

자동화를 사용하여 신속하게 복구합니다. 

 AWS
CloudFormation

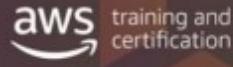
템플릿을 사용하여 필요에 따라 리소스 모음을 신속하게 배포합니다.

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

zlagu COPY.com

AWS CloudFormation을 사용하면 텍스트 파일로 전체 인프라를 모델링할 수 있습니다. 이 템플릿은 인프라의 단일 정보 소스가 됩니다. 그러면 조직 전체에서 사용되는 인프라 구성 요소를 표준화하여 구성 규정을 준수하고 문제를 더 빠르게 해결할 수 있습니다.

AWS CloudFormation은 리소스를 안전하고 반복 가능한 방식으로 프로비저닝하므로 수작업을 수행하거나 사용자 지정 스크립트를 작성할 필요 없이 인프라와 애플리케이션을 구축 및 재구축 할 수 있습니다. AWS CloudFormation에서는 스택을 관리할 때 수행해야 할 적절한 작업을 결정하고, 오류가 탐지되면 변경 사항을 자동으로 롤백합니다.

자동화를 사용하여 신속하게 복구합니다. 

AWS CloudFormation
템플릿을 사용하여 필요에 따라 리소스 모음을 신속하게 배포합니다.

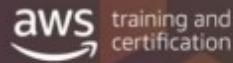
AWS Elastic Beanstalk
단 몇 번의 클릭으로 전체 스택을 빠르게 재배포합니다.

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

AWS Elastic Beanstalk을 사용하여 업데이트된 원본 번들을 업로드한 후 AWS Elastic Beanstalk 환경에 배포하거나 이전에 업로드한 버전을 다시 배포할 수 있습니다.

이전에 업로드한 애플리케이션 버전을 환경에 배포할 수 있습니다.

자동화를 사용하여 신속하게 복구합니다.



AWS CloudFormation
템플릿을 사용하여 필요에 따라 리소스 모음을 신속하게 배포합니다.

AWS Elastic Beanstalk
단 몇 번의 클릭으로 전체 스택을 빠르게 재배포합니다.

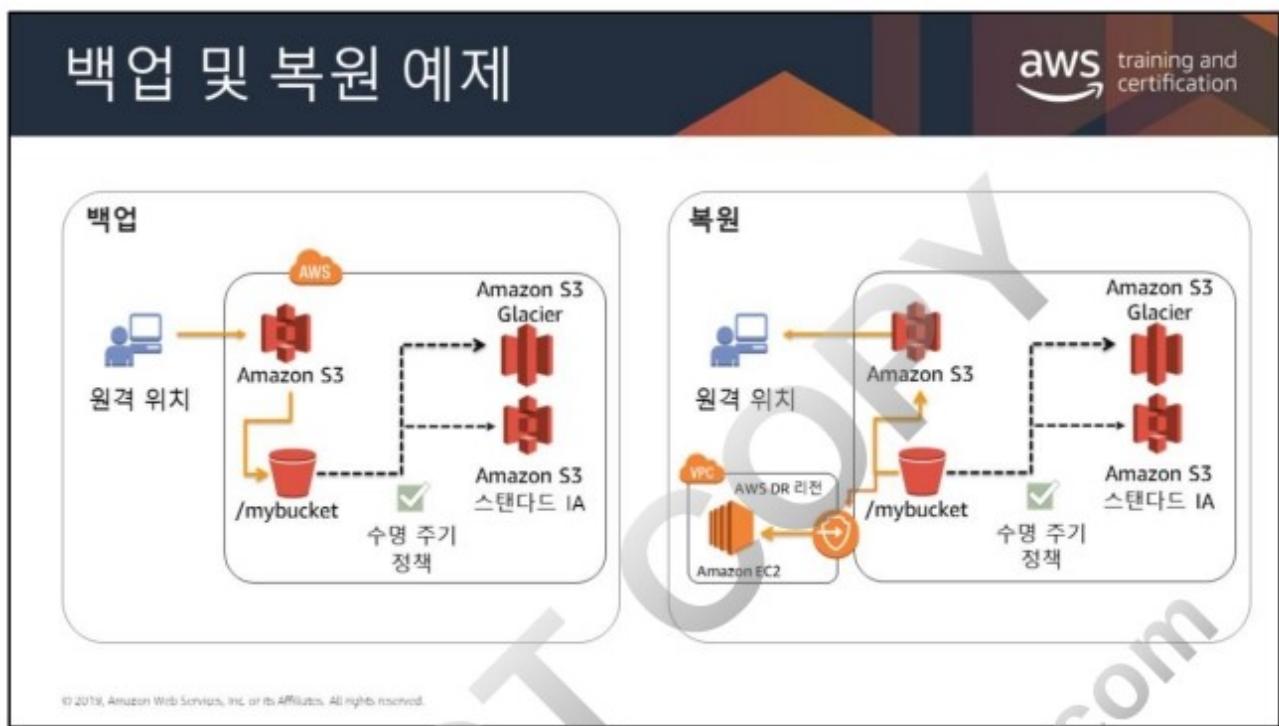
AWS OpsWorks

- 자동 호스트 교체
- 복구 단계에서 AWS CloudFormation과 결합합니다.
- 정의된 RTO를 지원하는 새로운 스택을 제공합니다.

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

AWS OpsWorks는 모든 유형과 규모의 애플리케이션을 간편하게 배포하고 운영할 수 있는 애플리케이션 관리 서비스입니다. 사용자의 환경은 일련의 계층으로서 정의할 수 있으며, 각 계층은 애플리케이션 티어로 구성할 수 있습니다. AWS OpsWorks는 자동 호스트 교체를 포함하고 있기 때문에 인스턴스 장애가 발생하면 자동으로 교체됩니다. 준비 단계에서 AWS OpsWorks를 사용하면 환경을 템플릿으로 만들 수 있으며, 복구 단계에서 AWS CloudFormation과 결합할 수 있습니다. 정의된 RTO를 지원하는 저장된 구성에서 새 스택을 신속하게 프로비저닝할 수 있습니다.

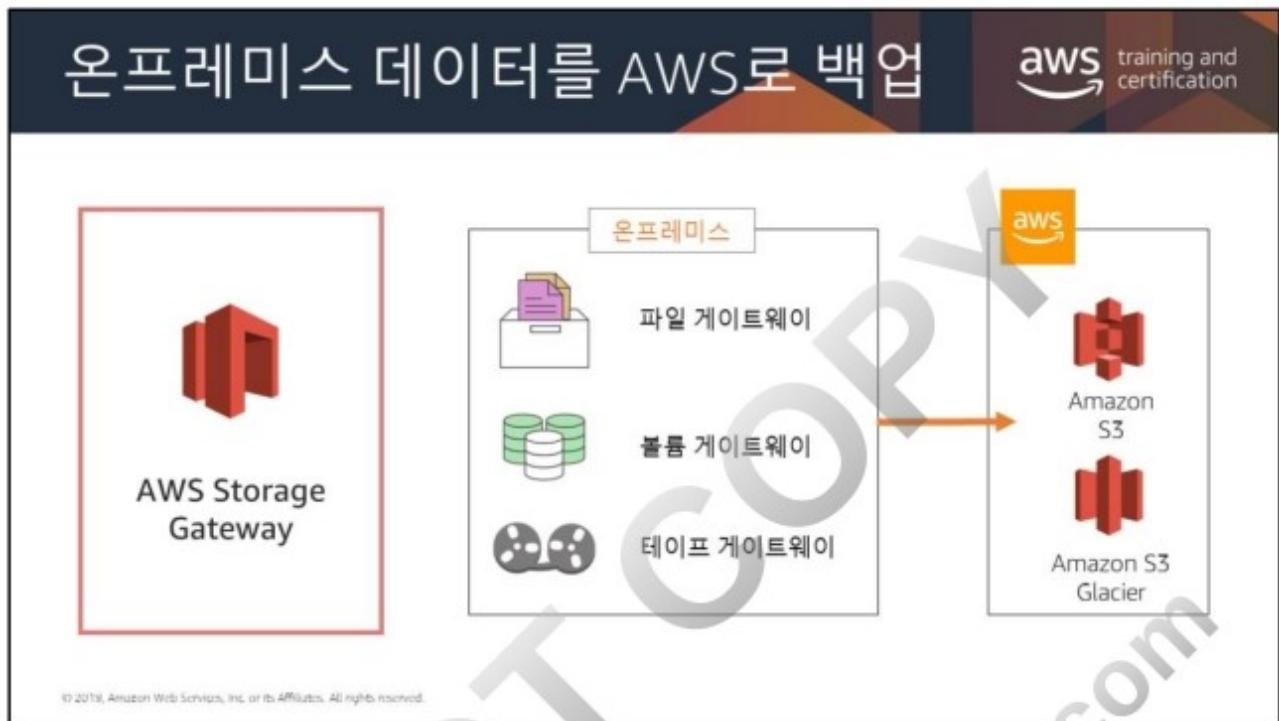




대부분의 기존 환경에서는 데이터를 테이프에 백업하여 정기적으로 오프사이트로 전송합니다. 이 방법을 사용할 경우, 가동 중단 또는 재해 발생 시 시스템을 복원하는 데 많은 시간이 걸릴 수 있습니다.

Amazon S3는 복원을 수행하기 위해 급하게 필요할 수 있는 백업 데이터를 위한 이상적인 대상입니다. Amazon S3와 데이터를 주고 받는 과정은 주로 네트워크를 통해 이루어지므로 어떤 위치에서든 액세스가 가능합니다. Amazon S3와 통합되는 다양한 상용 및 오픈 소스 백업 솔루션이 있습니다. 예:

- AWS Snowball을 사용하면 스토리지 디바이스를 AWS에 직접 연결하여 대규모의 데이터 세트를 전송할 수 있습니다.
- 몇 시간의 검색 시간으로 충분한 장기간 데이터 저장의 경우, Amazon S3와 동일한 내구성 모델을 가진 Amazon S3 Glacier가 있습니다. Amazon S3 Glacier 및 Amazon S3를 함께 사용하면 계층화된 백업 솔루션을 생성할 수 있습니다.



AWS Storage Gateway는 온프레미스 소프트웨어 어플라이언스를 클라우드 기반 스토리지와 연결하여 온프레미스 IT 환경과 AWS 스토리지 인프라 간에 원활하면서도 매우 안전한 통합을 제공합니다. 이 서비스를 사용하면 확장 가능하고 비용 효율적인 스토리지인 AWS 클라우드에 데이터를 안전하게 저장할 수 있습니다. Storage Gateway는 기존 애플리케이션과 연동하는 업계 표준 스토리지 프로토콜을 지원하는 동시에 Amazon S3 또는 Amazon S3 Glacier에서 암호화된 모든 데이터를 안전하게 저장합니다.

AWS Storage Gateway를 사용하면 AWS 관리 서비스를 로컬로 확장할 수 있습니다. 이 서비스는 Amazon CloudWatch, AWS CloudTrail, AWS KMS, AWS IAM 등과도 통합됩니다.

AWS Storage Gateway는 파일, 볼륨 및 테이프라는 3가지 스토리지 인터페이스를 지원합니다. 각 게이트웨이에서는 1가지 유형의 인터페이스를 제공할 수 있습니다.

파일 게이트웨이를 사용하면 NFS 및 SMB 파일 프로토콜을 사용하여 Amazon S3에서 객체를 저장 및 검색할 수 있습니다. 파일 게이트웨이를 통해 작성된 객체는 S3에서 직접 액세스할 수 있습니다.

볼륨 게이트웨이는 iSCSI 프로토콜을 사용하여 애플리케이션에 블록 스토리지를 제공합니다. 볼륨의 데이터는 Amazon S3에 저장됩니다. AWS에서 iSCSI 볼륨에 액세스하려면 EBS 볼륨을 생성하는 데 사용될 수 있는 EBS 스냅샷을 작성할 수 있습니다.

테이프 게이트웨이는 백업 애플리케이션에 가상 미디어 체인저 및 가상 테이프 드라이브로 구성된 iSCSI 가상 테이프 라이브러리(VTL) 인터페이스를 제공합니다. 가상 테이프 데이터는 Amazon S3에 저장되거나 Amazon S3 Glacier에 보관할 수 있습니다.

AWS 클라우드에 온프레미스 데이터를 백업하려면 다음의 두 가지 일반적인 방식 중 한 가지를 선택하면 됩니다.

- AWS 서비스에 API 호출을 실행하여 Amazon S3에 백업 데이터를 직접 작성합니다.
- 인터넷에서 안전한 HTTP PUT 및 GET 요청을 통해 백업 데이터를 직접 작성하거나 검색합니다. 여기서 엔드포인트가 Amazon S3에 직접 연결되어 데이터를 작성 및 검색합니다.

게이트웨이 가상 테이프 라이브러리(VTL)

가상 테이프 컬렉션을 무제한으로 보유할 수 있습니다.

각 가상 테이프는 Amazon S3에서 지원하는 가상 테이프 라이브러리에 저장하거나 혹은 Amazon S3 Glacier에서 지원하는 가상 테이프 선반에 저장할 수 있습니다.

게이트웨이 캐싱 볼륨

기본 데이터는 Amazon S3에 저장하고 자주 액세스하는 데이터는 로컬에 보관할 수 있습니다. 게이트웨이 캐싱 볼륨은 기본 스토리지에서 상당한 비용 절감을 제공하며 온프레미스로 스토리지를 확장할 필요성을 최소화하고 자주 액세스하는 데이터에 대해 액세스 지연 시간을 짧게 유지합니다.

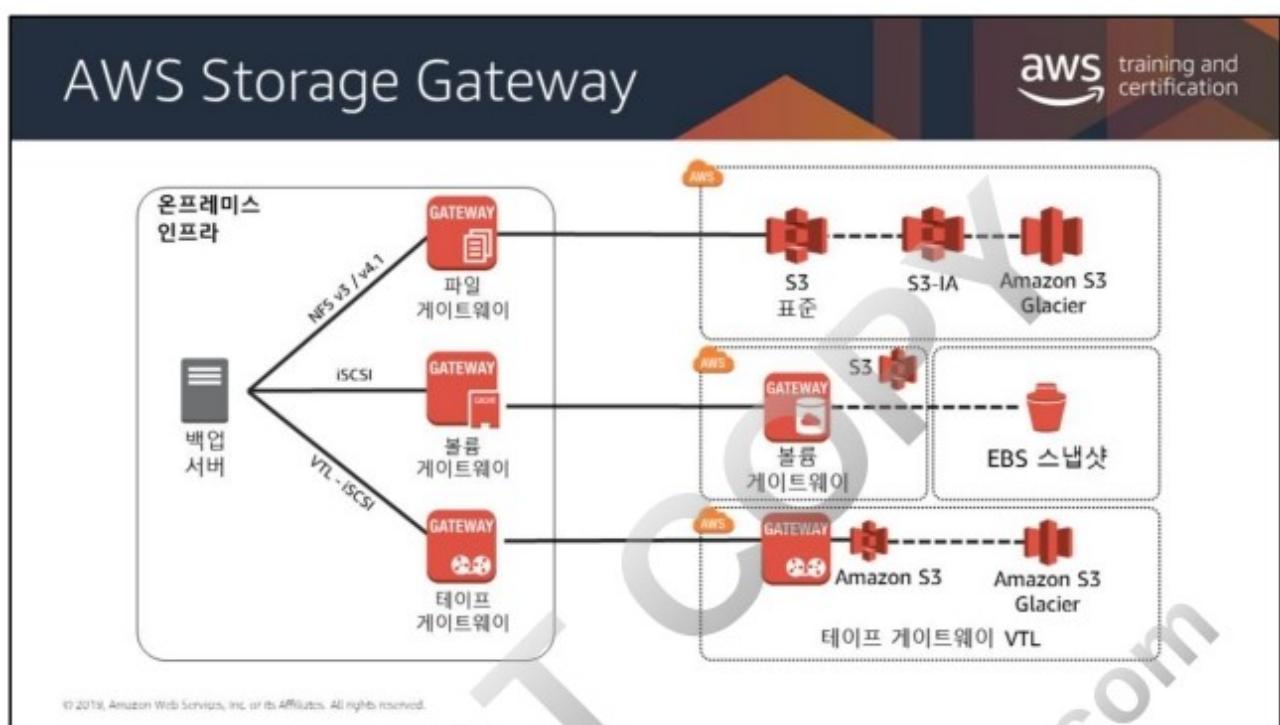
게이트웨이 저장 볼륨

전체 데이터 세트에 액세스 시 지연 시간이 짧아야 하는 경우 기본 데이터를 로컬에 저장하도록 온프레미스 데이터 게이트웨이를 구성하고 이 데이터의 특정 시점 스냅샷을 비동기적으로 Amazon S3에 백업할 수 있습니다.

AWS Storage Gateway 하드웨어 어플라이언스

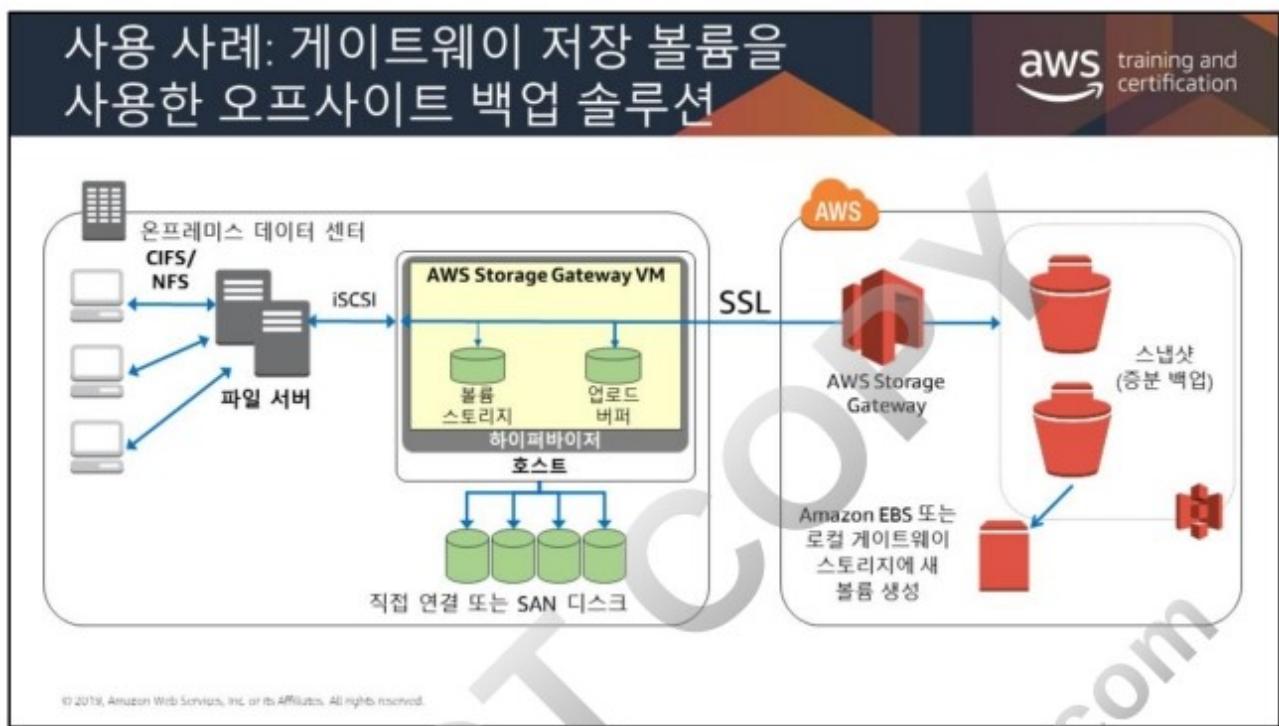
AWS Storage Gateway 하드웨어 어플라이언스는 온프레미스에 설치할 수 있는 타사 서버에 미리 설치된 AWS Storage Gateway 소프트웨어를 제공하는 하드웨어 어플라이언스입니다. AWS Storage Gateway 하드웨어 어플라이언스는 AWS Management Console의 하드웨어 페이지에서 관리할 수 있습니다.

<https://docs.aws.amazon.com/storagegateway/latest/userguide/HardwareAppliance.html>



NFS v3 및 v4.1 프로토콜 외에 AWS Storage Gateway 서비스는 파일 게이트웨이에 SMB(서버 메시지 블록) 프로토콜을 추가했습니다. 이 프로토콜은 Microsoft Windows용으로 개발된 파일 기반 애플리케이션에서 손쉽게 Amazon Simple Storage Service (S3)에 객체를 저장하고 해당 객체에 액세스할 수 있도록 지원합니다. 자세한 내용은 다음을 참조하십시오.

<https://aws.amazon.com/about-aws/whats-new/2018/06/aws-storage-gateway-adds-smb-support-to-store-objects-in-amazon-s3/>



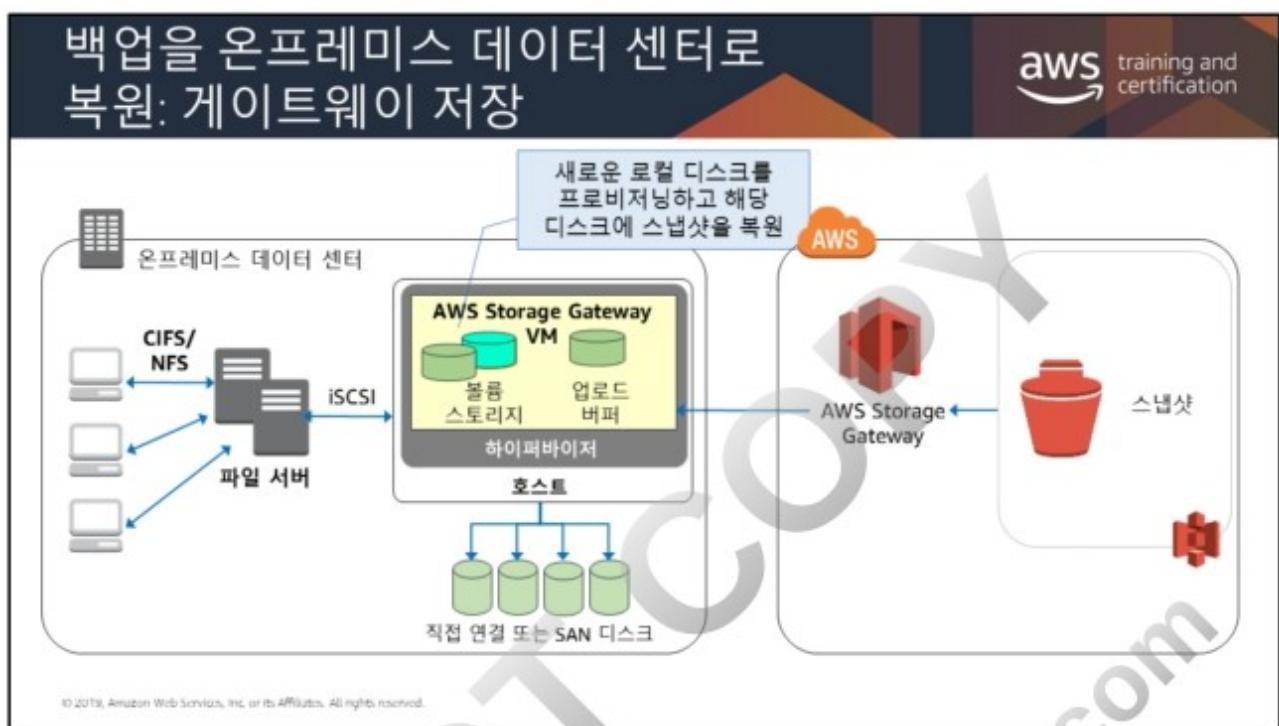
AWS Storage Gateway 소프트웨어 어플라이언스(가상 머신)를 데이터 센터의 호스트에 설치하고 활성화하면 게이트웨이 스토리지 볼륨을 생성하여 온프레미스 Direct Attached Storage (DAS) 또는 Storage Area Network (SAN) 디스크에 매핑할 수 있습니다. 새 디스크로 진행해도 되고, 데이터를 이미 저장하고 있는 디스크로 진행해도 됩니다. 그리고 나서 이러한 스토리지 볼륨을 온프레미스 애플리케이션 서버에 iSCSI 디바이스로서 마운트할 수 있습니다. 온프레미스 애플리케이션이 데이터를 게이트웨이 저장 볼륨에서 쓰고 읽으면 이 데이터는 볼륨에 할당된 디스크에 저장되고 검색됩니다.

Amazon S3로의 업로드를 준비하기 위해 게이트웨이는 업로드 버퍼라고 불리는 스테이징 영역에도 수신 데이터를 저장합니다. 온프레미스 DAS 또는 SAN 디스크를 작업 스토리지로 사용할 수 있습니다. 게이트웨이는 업로드 버퍼의 데이터를 암호화된 Secure Sockets Layer (SSL) 연결을 통해 AWS 클라우드에서 실행되는 AWS Storage Gateway 서비스로 업로드합니다. 이때 서비스는 Amazon S3에 데이터를 암호화하여 저장합니다.

스냅샷이라고 불리는 저장 볼륨에 대한 증분 백업을 실행할 수 있습니다. 게이트웨이는 Amazon S3의 이러한 스냅샷을 Amazon EBS로 저장합니다. 새 스냅샷을 만들 때 마지막 스냅샷 이후에 변경된 데이터만 저장됩니다. 스냅샷은 예정된 시간에 또는 일회성으로 시작할 수 있습니다. 스냅샷을 삭제할 때는 다른 스냅샷에 필요하지 않은 데이터만 제거됩니다.

데이터의 백업을 복원해야 할 경우, Amazon EBS 스냅샷을 온프레미스 게이트웨이 스토리지 볼륨에 복원할 수 있습니다. 또한 스냅샷을 새 Amazon EBS 볼륨의 기반으로 사용한 다음, Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스에 연결할 수 있습니다.

DO NOT COPY
zlagusdbs@gmail.com

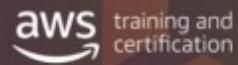


게이트웨이 저장 볼륨의 경우, 볼륨 데이터가 온프레미스에 저장됩니다. 이 경우, 스냅샷이 Amazon S3에 지속적인 오프사이트 백업을 제공합니다. 예를 들어, 스토리지 볼륨으로 할당된 로컬 디스크가 손상된 경우 새로운 로컬 디스크를 프로비저닝하고 볼륨 생성 과정 중에 스냅샷으로 복원할 수 있습니다. (이 방법에 대한 자세한 내용은

<http://docs.aws.amazon.com/storagegateway/latest/userguide/ApplicationStorageVolumesStored-Adding.html>에서 Adding a Storage Volume을 참조하십시오).

게이트웨이 저장 볼륨으로의 스냅샷 복원을 시작하고 나면 백그라운드에 스냅샷 데이터가 다운로드됩니다. 이 기능으로 스냅샷에서 볼륨을 생성한 후 Amazon S3에서 볼륨으로 모든 데이터가 전송될 때까지 기다리지 않아도 애플리케이션에서 볼륨과 모든 데이터에 액세스할 수 있습니다. 애플리케이션에서 아직 로드되지 않은 데이터에 액세스하는 경우, 게이트웨이는 요청된 데이터를 즉시 Amazon S3에서 다운로드하며, 백그라운드에서 볼륨의 나머지 데이터 로드를 진행합니다.

백업 및 복원

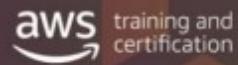


준비 단계

- 현재 시스템을 백업합니다.
- Amazon S3에 백업을 저장합니다.
- AWS의 백업으로부터 복원하는 절차를 기술합니다.
 - 사용할 AMI(필요하면 자체 AMI를 구성)
 - 백업으로부터 시스템을 복원하는 방법
 - 새로운 시스템으로 전환하는 방법
 - 배포를 구성하는 방법

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

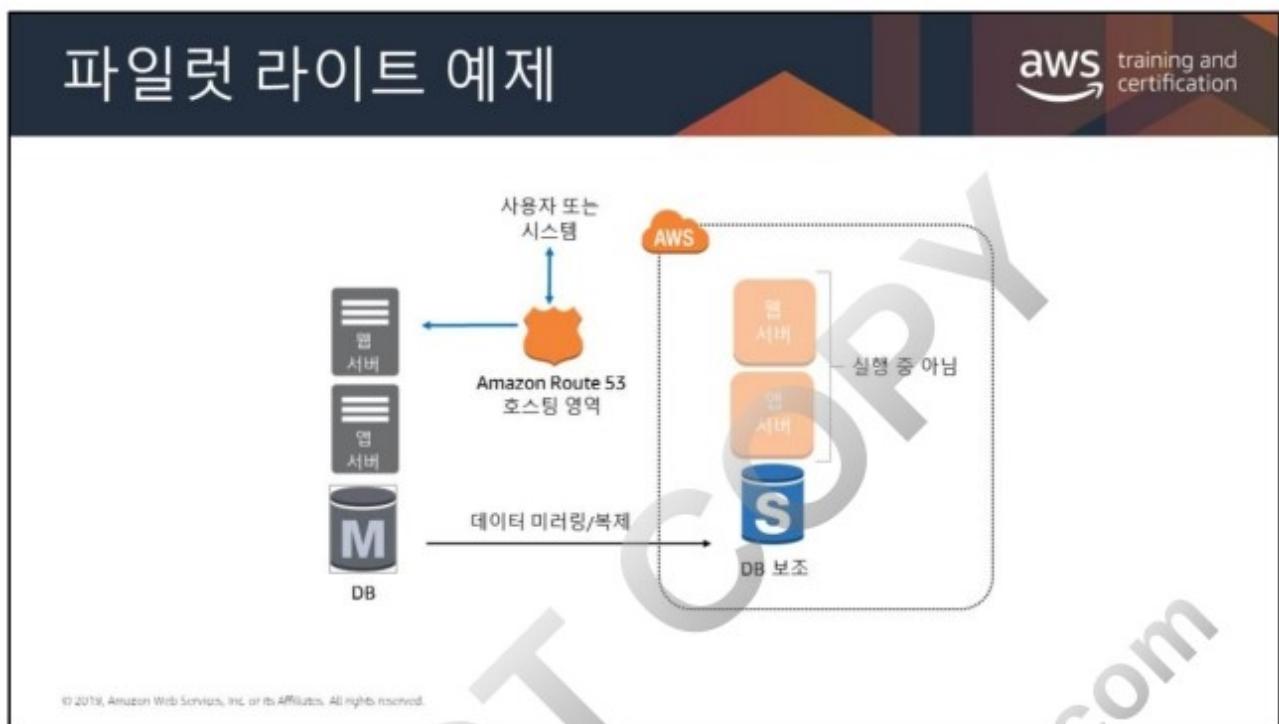
백업 및 복원



재해 발생 시:

- Amazon S3로부터 백업을 검색합니다.
- 필요한 인프라를 준비합니다.
 - 준비된 AMI, ELB 등이 있는 Amazon EC2 인스턴스.
 - AWS CloudFormation을 사용해 주요 네트워킹 배포를 자동화합니다.
- 백업으로부터 시스템을 복원합니다.
- 새로운 시스템으로 전환합니다.
 - AWS를 가리키도록 DNS 레코드를 조정합니다.

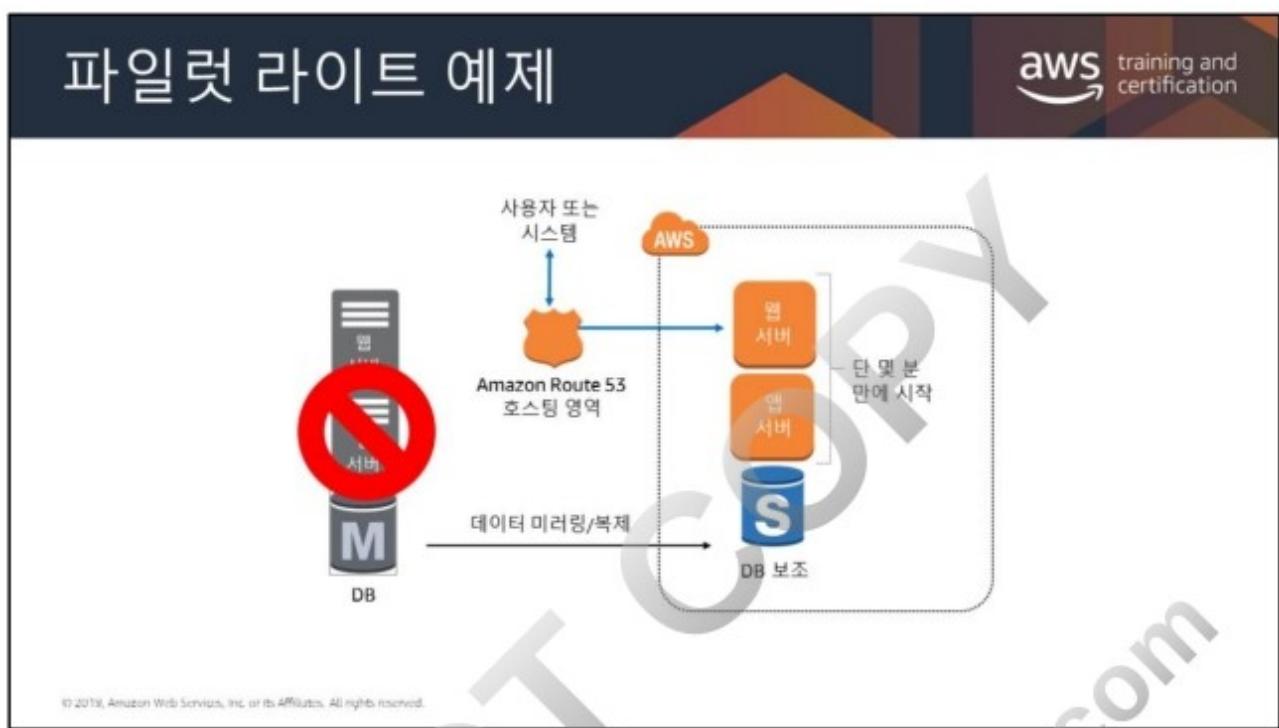
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



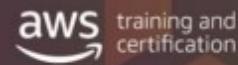
이 패턴은 비교적 저렴하게 구현할 수 있습니다. 재해 복구 준비 단계에서는 데이터 마이그레이션과 영구 스토리지를 지원하는 서비스 및 기능의 사용을 고려하는 것이 중요한데, 이로써 재해가 닥쳤을 때 AWS에 백업한 데이터를 복구할 수 있기 때문입니다. AWS에서 시스템의 규모 축소 또는 최대 확장이 수반되는 일부 시나리오의 경우, 컴퓨팅 리소스가 필요한 경우도 있습니다.

재해에 대처할 때, AWS에서 시스템을 실행할 수 있도록 신속하게 컴퓨팅 리소스를 공급하거나 AWS에서 이미 실행 중인 리소스에 장애 조치를 취하는 것이 관건입니다. 필수적인 인프라에는 DNS, 네트워킹 기능 및 다양한 Amazon EC2 기능들이 포함됩니다.

준비 단계에서는 복구 단계 시, 전체 환경이 시작되는 작은 코어인 파일럿 라이트에 정기적으로 변경되는 데이터를 복제해야 합니다. 운영 체제 및 애플리케이션과 같이 업데이트 빈도가 비교적 적은 데이터는 정기적으로 업데이트하고 AMI로 저장할 수 있습니다.



파일럿 라이트(Pilot Light)



이점

- 매우 비용 효율적(더 적은 24/7 리소스 사용)

준비 단계

- 데이터를 복제 또는 미러링하도록 Amazon EC2 인스턴스를 설정합니다.
- AWS에 모든 사용자 정의 지원 소프트웨어 패키지가 있는지 확인합니다.
- 빠른 복구가 필요한 핵심 서버의 Amazon 머신 이미지(AMI)를 생성 및 관리합니다.
- 이 서버를 정기적으로 실행하고, 테스트하고, 소프트웨어 업데이트 및 구성 변경 사항을 적용합니다.
- AWS 리소스의 프로비저닝을 자동화할 것인지 고려합니다.

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

파일럿 라이트(Pilot Light)



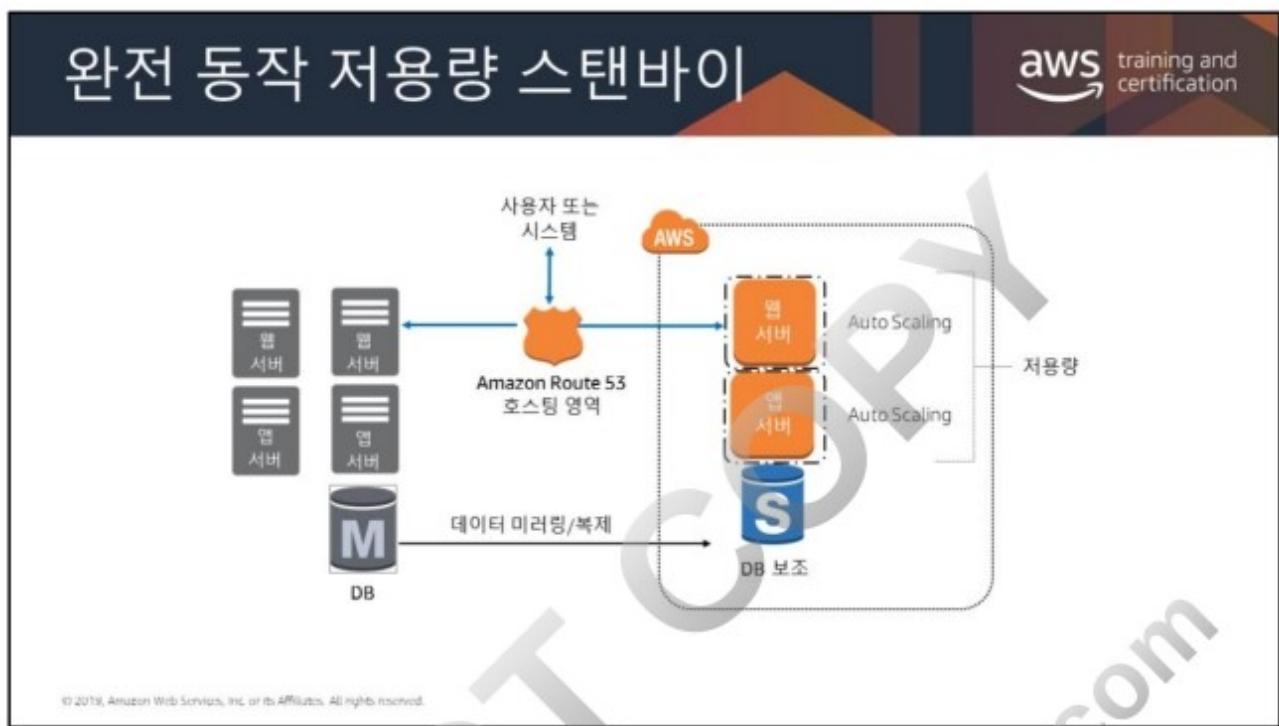
재해 발생 시

- 복제된 핵심 데이터 세트 주위의 리소스를 자동으로 준비합니다.
- 필요에 따라 시스템을 확장하여 현재 프로덕션 트래픽을 처리합니다.
- 새로운 시스템으로 전환합니다.
 - AWS를 가리키도록 DNS 레코드를 조정합니다.

목표

- RTO: DR 필요성을 감지하고 자동으로 대체 시스템을 확장하는데 소요되는 시간
- RPO: 복제 유형에 따라 다름

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



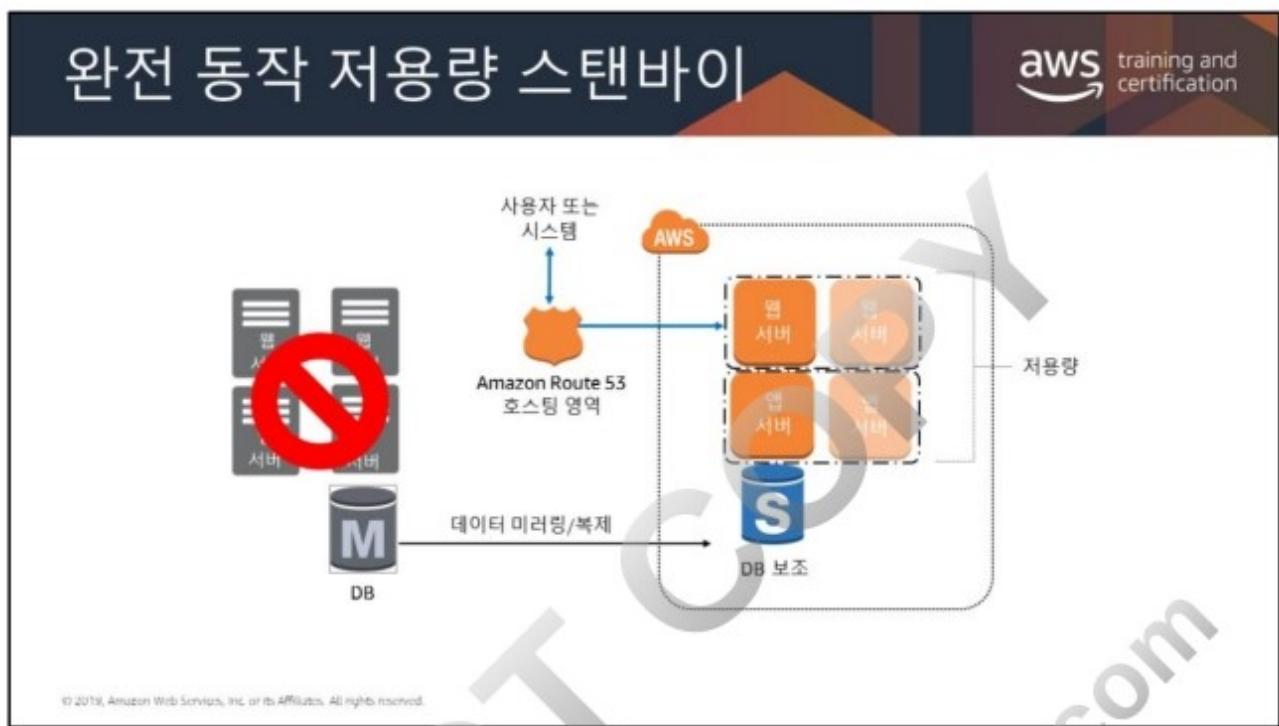
저용량 스탠바이는 파일럿 라이트의 다음 수준과 유사합니다. 웜 대기(*warm standby*)라는 용어는 완전한 기능 환경의 축소 버전이 항상 클라우드에서 실행되는 DR 시나리오를 설명하는데 사용됩니다. 웜 대기 솔루션은 파일럿 라이트 요소와 준비 과정을 확장합니다. 일부 서비스는 항상 실행 중이므로 이 솔루션을 사용하면 복구 시간을 더욱 단축할 수 있습니다. 비즈니스 크리티컬한 시스템을 확인한 후 AWS상에 이러한 시스템을 모두 복제하고 상시 접속되도록 합니다.

이러한 서버는 규모가 제일 작은 Amazon EC2 인스턴스에서 실행할 수 있습니다. 이 솔루션은 최대 프로덕션 부하를 처리할 정도로 규모가 확장되지는 않지만 기능은 온전하게 작동합니다. 이 솔루션은 테스트, 품질 보장 및 내부 사용 등 프로덕션 이외의 작업에 사용할 수 있습니다.

재해 시, 이 시스템은 프로덕션 부하를 처리할 수 있도록 신속하게 규모를 확장합니다. AWS에서는 로드 밸런서에 더 많은 인스턴스를 추가하거나 작은 용량의 서버가 더 큰 Amazon EC2 인스턴스 유형에서 실행되도록 크기를 조정함으로써 규모를 확장할 수 있습니다. 이전 단원에서 설명한 것처럼 수평적 확장은 수직 확장보다 우선적으로 적용됩니다.

상기의 다이어그램에서는 주 시스템과 AWS에서 실행되는 저용량 시스템이라는 두 개의 시스템이 실행됩니다. Amazon Route 53을 사용하여 주 시스템과 클라우드 시스템 간에 요청을 분산합니다.

DO NOT COPY
zlagusdbs@gmail.com



기본 환경을 사용할 수 없는 경우, Amazon Route 53은 보조 환경으로 전환됩니다. 보조 시스템은 기본 시스템으로부터 장애 조치가 실행될 경우 용량을 자동으로 확장하도록 설계되어 있습니다.

완전 동작 저용량 스탠바이

aws training and certification

이점

- 언제라도 서비스 트래픽 일부를 처리할 수 있음
- 비용 절감(완전 DR 대비 IT 리소스가 적음)

준비

- 파일럿 라이트와 유사
- 모든 필수 구성 요소가 24/7 실행되지만, 서비스 트래픽에 맞게 확장되지는 않음
- 모범 사례: 지속적인 테스트
 - 서비스 트래픽의 통계적 하위 집합을 DR 사이트로 "드문드문" 보냄

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

이 패턴에서는 활성 시스템이 실행되므로 비용이 더 많이 듭니다.

완전 동작 저용량 스탠바이



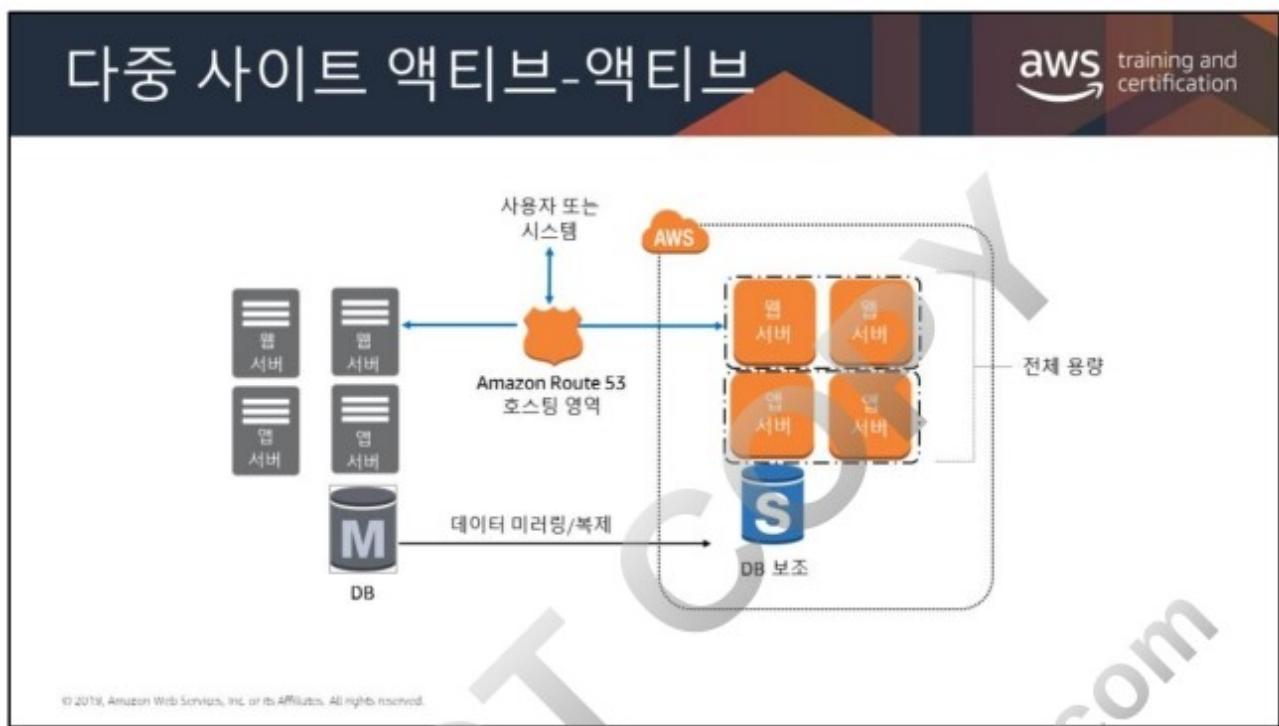
재해 발생 시

- 가장 중요한 서비스 부하에 대해 즉시 장애 조치
 - DNS 레코드가 AWS를 가리키도록 DNS 레코드를 조정
 - 모든 서비스 부하를 처리하도록 시스템을 자동 확장

목표

- RTO: 중요 로드의 경우는 장애 조치를 취하는 데 소요되는 시간, 나머지 로드의 경우는 추가 확장에 소요되는 시간
- RPO: 복제 유형에 따라 다름

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



다음 수준의 재해 복구는 온프레미스 시스템과 동시에 AWS에서 완벽하게 작동하는 시스템을 실행하는 것입니다.

다중 사이트 솔루션은 액티브-액티브 구성으로 기존의 현장 인프라뿐 아니라 AWS에서도 실행됩니다. 사용하는 데이터 복제 방법은 선택한 복구 시점에 의해 결정됩니다.

Amazon Route 53과 같은 가중치 기반 라우팅을 지원하는 DNS 서비스를 사용하면 프로덕션 트래픽을 동일한 애플리케이션이나 서비스를 제공하는 다른 사이트로 라우팅할 수 있습니다. 일정량의 트래픽은 AWS의 인프라로 전송되며, 나머지는 현장 인프라로 전송됩니다.

현장 재해 시, DNS 가중치를 조정하여 모든 트래픽을 AWS 서버로 보낼 수도 있습니다. 최대 프로덕션 부하량을 처리할 수 있도록 AWS 서비스 용량을 신속하게 증가시킬 수 있습니다. Amazon EC2 Auto Scaling을 사용하면 이 프로세스를 자동으로 실행할 수 있습니다. 기본 데이터베이스 서비스의 장애를 감지하고 AWS에서 실행되는 병렬 데이터베이스 서비스로 이관되기 위해 몇 가지 애플리케이션 로직이 필요할 수 있습니다.

이 시나리오의 비용은 정상 운영 중 AWS가 처리하는 프로덕션 트래픽의 양에 따라 결정됩니다. 복구 단계에서는 전체 재해 복구 환경이 필요한 기간 동안 트래픽을 실제로 사용한 만큼만 비용을 지불하면 됩니다. "항시 작동하는" AWS 서버에 대해 Amazon EC2 예약 인스턴스를 구입하면 비용을 더욱 줄일 수 있습니다.

DO NOT COPY
zlagusdbs@gmail.com

다중 사이트 액티브-액티브



이점

- 언제라도 모든 프로덕션 로드를 처리할 수 있습니다.

준비

- 저용량 스탠바이와 유사합니다.
- 프로덕션 로드에 따라 완전히 확장/축소

재해 발생 시

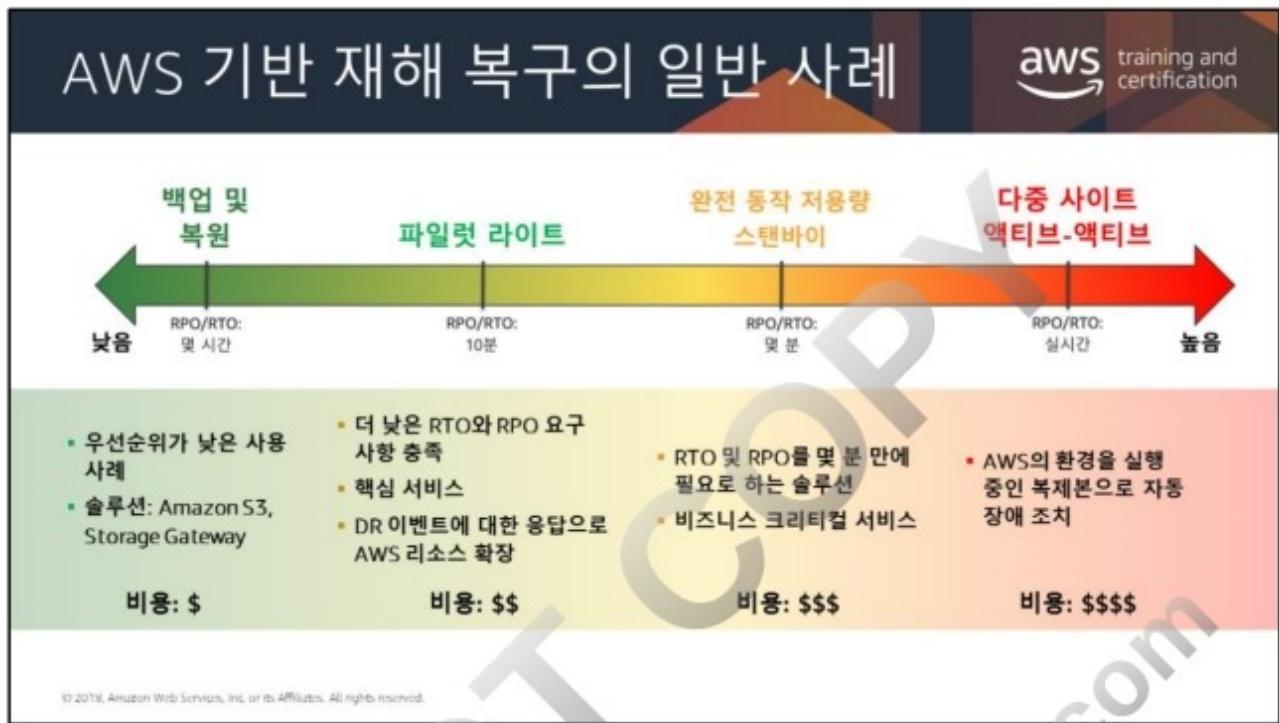
- 모든 프로덕션 로드에 대해 즉시 장애 조치

목표

- RTO: 장애 조치에 소요되는 시간
- RPO: 복제 유형에 따라 다름

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

이 패턴은 모든 패턴 중에서 잠재적으로 가동 중단 시간이 가장 짧습니다. 더 많은 시스템이 실행되므로 비용도 더 많이 듭니다.



애플리케이션은 복잡성의 스펙트럼에 배치할 수 있습니다. 심각한 재해가 발생하더라도 비즈니스 연속성을 통해 중요한 비즈니스 기능이 계속 빠르게 작동하거나 복구됩니다.

다음 슬라이드에서는 AWS의 사용을 강조하고 AWS와 기존 DR 방법 (RTO/RPO가 가장 높은 순위에서 가장 낮은 순위로 정렬됨)을 비교하는 4가지 DR 시나리오에 대해 간략히 설명합니다.

- 백업 및 복원
- 파일럿 라이트
- 완전 동작 저용량 스탠바이
- 다중 사이트 액티브-액티브

위 그림은 DR 이벤트가 발생한 후 얼마나 빨리 사용자에게 시스템을 사용 가능한 상태로 다시 제공할 수 있는지에 따라 4가지 시나리오로 분류한 스펙트럼입니다.

AWS를 사용하면 이러한 DR 전략 각각을 저렴한 비용으로 운영할 수 있습니다. 위의 예는 단편적인 예일 뿐이며, 이러한 예를 변형하거나 서로 조합하여 사용하는 것도 가능합니다. 애플리케이션을 이미 AWS에서 실행 중인 경우, 여러 리전을 적용할 수 있으며 동일한 DR 전략도 계속 적용할 수 있습니다.



단순하게 시작하여 차차 보강

- 첫 번째 단계로 AWS에 백업
- RTO/RPO를 점진적으로 개선하기 위해 지속적으로 노력

소프트웨어 라이선스 문제가 있는지 확인

DR 솔루션에 대한 연습

- “게임 데이” 시나리오를 연습. 이 시나리오에서는 크리티컬한 시스템을 오프라인 또는 전체 리전에 걸쳐 테스트합니다. 전체 인스턴스 집합에 충돌이 발생할 경우 어떻게 해야 하나요?
- 백업, 스냅샷, AMI 등이 작동하는지 확인
- 모니터링 시스템을 모니터링

한 가지 추가 사항은..



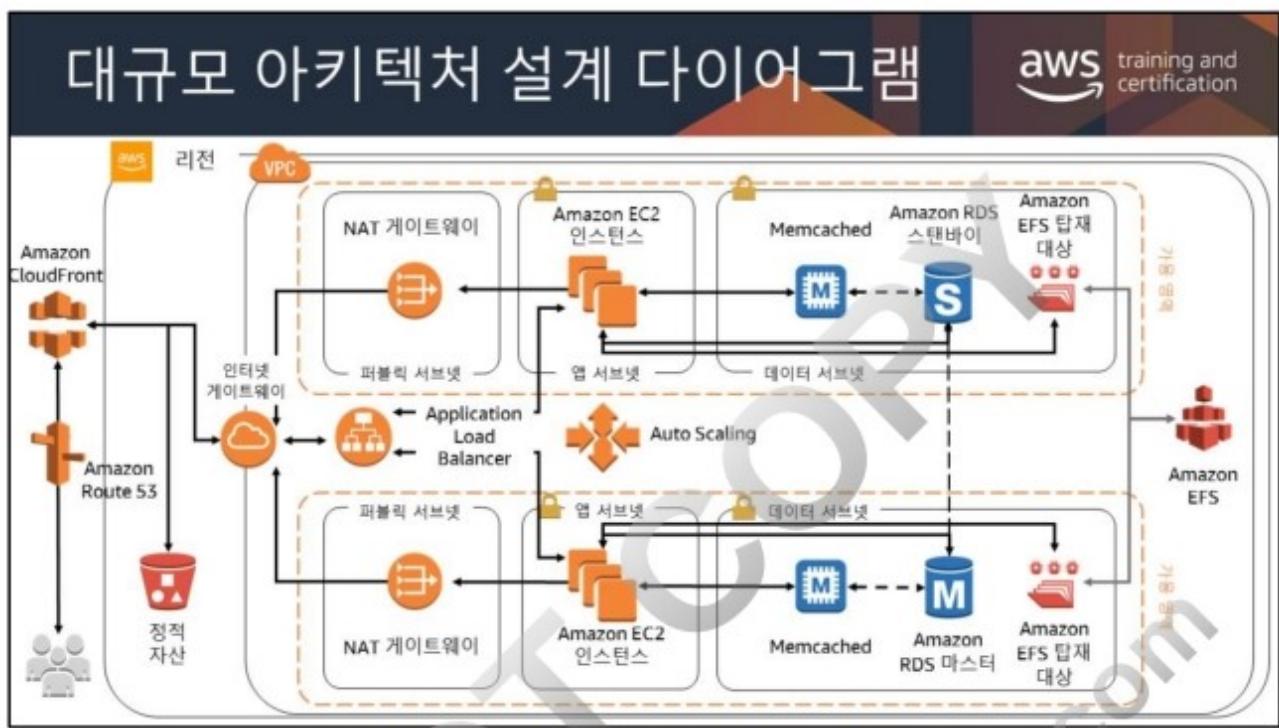
귀하의 의견은 매우 중요합니다!

- <https://aws.training>에 로그인합니다.
- My Transcript(내 스크립트)를 선택한 다음 Archived(아카이브) 탭을 클릭합니다.
- AWS 기반 아키텍처 완료 교육을 찾은 다음 Evaluate(평가)를 클릭합니다.

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



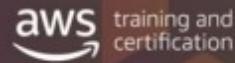




수업이 끝나면 이 아키텍처 다이어그램의 모든 구성 요소를 이해할 수 있습니다.
또한 마찬가지로 규모가 크고 강력한 자체 아키텍처 솔루션을 구축할 수
있습니다.



Q1



이 아키텍처가 해결 중인 문제에 대해 **가장 적합한 리소스를 사용하고 있습니까?**

- 더 적합한 **인스턴스 유형**이 있습니까?
- 관리형 서비스를 사용해야 합니까?

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

조직이 데이터 센터에 배치할 하드웨어를 구매할 때 흔히 예산이 책정되어 있으며 미래 지향적으로 구매를 결정합니다. 오늘 최고 성능을 자랑하는 하드웨어도 대략 18개월이면 쓸모가 없어질 것입니다. 그러므로 한 번의 구매로 미래의 요구 사항까지 대비하기 위해 당장 필요한 것보다 상향하여 구매할 수 밖에 없습니다.

따라서 데이터 센터 구축을 고려할 때 "지금 당장 필요한 것이 무엇인가?"를 물어야 합니다. 규모를 늘리거나 줄여야 한다면 그럴 수 있습니다.

또 하나의 방법은 "실제로 사용하지 않는 용량에 비용을 지불할 여력이 있는가?"를 묻는 것입니다.

Amazon RDS 같은 관리형 서비스는 모든 요구 사항을 미리 따져보지 않아도 AWS에서 리소스를 구축할 수 있게 해줍니다. 예를 들어 개발자가 Microsoft SQL Server 데이터베이스를 요청했는데 여러분이 이 소프트웨어를 설치해본 적이 없다고 한다면 상당한 시간과 노력이 필요할 것입니다. 그 대신, 어떤 종류의 데이터베이스가 필요한지 파악하여 AWS가 대신 구축하게 하십시오. 데이터베이스, 데이터, 액세스 권한은 여러분이 완벽하게 제어합니다. AWS가 하는 모든 것은 여러분이 설계나 구현 같은 창의적인 작업에만 집중할 수 있게 하는 것입니다.

Q2



이 아키텍처는 **복원력**이 있습니까?

- 단일 장애 지점이 존재합니까?
- 장애에서 **복구**할 수 있습니까?
- 자체 **복구 기능**이 있습니까?

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

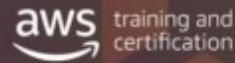
Amazon에서 모범 사례 중 하나는 프로세스의 끝에서 시작하여 그 과정을 되짚어보는 것입니다. 우리는 스스로 묻습니다. 어디에서 장애가 발생할 것인가? 장애가 발생한다면 어떻게 될 것인가?

예, 여러분이 해결할 수 있습니다. 하지만 반드시 여러분이 해결해야 합니까? 대신, 잘못된 비트를 삭제하고 새로운 비트를 캐시시오. 여러분은 재해에 대한 사후 보고를 할 수 있고 또 해야 합니다. 하지만 인스턴스, 컨테이너 또는 유사한 구성 요소가 가동 중지되었다면 먼저 교체한 다음 원인을 파악하십시오.

AWS를 사용하면 프로덕션 환경에 영향을 주지 않고 전체재해 대응 연습을 실시할 수 있습니다. 예, 비용은 몇 달러에 불과하니까 두 번째 데이터 센터를 구축하는 것과는 비교할 수 없습니다.

여러분은 창의적인 작업을 수행하기 위해 채용된 것입니다. 그런데 자동화할 수 있는 작업을 수동으로 처리하는 라 시간을 소비할 이유가 있을까요? AWS를 사용하면 스스로 모니터링하고, 필요에 따라 장애가 발생한 요소를 삭제한 후 사용자에게 무엇이 발생했는지 보고할 수 있는 환경을 구축할 수 있습니다.

Q3



이 아키텍처에서 구성 요소 간 긴밀한 종속성을 제거할 수 있습니까?

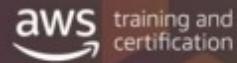
- 마이크로 서비스
- 결합 해제

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

마이크로 서비스는 아키텍처의 안정성과 가용성을 높이기 위한 최선의 방법 중 하나입니다. 이 접근 방식을 사용하면 애플리케이션 구성 요소 간 종속성을 축소하여 혁신 속도를 개선할 수 있습니다. IT 팀이 비동기식으로 작업하여 전체 시스템에 영향을 미치지 않고 기능 업데이트를 제공할 수 있습니다.

이러한 종류의 접근 방식을 선택하기 위해서는 시스템 작동 방식에 대한 새로운 사고 방식이 필요할 수 있습니다. 모놀리식 시스템은 상당한 매몰 비용이 발생하지만, 시간을 들여 각 구성 요소를 분리한 후 어떻게 작동하는지 살펴본다면 클라우드에서는 이러한 비용을 방지할 수 있습니다.

Q4



이 아키텍처를 효과적으로 확장할 수 있습니까?
(100명의 사용자 -> 100만 명의 사용자)

- 최소 인프라
- Auto Scaling

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

여러분은 가능한 한 규모가 크고 속도가 빠르며 성능이 강력한 시스템을 원할 수 있겠지만, 여러분이 시스템을 최대 성능까지 사용하지 않는다면 낭비일 뿐입니다.

구체적인 수치를 파악하는 데 충분한 데이터 포인트를 수집하는 데 약 3개월이 걸립니다. 따라서 실제로 얼마의 용량이 필요한지 파악하려면 먼저 필요한 것보다 큰 용량의 시스템에서 시작하십시오. 그런 다음 실제 시스템 사용량을 보면서 환경의 "적정 크기"를 판단할 수 있습니다.

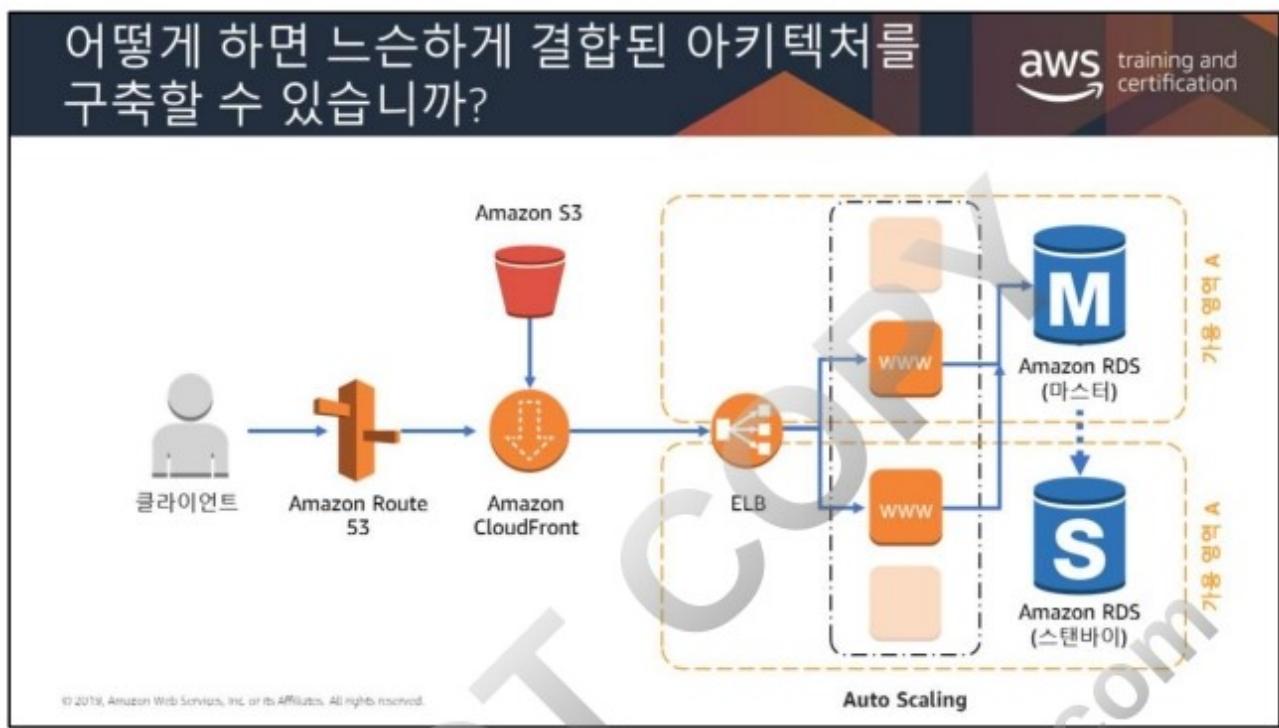
온프레미스 데이터 센터를 운영 중이라면 새로운 리소스 가동은 데이터 센터 자체와 관련된 모든 라인 항목에 의해 흡수됩니다. AWS에서 새로운 워크로드를 실행할 경우 이와 관련된 비용이 발생합니다. 다행히 AWS에서 비용은 투명합니다. 여러분은 새 인스턴스를 시작하는 것이 물리적 서버를 구축하는 것보다 훨씬 저렴하다는 것을 알 수 있을 것입니다.

IT 세계에서는 두 가지 유형의 장애가 있습니다.

- 여러분이 하드웨어 또는 소프트웨어에 지정된 한계를 벗어난 무언가를 요청했기 때문에 어딘가에서 장애가 발생합니다.
- 여러분의 제품이 너무 성공적이어서 사용자의 요청이 하드웨어 및 소프트웨어의 지정된 한계를 초과합니다.

Auto Scaling은 비용을 수반합니다. 하지만 이 기능이 없다면 소프트웨어가 로드를 처리하지 못하고 다운될 것입니다. 소프트웨어를 계속 실행하기 위해 충분한 투자 가치가 있습니다.

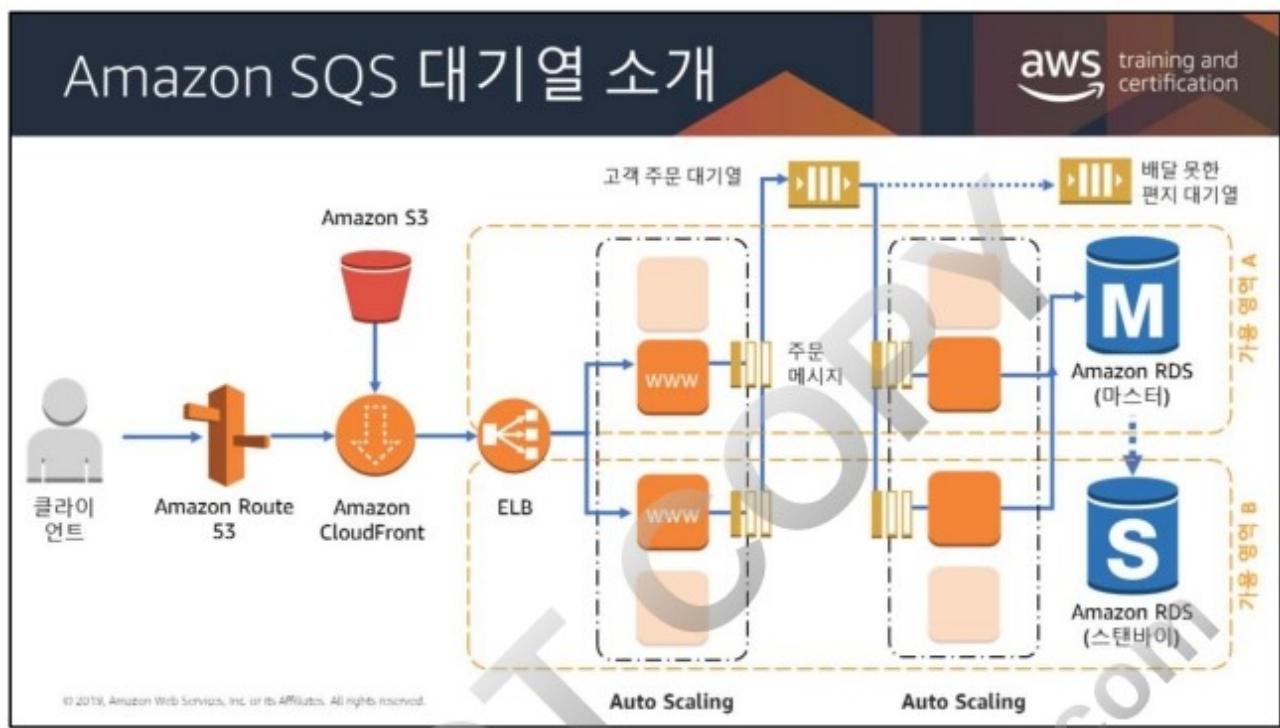




이 예제에서는 애플리케이션이 주문 데이터를 처리 및 유지해야 하고 인기 아이템에 대한 트래픽 증가도 처리해야 합니다.

주문 처리 워크플로우의 한 가지 잠재적인 취약점은 해당 주문을 데이터베이스에 저장하는 데 있습니다. 기업은 모든 주문이 데이터베이스에 계속 유지되는 것으로 기대합니다. 그러나 잠재적 교착, 경합 조건 또는 네트워크 문제가 발생할 경우, 해당 주문을 계속 유지할 수 없게 됩니다. 결국 해당 주문은 복원을 위한 어떤 수단도 없이 손실됩니다.

우수한 로깅 기능을 사용한다면 언제 오류가 발생했는지, 어떤 고객의 주문에 장애가 발생했는지 식별할 수 있을 것입니다. 그렇다고 거래를 "복원"할 수 있는 것은 아닙니다. 이 정도면 그 고객은 더 이상 여러분의 고객이 아닐 것입니다.



이 시나리오에서 어떻게 비용을 절감할 수 있습니까?

aws training and certification

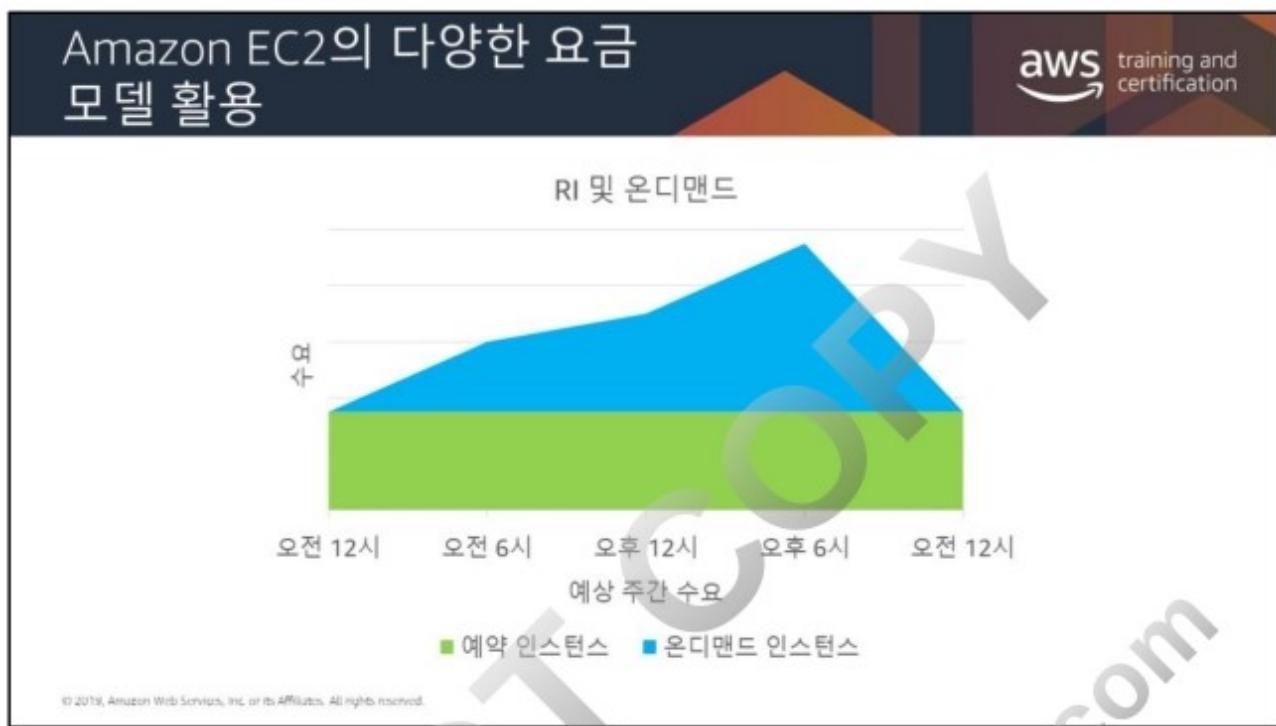
클라이언트 트래픽 볼륨:

- 근무 시간 중 트래픽 급증
- 항상 일관된 기본 수준 트래픽
- 온디맨드 인스턴스 사용

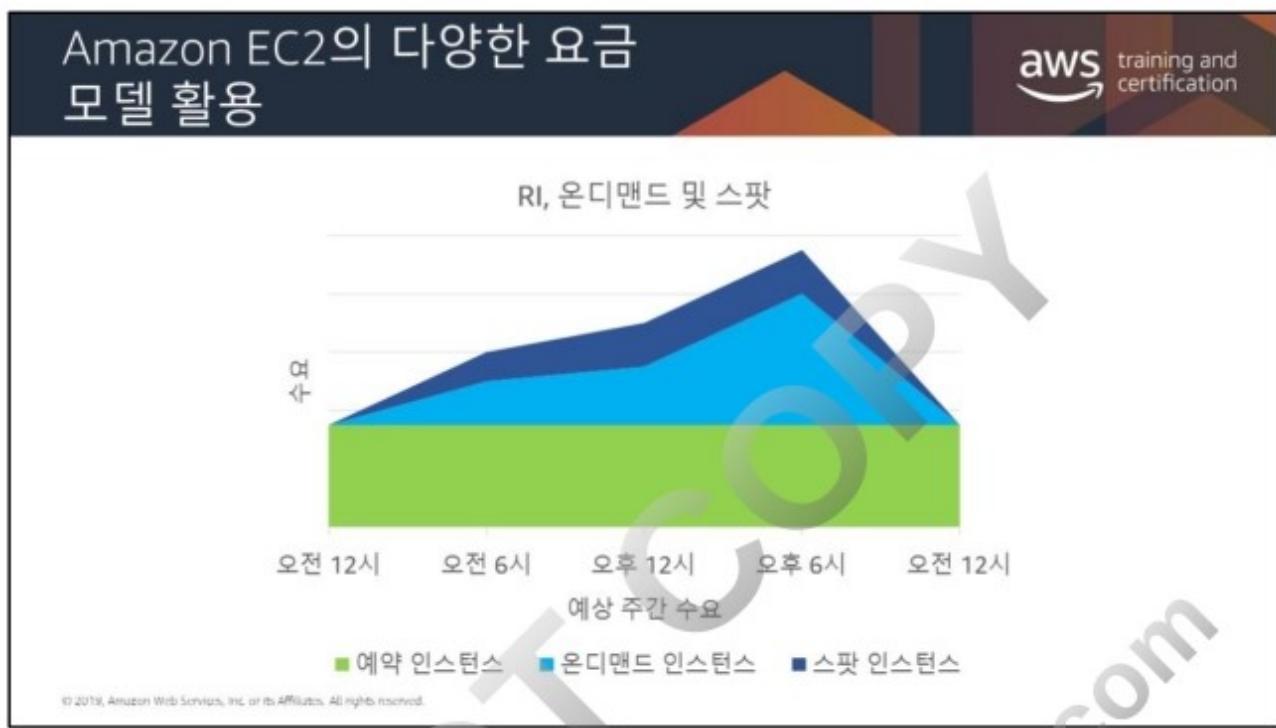
예상 주간 수요

■ 온디맨드 인스턴스

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



이 차트는 예약 인스턴스와 온디맨드 인스턴스를 함께 사용하여 시간이 지나면서 변동하는 수요를 수용하는 매우 기본적인 방법을 보여줍니다. 이 예제에서는 여러 예약 인스턴스를 구매하여 실행하고 있습니다. 하지만 시간이 지나면서 수요가 증가함에 따라 인스턴스 추가에 대한 요구도 증가합니다. 이 고객은 저녁이 되어 필요 없어지면 종료할 수 있는 온디맨드 인스턴스로 용량을 보충합니다.



두 번째 차트에서는 이 고객이 좀 더 복잡한 접근 방식을 택하여 3가지 요금 모델을 모두 활용하려고 시도합니다. 이 경우에는 일부 인스턴스를 먼저 온디맨드로 보충하고, 추가로 필요한 용량은 스팟 인스턴스로 채웁니다. 이렇게 하면 온디맨드 대신 스팟을 사용하여 비용을 절감할 수 있지만, 입찰에 실패하면 해당 인스턴스를 잃게 되므로 예상치 못한 인스턴스 종료가 발생할 수 있습니다. 이는 고객의 데이터 손실 또는 용량 부족으로 이어질 수 있습니다. 즉, 이러한 모델은 갑작스러운 인스턴스 종료가 허용되고 적절하게 처리될 수 있는 상황에서만 구현해야 합니다.

2017년에 도입된 새로운 요금 모델에서는 스팟 인스턴스에 대한 입찰이 더 이상 필요하지 않습니다. 시작된 인스턴스에 대해 현재 시간에 적용되는 스팟 가격을 지불하면 됩니다. 스팟 가격은 예측 가능하고, 자주 업데이트되지 않으며, Amazon EC2 예비 용량의 공급과 수요에 따라 결정됩니다. 또한 중단의 영향을 줄이고 스팟 인스턴스를 최적화하려면 여러 용량 풀에서 애플리케이션을 다양화하고 실행합니다. 모든 리전의 각 가용 영역에 있는 각 인스턴스 패밀리와 각 인스턴스 크기는 별도의 스팟 풀입니다. RequestSpotFleet API를 사용하여 자동으로 수천 개의 스팟 인스턴스를 시작하고 리소스를 다양화할 수 있습니다.

중단의 영향을 더욱 줄이기 위해서는 용량을 더 이상 사용할 수 없을 때 인스턴스를 종료하는 대신 중지하거나 최대 절전 모드로 전환하여 중단 알림에 대응하도록 스팟 인스턴스와 스팟 플릿을 설정할 수도 있습니다.

스팟 인스턴스 사용에 대한 자세한 내용은 다음을 참조하십시오.

<https://aws.amazon.com/ec2/spot/getting-started/>

스팟 요금에 대한 자세한 내용은 다음을 참조하십시오.

<https://aws.amazon.com/blogs/compute/new-amazon-ec2-spot-pricing/>

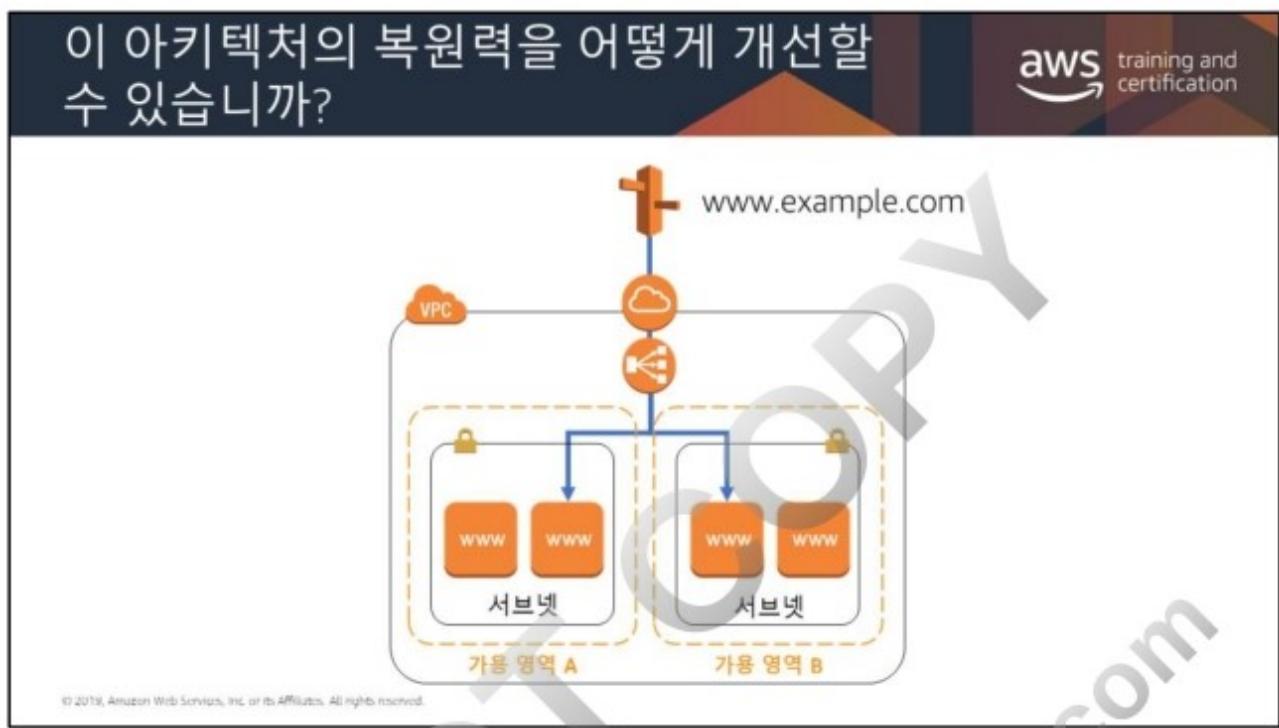
3가지 모델을 모두 함께 활용한 AWS 고객 중 하나가 Pinterest입니다. 자세한 내용은 <http://www.allthingsdistributed.com/2012/08/tco-and-return-on-agility.html>을 참조하십시오.

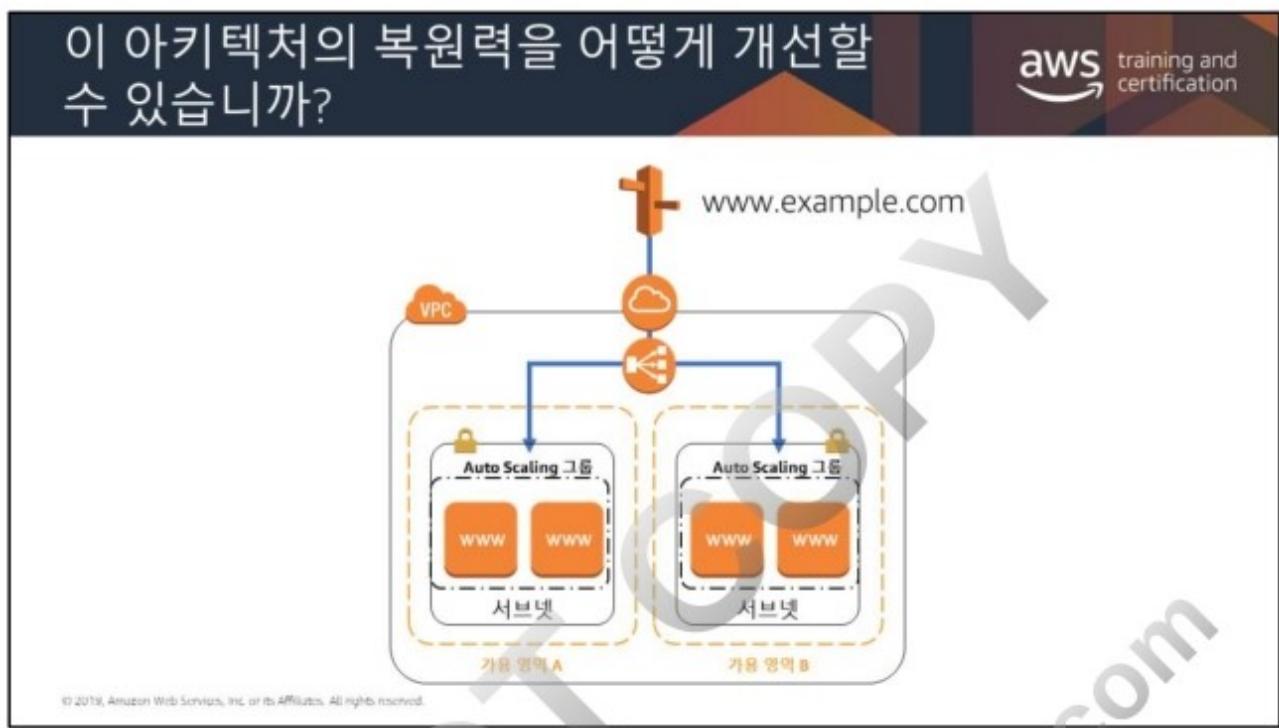
이 아키텍처의 복원력을 어떻게 개선할 수 있습니까?

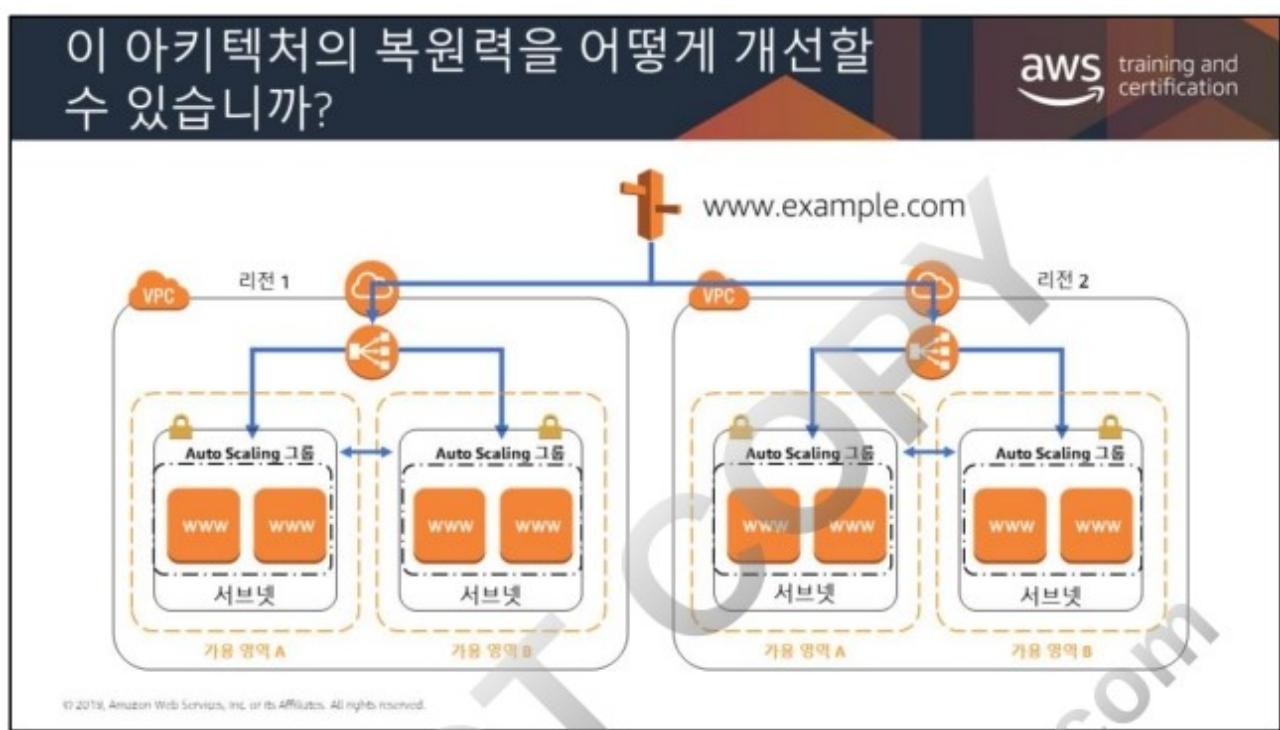
aws training and certification

The diagram illustrates a network architecture within a Virtual Private Cloud (VPC). At the top, a URL 'www.example.com' is connected to an orange cloud icon representing an Application Load Balancer (ALB). This ALB is connected to an orange VPC icon. Below the VPC icon is a network switch icon. Two arrows point from the switch to two orange boxes labeled 'www'. These two 'www' boxes are grouped under a grey box labeled '서브넷' (Subnet). A dashed orange line encloses the entire subnet, which is labeled '가용 영역 A' (Availability Zone A) at the bottom. The entire VPC structure is contained within a larger dashed orange border.

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

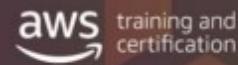






참고: Auto Scaling 그룹은 여러 가용 영역에 걸쳐 있을 수 있습니다.

인스턴스 유형은 무엇입니까?



클라이언트에 기계 학습을 사용하여 사용자가 제출한 이미지에 상표 등록된 로고가 포함되어 있는지 여부를 확인하는 웹 애플리케이션이 있습니다.

웹 서버로 어떤 유형의 인스턴스를 추천하시겠습니까?

백엔드 기계 학습을 위해 어떤 유형의 인스턴스를 추천하시겠습니까?

<https://aws.amazon.com/ec2/instance-types/>를 자유롭게 사용

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

인스턴스 유형에 대한 자세한 내용은 <https://aws.amazon.com/ec2/instance-types/>를 참조하십시오.



AWS Well Architected Tool은 사용자가 콘솔에서 최신 AWS 아키텍처 모범 사례를 사용자의 워크로드와 비교하고 검토할 수 있는 무료 도구입니다. 이는 AWS Well-Architected Framework를 기반으로 합니다. AWS 솔루션 아키텍처 팀이 수만 개의 워크로드를 검토하는데 사용했습니다.

<https://aws.amazon.com/well-architected-tool/>