋은 방법입니다. 예를 들어 최초 벤치마킹을 완료한 후 확실한 성능과 가격을 제공하는 인스턴스를 결정합니다. 이때 예약 인스턴스 구입을 고려할 수 있습니다. 1년 또는 3년 약정을 체결하면 이러한 인스턴스의 전체 실행 비용이 크게 절감됩니다. **Trusted Advisor** 같은 도구는 예약 인스턴스를 구입할 경우 실현 가능한 월간 절감액을 빠르게 파악하는 데 도움을 줍니다. **Trusted Advisor**는 EC2 인스턴스 사용률, 유휴 로드 밸런서, **Amazon RDS** 데이터베이스 인스턴스, 잘 사용하지 않는 **EBS** 볼륨, 무관한 탄력적 IP 주소 등 일반적인 측정치를 평가합니다. **Trusted Advisor**는 260만 건 이상의 사용량 알림을 고객에게 전송했습니다. 현재까지 **Trusted Advisor**를 통해 고객이 절감한 액수는 3억 5천만 USD 이상으로 추정됩니다.

세분화된 접근 방식

AWS 요금을 이용하면 워크로드별 리소스를 세분화할 수 있습니다. 더 이상 물리적 서버에서 호스팅할 수 있는 워크로드 구성 요소의 개수에 따라 프로젝트 시작 시점에 한 번만 구입하는 방식으로 제한할 필요가 없습니다. 이제 각 애플리케이션에 속한 다양한 요소를 분리하고, 워크로드에 프로비저닝된 리소스 중 구성 요소가 사용 중인 리소스 개수를 파악할 수 있습니다. 가장 경제적인 배포 방식은 설계 결정을 계속 검토하고 리소스 사용률을 평가하는 것입니다.

Amazon EMR 작업을 완료해야 하는 애플리케이션을 떠올려보십시오. 벤치마크 테스트 결과, 이 작업은 다섯 시간 내에 작은 크기의 인스턴스 유형에서 완료할 수 있습니다. 큰 인스턴스 유형에서는 이 작업을 한 시간 만에 완료할 수 있습니다.

큰 인스턴스의 시간당 요금은 두 배이지만 더 짧은 시간 안에 작업을 완료할 수 있기 때문에 전체 실행 비용이 더 저렴합니다.

이 예에서는 실행 비용을 더 줄이기 위해 스팟 인스턴스 사용도 고려해볼 만합니다. 현재 시간당 요금보다 낮은 요금에 입찰할 경우 더 큰 절감 효과를 누릴 수 있습니다.

비용 인지

AWS Config, Amazon CloudWatch, AWS OpsWorks, Auto Scaling, AWS CloudFormation 등의 도구를 사용하면 리소스의 효율적인 사용을 보장할 수 있지만 지속적인 평가와 벤치마킹, 운영과의 통합 주기를 반복해야만 비용 최적화를 달성할 수 있습니다.

1. 최초 평가

어떤 프로젝트를 AWS로 마이그레이션할 수 있는가? 어떤 인스턴스 유형과 스토리지 옵션이 프로젝트 요구 사항과 예산에 부합할 것인가?

2. 벤치마킹

시스템이 제대로 작동하고 있습니까? 그렇지 않다면 아키텍처를 변경하십시오(예: 인스턴스 크기 조정, SSD 드라이브 선택 등).

3. 운영 통합

AWS 솔루션을 이용한 기존 운영 프로세스 보강 또는 교체:

- a. AWS OpsWorks, AWS CloudFormation, Auto Scaling을 이용하여 리소스 탄력성을 자동화하고 지원합니다.
- b. 이 단계에 스팟 인스턴스와 예약 인스턴스 중 어떤 것이 적절한지 판단합니다.
- c. 지출 내역을 즉시 파악할 수 있도록 Amazon CloudWatch 경보와 알림을 설정하고, 필요에 따라 자동으로 취해야 할 조치를 정의합니다.

배포가 확장됨에 따라 프로젝트와 서비스를 AWS 기반으로 마이그레이션하거나 생성하면 AWS Config 같은 도구 사용을 통해 인식 수준을 높일 수 있습니다. [AWS 월 사용량 계산기](#) 를 사용하여 AWS 지출을 추정할 수도 있습니다.

AWS 가격표 API

AWS 가격표 API에서 AWS 서비스 가격을 쿼리로 보낼 수 있습니다. 서비스 변경 시 가격을 알려주는 Amazon SNS 알림을 신청할 수도 있습니다. 새로운 인스턴스 유형이 시작될 때 또는 새로운 서비스가 도입될 때처럼 AWS 가격은 때에 따라 변경됩니다.

세부 결제 보고서

세부 결제 보고서에는 요구 사항을 충족하기에 충분한 예약 인스턴스가 있는지 등을 예측할 수 있도록 해주는 시간별 항목이 포함됩니다. 이러한 보고서를 통합 결제 기능과 함께 사용하면 예약 인스턴스와 관련된 비용을 간편하게 추적하고 관리할 수 있습니다.

Jeff Barr가 이 블로그 게시물에서 지적했듯이 세부 보고서는 “연결 계정의 추가 할당 모델에 RI 선호도, 일반 요금 적용 비율 등 두 가지 핵심 기능을 제공합니다. 자세한 내용은 [블로그 게시물](#)을 참조하십시오.

E	I	J
UsageType	ItemDescription	UsageStartDate
BoxUsage	\$0.080 per Small Instance (m1.small)	10/1/2012 0:00
BoxUsage:t1.micro	\$0.012 per Linux/UNIX Micro Instance	10/1/2012 0:00
BoxUsage:t1.micro	\$0.020 per Micro Instance (t1.micro)	10/1/2012 0:00
BoxUsage:t1.micro	\$0.020 per Micro Instance (t1.micro)	10/1/2012 0:00
BoxUsage:t1.micro	\$0.012 per Linux/UNIX Micro Instance	10/1/2012 0:00
DataTransfer-In-Bytes	\$0.000 per GB - data transfer in per m	10/1/2012 0:00
DataTransfer-In-Bytes	\$0.000 per GB - data transfer in per m	10/1/2012 0:00

다음과 같은 기준에 따라 예상 비용을 보여주도록 세부 결제 보고서를 사용자 지정할 수도 있습니다.

- 시간, 일 또는 월별
- 조직의 계정별
- 제품이나 제품 리소스별
- 정의한 태그별

AWS 리소스 태그를 생성해 리포트에 있는 거의 모든 항목에 본인의 라벨을 추가할 수도 있습니다. 결제 보고서를 사용해 다음의 작업을 수행할 수 있습니다.

- 데이터를 읽을 수 있는 애플리케이션으로 결제 데이터를 가져오려는 경우.
- 결제 데이터를 사용하는 애플리케이션을 작성하려는 경우

- 당월 누계 요금을 모니터링하려는 경우
- 월별 요금을 예측하려는 경우
- 데이터를 파트너와 공유합니다.
- 결제 데이터를 회계 시스템으로 가져오려는 경우
- 여러 계정에 대한 청구서를 가져오려는 경우

이와 같은 사용자 지정을 통해 조직에 의미 있는 보고서를 생성할 수 있습니다. 추가 분석, 보고, 예측을 위해 **Amazon Redshift** 같은 데이터 웨어하우징 솔루션에서 이 데이터를 수집할 수 있습니다.

시간에 따른 최적화

최초 평가가 100% 정확할 수 없지만 AWS 서비스는 유연한 특성을 내재하고 있기 때문에 최초 설계 결정을 변경할 수 있습니다. 시간이 지나면서 요구 사항이 변하기 때문에 AWS 리소스에 대한 지속적인 재평가와 모니터링이 필수적입니다.

AWS는 비용을 절약하고 전체 효율성을 높일 수 있는 새로운 기능과 서비스도 출시할 예정입니다. 예를 들어 저렴하면서도 확장 가능한 데이터 웨어하우징 솔루션인 **Amazon Redshift**는 데이터 웨어하우징의 초기 비용을 줄였습니다. **Amazon Aurora**는 완전 관리 형태로 **MySQL**과 호환되는 관계형 데이터베이스 엔진으로서 고사양 상업용 데이터베이스의 속도 및 안정성이 오픈 소스 데이터베이스의 간편성 및 비용 효율성과 결합되었습니다.

결론

비용 최적화는 지속적인 노력입니다. 파일럿 또는 테스트 워크로드의 최초 평가에서 성숙한 AWS 인프라까지, 관리자는 아키텍처 접근법을 계속해서 다시 평가하고 테스트해야 합니다. 이 문서에서 언급한 프로그래밍 방식 기능과 액세스, AWS 기능과 서비스 덕분에 이러한 노력이 더 쉬워졌습니다.

AWS는 뛰어난 복원성과 응답성, 적응성이 보장되는 배포를 구축하는 동안 효율성을 극대화하도록 돕습니다. 진정으로 최적화된 비용의 배포를 원한다면 이 문서에 언급된 도구와 기술을 가능한 많이 사용해야 합니다.

기고자

Bryan Tang, Callum Hughes

부록: 비용 효율적인 스토리지의 예

컴퓨팅 관련 스토리지와 관련하여, 인스턴스 스토리지가 워크로드 요구 사항을 만족시킬 수 있는가 아니면 그 대신 Amazon EBS가 필요할 것인가라는 중요한 질문이 제기됩니다.

다음 예는 미국 동부(버지니아 북부) 리전에서 한달 동안 연중무휴 24시간 m3.xlarge EC2 인스턴스를 실행한다고 가정했을 때의 결과입니다.

1. EC2 비용: 194.72 USD/월
2. EC2 + 500GB EBS 범용(SSD) 볼륨: 194.72 USD + 50 USD = 244.72 USD/월
3. EC2 + 500GB EBS 프로비저닝된 IOPS(SSD) 볼륨(3000 IOPS): 194.72 USD + 257.50 USD = 452.22 USD/월

첫 번째 시나리오에서는 인스턴스 스토리지 활용만으로 스토리지 요구 사항이 충족됩니다. 워크로드에 영구 데이터 스토리지가 필요한 나머지 두 시나리오에는 Amazon EBS가 추가되었습니다.

AWS는 고객들에게 지속적으로 높은 IOPS가 필요한지 판단해야 한다고 조언합니다. 세 번째 시나리오에서 보듯이 PIOPS의 성능 장점은 월간 비용에서 실감할 수 있습니다. 설정한 PIOPS가 충분히 사용되고 있는지 비용 최적화의 관점에서 조사해야 합니다. 그렇지 않다면 PIOPS를 낮추거나 EBS 범용(SSD)의 버스트 가능한 모델로 이전하는 방안을 고려하십시오.

AWS는 비용과 성능 사이에서 최적의 균형을 달성할 수 있는 옵션을 제공하지만, 모니터링을 통해 각자의 결정을 확인하는 것이 중요합니다.

Amazon S3의 기술적 장점과 저렴한 월간 요금 때문에 일부 AWS 고객은 모든 데이터를 S3에 보관합니다. 수천만 개 파일을 S3에 보관하는 고객에게는 아카이브 전략이 스토리지 비용에 긍정적인 영향을 미칠 것입니다. 현재 사용량과 상관없이 아카이브 전략을 구현하여, 지속적인 비즈니스 성장에 따라 아카이빙 프로세스를 통합하는 방법은 언제나 좋은 아이디어라고 생각합니다.

다음 예는 미국 동부(버지니아 북부) 리전에서 스토리지와 아카이브가 필요한 10TB의 데이터가 있다고 가정했을 때의 결과입니다.

1. S3: 302.45 USD/월
2. S3 Standard -Infrequent Access(Standard IA): 128.00 USD/월
3. S3 Reduced Redundancy Storage(RRS): 242.08 USD/월
4. Amazon Glacier: 71.68 USD/월

Amazon S3는 내구성, 가용성이 서로 다른 3개 스토리지 클래스를 제공합니다. 그러나 거의 사용되지 않지만 규제나 기타 요구 사항으로 인해 필요한 데이터의 경우, Amazon Glacier가 매력적인 옵션입니다. 현재 데이터 보존 요구 사항을 평가한 후, 원하는 일정에 맞춰 저렴한 스토리지 클래스로 데이터가 자동으로 이전되는 Amazon S3 수명 주기 관리의 장점을 누리시기 바랍니다. 이러한 월간 예상 비용을 살펴보면 Amazon S3에 7년 간 객체를 보관하는 경우와 Amazon S3에 1년 간 보관한 후 Amazon Glacier에 6년 간 아카이빙하는 방법 사이의 차이를 알 수 있습니다.

참고 문헌

자세한 내용은 다음을 참조하십시오.

- [AWS Well-Architected 프레임워크](#)¹