



클라우드로 AWS 마이그레이션하는 동안 COTS 및 사내 애플리케이션 리플랫폼

AWS 권장 가이드



AWS 권장 가이드: 클라우드로 AWS 마이그레이션하는 동안 COTS 및 사내 애플리케이션 리플랫폼

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

소개	1
목표 비즈니스 성과	2
리플랫폼 환경 선택	3
지원되지 않는 OSs에서 실행되는 애플리케이션 구성 요소 리플랫폼	4
지원되지 않는 OSs 또는 애플리케이션 서버 교체	4
COTS 애플리케이션용 OS 업그레이드	5
사내 애플리케이션의 OS 업그레이드	5
애플리케이션 라이브러리 및 종속 소프트웨어 리플랫폼	5
백엔드 데이터베이스 리플랫폼	6
COTS 애플리케이션을 위한 백엔드 데이터베이스 리플랫폼	6
사내 애플리케이션을 위한 백엔드 데이터베이스 리플랫폼	7
파일 공유 리플랫폼	8
로깅 및 모니터링 구성 요소 업데이트	9
애플리케이션 테스트 및 검증	10
지속적 OS 패치 적용 자동화	11
자동화 도구 및 인프라를 코드로 사용(IaC)	11
FAQ	12
언제 리호스팅 대신 리플랫폼을 선택해야 하나요?	12
COTS 애플리케이션을 오픈 소스 데이터베이스로 리팩터링할 수 있나요?	12
서버를 AWS 클라우드로 빠르게 리호스팅하는 데 사용할 수 있는 AWS 도구는 무엇입니까?	12
애플리케이션을 리플랫폼하는 데 사용할 수 있는 AWS 도구는 무엇입니까?	12
리소스	14
참조	14
도구	14
문서 기록	15
.....	xvi

AWS 클라우드로 마이그레이션하는 동안 COTS 및 사내 애플리케이션 리플랫폼

Anbu Selvan, Amazon Web Services(AWS)

2021년 3월([문서 기록](#))

이 가이드에서는 Amazon Web Services(AWS)클라우드에서 상용 off-the-shelf(COTS) 및 사내 애플리케이션을 리플랫폼할 때 집중해야 할 7가지 영역에 대해 설명합니다. 또한 이 가이드는 애플리케이션 구성 요소를 리플랫폼하는 데 도움이 되는 전략, 도구 및 AWS 서비스도 제공합니다. COTS 애플리케이션은 상용 시장(예:)에서 구매할 수 있는 서드 파티 애플리케이션입니다. [AWS Marketplace](#). 사내 애플리케이션은 조직에서 내부적으로 개발하고 사용합니다.

COTS 또는 사내 애플리케이션을 로 마이그레이션하기로 결정 AWS 클라우드한 후에는 사용할 7가지 일반적인 마이그레이션 전략(7R) 중 어느 것을 평가해야 하는지 평가해야 합니다. 이러한 전략은 리팩터링, 리플랫폼, 재구매, 리호스팅, 재배포, 유지, 사용 중지입니다. end-of-support(EOS) 날짜에 도달했거나 근접한 구성 요소 또는 데이터베이스를 사용하는 애플리케이션을 리플랫폼하는 것이 좋습니다. EOS는 공급업체가 제품에 대한 기술 지원을 철회하는 경우입니다. AWS 클라우드에서 애플리케이션을 리플랫폼하기로 선택한 경우 다음 기능을 활용할 수 있습니다.

- 를 사용하여 현재 [위치 운영 체제\(OS\) 업그레이드](#)를 자동화합니다 AWS Systems Manager.
- 스냅샷 스토리지 볼륨을 사용하여 Amazon Elastic Compute Cloud(AMIs)를 빠르게 생성할 수 있습니다. Amazon EC2
- 프라이빗 서브넷을 생성하여 오래된 운영 체제(OSs).
- 고속 네트워킹을 사용하여 리플랫폼을 테스트하기 위해 프로덕션 환경을 빠르게 복제합니다.
- 추가 온프레미스 하드웨어를 사용하지 않고 온디맨드 EC2 인스턴스로 별도의 애플리케이션 스택을 빠르게 설정합니다.

AWS 클라우드에서 사용할 수 있는 이러한 기능 및 기타 기능을 활용하려면 먼저를 사용하여 애플리케이션을 리호스팅하는 것이 좋습니다. [AWS Transform MGN](#). 그런 다음 AWS 클라우드에서 애플리케이션을 업그레이드할 수 있습니다. 다음 목록은 애플리케이션을 리플랫폼해야 하는 경우의 예를 제공합니다.

- 애플리케이션의 OS, 런타임(예: [Apache Tomcat](#), [JBoss](#) 또는 [Oracle WebLogic Server](#)), 데이터베이스 또는 런타임 구성 요소(예: Java, Python 또는 Perl)에 대한 지원이 더 이상 제공되지 않습니다.

- 애플리케이션은 복원력이 향상되고 장애(예: 소프트웨어 버그 또는 인프라 문제)로부터 자동으로 복구되어야 합니다.
- 새로운 고객 세그먼트 또는 증가된 로드를 지원하려면 새로운 애플리케이션 기능이 필요합니다.
- 애플리케이션은 불안정하며 운영 안정성을 높이기 위해 개선이 필요합니다.

리플랫폼 여정을 시작하기 전에 애플리케이션 기능의 대안을 탐색해야 합니다. 예를 들어, 이를 독립 소프트웨어 공급업체(ISV)의 서비스형 소프트웨어(SaaS) 솔루션으로 대체할 수 있는지 여부를 평가해야 합니다. [AWS Lambda](#), [Amazon Cognito](#), Amazon [Amazon MQ](#), , [AWS Glue](#) [Amazon Quick](#) 또는 [Amazon Aurora](#)와 같은 AWS 서비스를 사용하여 애플리케이션 기능을 재구축할 수도 있습니다.

이 가이드는 IT 관리자, 애플리케이션 소유자, 아키텍트, 기술 책임자 및 프로젝트 관리자를 위한 것입니다. 이 가이드는 AWS 클라우드에서 COTS 및 사내 애플리케이션을 리플랫폼할 때 중점을 두어야 할 다음 7가지 영역을 제공합니다.

- [리플랫폼 환경 선택](#)
- [지원되지 않는 OSs에서 실행되는 애플리케이션 구성 요소 리플랫폼](#)
- [백엔드 데이터베이스 리플랫폼](#)
- [파일 공유 리플랫폼](#)
- [로깅 및 모니터링 구성 요소 업데이트](#)
- [애플리케이션 테스트 및 검증](#)
- [지속적 OS 패치 적용 자동화](#)

목표 비즈니스 성과

AWS 클라우드에서 COTS 및 사내 애플리케이션을 리플랫폼한 후 다음 네 가지 결과를 기대해야 합니다.

- 지원되지 않는 소프트웨어 또는 OSs를 실행하는 레거시 애플리케이션의 보안 위험을 줄입니다.
- 비용이 많이 드는 필수적이지 않은 데이터베이스 에디션을 제거하거나 오픈 소스 데이터베이스를 채택하여 전체 애플리케이션 소유권 비용을 절감합니다.
- AWS 관리형 데이터베이스(예: [Amazon Relational Database Service\(RDS\)](#) 또는 Aurora)를 사용하여 애플리케이션의 가용성과 안정성을 높임으로써 운영 오버헤드를 줄입니다.
- [Amazon CloudWatch](#) 모니터링 또는 Systems Manager 기반 OS 패치와 같은 클라우드 네이티브 자동화 및 모니터링 기능을 채택하여 레거시 애플리케이션의 복원력을 높입니다.

리플랫폼 환경 선택

[Amazon Elastic Block Store\(Amazon EBS\) 스냅샷](#) 또는 EC2 인스턴스 복제와 같은 기능을 사용하여 AMI를 생성하려면 AWS 클라우드에서 애플리케이션을 리플랫폼하는 것이 좋습니다. [EC2](#) 이러한 기능은 업그레이드 및 테스트 프로세스에 도움이 됩니다.

일반적으로 애플리케이션을 AWS 클라우드로 리호스팅하는 것으로 시작한 다음 리플랫폼 프로세스를 시작합니다. 그러나 취약성이 있는 애플리케이션을 마이그레이션하거나 지원되지 않는 소프트웨어 및 OSs 실행하는 경우 주의해야 합니다. 이러한 애플리케이션은 마이그레이션 및 향후 운영에 위험한 보안 취약성을 노출할 수 있습니다. 대신 [MGN](#)을 사용하여 애플리케이션 구성 요소를 송신이 제한된 프라이빗 서브넷에 복제하는 것이 좋습니다. 이 접근 방식은 워크로드를 격리한 다음 프로덕션 환경에 배포하기 전에 안전하게 리플랫폼하고 테스트할 수 있습니다.

조직에서 공식 EOS 날짜 이후 3~5년 동안 OS 패치를 연장하는 ISV에서 확장 OS 지원을 구매했을 수 있습니다. 이는 임시 조치이며 제품 업그레이드 또는 마이그레이션 타임라인을 COTS 애플리케이션 공급업체의 제품 릴리스 일정에 맞게 리팩터링하거나 조정할 수 있는 추가 시간을 제공합니다. 그러나 [AWS 전문 서비스](#) 또는 [AWS 마이그레이션 역량 파트너의](#) AWS 도구와 전문 지식을 사용하여 이러한 워크로드를 최신 OSs로 리플랫폼하는 것이 좋습니다. 이렇게 하면 비용이 많이 드는 확장 라이선스 계약을 방지하는 데 도움이 되며 마이그레이션을 더 빠르게 완료할 수 있습니다.

지원되지 않는 OSs에서 실행되는 애플리케이션 구성 요소 리플랫폼

지원되지 않는 OSs에서 실행되는 애플리케이션 구성 요소에 대한 리플랫폼 접근 방식은 애플리케이션 구성 요소마다 다릅니다. 다음 표에는 EOS에 도달한 애플리케이션 구성 요소에 사용할 수 있는 리플랫폼 옵션이 요약되어 있습니다.

애플리케이션 구성 요소	COTS 애플리케이션용 솔루션	사내 애플리케이션을 위한 솔루션
애플리케이션 서버	애플리케이션 공급업체가 권장하는 버전으로 업그레이드합니다.	최신 애플리케이션 서버 버전을 식별합니다. 업그레이드하기 전에 개발 환경에서 빌드하고 검증합니다.
OS	애플리케이션 공급업체가 권장하는 버전으로 업그레이드합니다.	최신 OS 버전을 식별합니다. 업그레이드하기 전에 개발 환경에서 빌드하고 검증합니다.
런타임 라이브러리	애플리케이션 공급업체가 권장하는 버전으로 업그레이드합니다.	최신 버전을 업그레이드하고 검증합니다.
기타 애플리케이션 구성 요소	애플리케이션 공급업체에 새 애플리케이션 바이너리를 요청합니다.	최신 OS, 런타임 및 애플리케이션 서버 버전으로 빌드합니다.

다음 섹션에서는 애플리케이션 구성 요소의 리플랫폼 접근 방식에 대한 자세한 정보를 제공합니다.

지원되지 않는 OSs 또는 애플리케이션 서버 교체

지원되지 않는 애플리케이션 서버(예: Apache Tomcat 6.0, Apache 2.2 또는 IIS 7.x)를 교체하는 경우 새 애플리케이션 서버 버전에 기본 OS 업그레이드가 필요할 수 있습니다. 지원되지 않는 대부분의 OSs는 Red Hat Enterprise Linux(RHEL) 버전 5 및 6, CentOS 버전 5 및 6 또는 Windows 2008 R2입니다. 이러한 OSs

1. 필요한 OS 버전으로 EC2 인스턴스를 시작합니다.
2. 필요한 애플리케이션 서버 버전을 설치합니다.
3. 사내 애플리케이션과 COTS 애플리케이션에는 두 가지 접근 방식이 있습니다.
 - 사내 애플리케이션 - 애플리케이션을 EC2 인스턴스에 재배포합니다.
 - COTS 애플리케이션 - 애플리케이션의 공급업체에 문의하여 필요한 OS 또는 애플리케이션 서버 버전에 대해 인증된 애플리케이션 바이너리를 요청합니다.

COTS 애플리케이션용 OS 업그레이드

대부분의 COTS 애플리케이션 공급업체는 Windows 2016 또는 RHEL 7을 지원합니다. 레거시 COTS 애플리케이션이 Windows 2016을 지원하지 않는 경우 Microsoft에서 제공하는 인플레이스 업그레이드 옵션을 사용하여 Windows 2008 R2에서 Windows 2012 R2로 인플레이스 업그레이드하는 것이 좋습니다. <https://docs.microsoft.com/en-us/windows-server/upgrade/upgrade-2008r2-to-2012r2> [AWS Systems Manager Automation 런북](#)을 사용하여 EC2 인스턴스에서 실행되는 Windows Server를 자동으로 업그레이드할 수도 있습니다. 애플리케이션의 공급업체에 문의하여 최신 OS 버전에 대한 소프트웨어를 인증하도록 요청하는 것이 좋습니다.

사내 애플리케이션의 OS 업그레이드

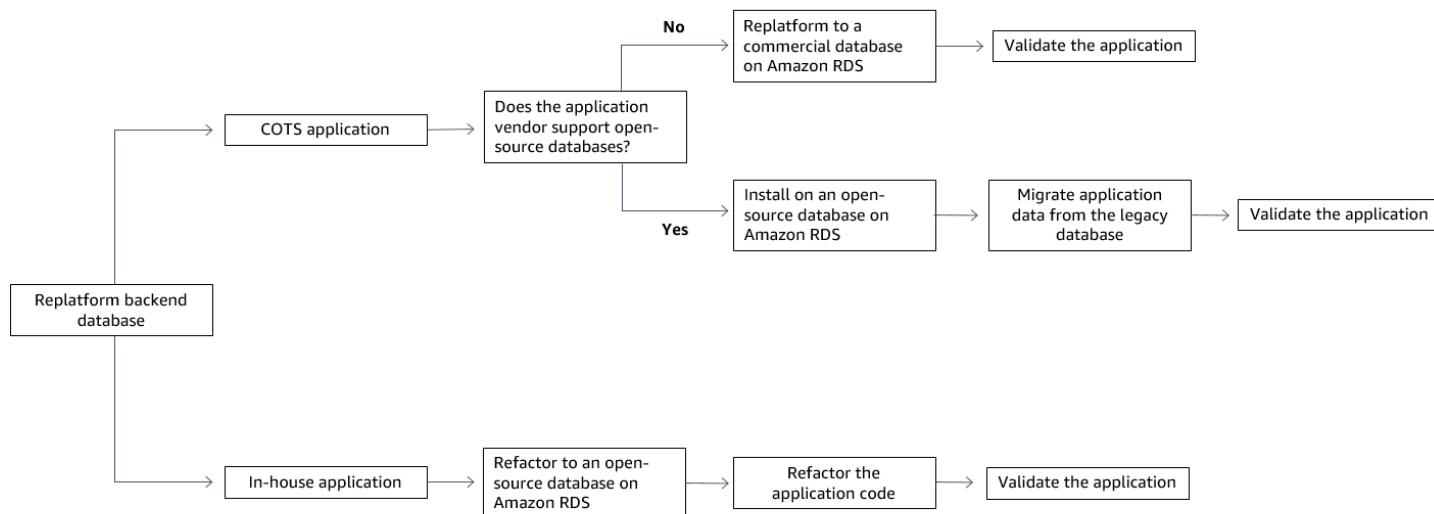
최신 OS 및 소프트웨어 런타임 버전(예: Java, C++, .NET 또는 Python)을 사용하여 사내 애플리케이션의 소프트웨어를 컴파일하고 재구축하는 것이 좋습니다. 그런 다음 프로덕션 환경으로 업그레이드하기 전에 기존 애플리케이션 환경을 복제하고, 기능을 수동으로 배포 및 검증하고, 빌드 환경을 최신 OS, 런타임 소프트웨어 구성 요소 및 라이브러리로 업데이트할 수 있습니다.

애플리케이션 라이브러리 및 종속 소프트웨어 리플랫폼

애플리케이션 라이브러리 및 종속 소프트웨어를 리플랫폼하는 접근 방식은 OSs에 대한 접근 방식과 비슷하지만 라이브러리만 업그레이드합니다. 그런 다음 애플리케이션의 기능을 테스트하고 사전 프로덕션 및 프로덕션 서버에 필요한 라이브러리를 복제합니다. 일반적으로 COTS 애플리케이션의 공급업체는 지속적인 소프트웨어 릴리스를 통해 애플리케이션 구성 요소에 필요한 업데이트를 처리합니다.

백엔드 데이터베이스 리플랫폼

백엔드 데이터베이스를 리플랫폼하는 접근 방식은 COTS와 사내 애플리케이션에 따라 다릅니다. 이는 소스 코드가 일반적으로 사내 애플리케이션에서만 사용 가능하기 때문입니다. 다음 그림은 애플리케이션의 백엔드 데이터베이스에 사용할 수 있는 리플랫폼 옵션을 보여줍니다.



다음 섹션에서는 COTS 또는 사내 애플리케이션에 속하는 백엔드 데이터베이스의 리플랫폼 접근 방식을 설명합니다.

COTS 애플리케이션을 위한 백엔드 데이터베이스 리플랫폼

COTS 애플리케이션이 오픈 소스 데이터베이스를 지원하는 경우 Aurora 데이터베이스를 사용하는 것이 좋습니다. 오픈 소스 데이터베이스를 사용하면 라이선스 비용을 절감할 수 있으며, [AWS Schema Conversion Tool \(AWS SCT\)](#) 및 [AWS Database Migration Service \(AWS DMS\)](#)와 같은 도구를 사용하여 마이그레이션 중에 가동 중지 시간을 최소화하면서 전환을 달성할 수도 있습니다.

COTS 애플리케이션이 오픈 소스 데이터베이스를 지원하지 않는 경우 Amazon RDS [for Oracle](#) 또는 Amazon RDS [for Microsoft SQL Server](#)와 같은 Amazon Relational Database Service(RDS)의 상용 데이터베이스로 리플랫폼하는 것이 좋습니다. 마이그레이션을 시작하기 전에 애플리케이션에서 사용하는 데이터베이스 기능을 평가하고 Amazon RDS에서 지원되는지 확인해야 합니다. 자세한 내용은 Amazon RDS 설명서의 [Microsoft SQL Server 데이터베이스 인스턴스에 대한 제한을 참조하세요](#).

나머지 데이터베이스 라이선스를 사용하고 EC2 인스턴스에서 자체 관리형 상용 데이터베이스를 실행할 수도 있습니다. 이 접근 방식을 선택하는 경우 데이터베이스의 공급업체에서 라이선스 확인 프로세

스를 시작하는 것이 좋습니다. 라이선스 확인 프로세스가 완료되면 애플리케이션의 필수 복구 시간 목표(RTO) 또는 복구 시점 목표(RPO)를 위해 Amazon EC2에서 자체 관리형 데이터베이스 솔루션을 설계해야 합니다.

마지막으로 SQL Server 데이터베이스를 사용하는 보안에 민감한 고성능 COTS 애플리케이션을 Amazon EC2 Linux 인스턴스에서 실행되는 SQL Server로 리플랫폼하는 것이 좋습니다. 이에 대한 자세한 내용은 [온프레미스 SQL Server Windows 워크로드를 Amazon EC2 Linux로 마이그레이션을 참조하세요](#).

사내 애플리케이션을 위한 백엔드 데이터베이스 리플랫폼

내부 애플리케이션의 백엔드 데이터베이스를 AWS 관리형 데이터베이스(예: [Amazon RDS for PostgreSQL](#), [Amazon RDS for MySQL](#), [Aurora](#) 또는 [Amazon DynamoDB](#))로 리플랫폼하여 데이터베이스 라이선스 비용을 줄이고 확장성을 높일 수 있습니다.

AWS 관리형 데이터베이스를 사용하면 데이터베이스의 반복 관리 작업(예: 백업 수행 또는 데이터베이스 및 OSs)을 줄일 수 있습니다. Amazon RDS 다중 AZ 배포를 사용하는 경우 데이터베이스 하드웨어 장애로 인한 중단을 방지하여 애플리케이션 가용성을 높일 수도 있습니다. 다중 AZ 데이터베이스는 다른 가용 영역에 지속적으로 복제되며 애플리케이션은 중단 중에 복제된 데이터베이스로 투명하게 장애 조치됩니다.

AWS DMS 및 AWS SCT 를 사용하여 상용 데이터베이스를 Aurora로 변환하고 Amazon RDS. AWS SCT automates 데이터베이스 스키마 변환 프로세스를 사용하고 온프레미스 데이터베이스에서 Amazon RDS로 데이터를 복제할 수 AWS DMS 있습니다. AWS DMS 또한 온프레미스 애플리케이션을 클라우드로 마이그레이션할 때 가동 중지 시간 전환을 최소화할 수 AWS 있습니다.

파일 공유 리플랫폼

사내 애플리케이션에서 NFS(Network File Systems) 또는 SMB(Server Message Block)와 같은 온프레미스 외부 파일 공유를 사용하는 경우 Amazon Simple Storage Service(Amazon S3) 및 [Amazon CloudFront](#)를 사용하도록 애플리케이션의 코드를 변경하는 것이 좋습니다. 그러나 블록 기반 액세스가 필요한 경우 [Amazon Elastic File System\(Amazon EFS\)](#) 또는 [Amazon FSx for Lustre](#)를 사용할 수 있습니다. Amazon EFS 및 Amazon FSx는 공유 파일 스토리지가 필요한 애플리케이션을 위한 AWS 옵션입니다.

COTS 애플리케이션의 경우 온프레미스 파일 공유에서 Linux 또는 FSx for Windows File Server의 Amazon EFS로 데이터를 마이그레이션하는 것이 좋습니다. [FSx](#) 온프레미스 파일 공유가 노출되는 것과 동일한 방식으로 블록 기반 파일 공유를 애플리케이션에 노출해야 합니다. Amazon EFS 및 Amazon FSx는 온프레미스 파일 공유가 제공한 것과 동일한 기능을 사용하여 NFS 및 SMB 기반 파일 공유를 제공합니다. 파일 공유에 Amazon EFS 또는 Amazon FSx를 사용하려면 COTS 애플리케이션에 대한 구성 변경만 필요합니다.

로깅 및 모니터링 구성 요소 업데이트

일부 레거시 환경에서는 인프라 및 애플리케이션 모니터링을 위해 중앙 집중식 로깅 및 모니터링 도구 (예: [Splunk](#), [SolarWinds](#) 또는 [Zabbix](#))를 사용합니다. 애플리케이션 지원 팀은 Secure Shell(SSH) 프로토콜 또는 원격 데스크톱 프로토콜(RDP)을 사용하여 모니터링하고 디버깅할 수도 있습니다. 이 수동적이고 반복적인 프로세스를 방지하려면 [CloudWatch](#)를 사용하여 AWS 클라우드에서 모니터링을 자동화할 수 있습니다.

CloudWatch 지표를 사용하여 인프라 및 CloudWatch 에이전트를 모니터링하여 애플리케이션 로그를 CloudWatch로 전송하는 것이 좋습니다. CloudWatch에서 애플리케이션 로그를 수신한 후 [CloudWatch 지표 필터](#)를 생성하고 [CloudWatch 경보](#)를 사용하여 애플리케이션 오류를 모니터링하고 지원 팀에 자동으로 알릴 수 있습니다.

또한 CloudWatch는 애플리케이션의 프로덕션 작업을 지속적으로 검토할 수 있는 운영 대시보드를 구축할 수 있는 도구를 제공합니다. 타사 중앙 집중식 모니터링 도구를 CloudWatch 및 기타 AWS 서비스와 통합할 수 있으므로 기존 운영 사례를 AWS 클라우드의 인프라 및 애플리케이션으로 확장하는 데 도움이 됩니다. 그러나 AWS 클라우드에서 애플리케이션을 운영하기로 선택한 경우 지원 또는 운영 팀을 재교육해야 할 수 있습니다. 이에 대한 자세한 내용은 AWS Cloud Adoption Framework(AWS CAF) 백서의 [운영 관점: 관리 및 규모 조정](#) 섹션을 참조하세요.

애플리케이션 테스트 및 검증

기능 및 성능 테스트는 애플리케이션의 리플랫폼 여정에서 중요한 부분입니다. 일반적으로 레거시 애플리케이션은 기능 세부 정보가 올바르게 문서화되지 않았거나 완전히 문서화되지 않았기 때문에 테스트에 애플리케이션 소유자의 지식에 의존합니다. 그러나 동작 및 자동 테스트를 사용하여 애플리케이션 사용 사례를 기록하는 것이 좋습니다. 이 접근 방식은 리플랫폼 전후에 애플리케이션의 기능을 빠르고 안정적으로 검증합니다. 자동 테스트 도구(예: [Selenium](#), [Tricentis](#) 또는 [Gatling](#))를 사용하여 기능 및 성능 테스트를 구축할 수 있습니다. 현재 애플리케이션 환경에서 기능 및 성능 테스트를 실행하여 기준 결과를 생성해야 합니다. 현재 애플리케이션 환경과 대상 애플리케이션 환경 간의 테스트 결과를 비교하여 수락 기준으로 사용할 수 있습니다.

고객 대면 애플리케이션에는 canary 테스트를 사용하는 것이 좋습니다. 카나리아 테스트는 프로덕션 환경에서 중요한 애플리케이션 워크플로를 주기적으로 테스트하고 지원 팀에 오류를 알립니다. 자세한 내용은 AWS Well-Architected Framework의 [Canary 배포](#) 섹션을 참조하세요.

지속적 OS 패치 적용 자동화

온프레미스 데이터 센터의 레거시 애플리케이션은 지속적인 OS 패치 및 소프트웨어 업데이트를 위해 수동 운영 프로세스에 의존하는 경우가 많습니다. 리플랫폼 여정 중에는 [Systems Manager Patch Manager](#) 또는 기타 자동 패치 프로세스를 사용하여 OS 패치 적용을 자동화하는 것이 좋습니다. 패치 관리자는 중앙 집중식의 일관된 프로세스를 제공하여 운영 인사이트를 수집하고 AWS 클라우드 및 온프레미스 리소스 모두에서 일상적인 운영 작업을 구현합니다.

프로덕션 환경에 사용되는 패치 적용 기간보다 빨리 개발 환경에 패치를 적용하는 것이 좋습니다. 이에 대한 자세한 내용은 OS 패치 적용 자동화를 위한 [패치 관리자 실행서](#)를 참조하세요. 또한 카나리아 테스트를 배포하여 사전 프로덕션 또는 프로덕션 환경에서 주요 애플리케이션 기능을 주기적으로 테스트하고 테스트가 실패하면 지원 팀에 알려야 합니다. 이렇게 하면 애플리케이션의 예상치 못한 종단을 방지할 수 있습니다.

자동화 도구 및 인프라를 코드로 사용(IaC)

애플리케이션의 리플랫폼 여정의 일환으로 [Chef](#), [Puppet](#) 또는 [Ansible](#)과 같은 구성 관리 도구를 사용하여 플랫폼 빌드를 자동화해야 합니다. 이러한 도구를 사용하면 애플리케이션 스택을 반복적으로 빌드할 수 있으며 스택의 구성을 포함하여 애플리케이션 인스턴스를 생성하는 단계를 공식화할 수 있습니다.

IaC 모범 사례를 사용하여 인프라를 프로비저닝하는 것이 좋습니다. 여기에는 [AWS Cloud Development Kit \(AWS CDK\)](#), [CloudFormation](#) 및 [Terraform](#)을 비롯한 몇 가지 옵션이 있습니다. Chef, Ansible 및 Puppet에도 사용 사례에 충분한 자동화를 제공할 수 있는 제한된 기능이 있습니다.

IaC 및 구성 관리 코드를 사용하는 반복 가능한 빌드를 사용하면 리소스를 재구축할 때 발생하는 오버헤드와 위험 없이 인프라를 테스트할 수 있습니다. 기존 인스턴스를 패치하고 업데이트하면 문제를 재현하고 식별하기 어려운 상태가 발생할 수 있습니다.

COTS 애플리케이션이 자동 설치를 지원하지 않는 경우 [AWS 파트너 네트워크\(APN\)](#)를 참조하는 것이 좋습니다. 이에 대한 자세한 내용은 AWS CAF 백서의 [플랫폼 관점: 애플리케이션 및 인프라](#) 섹션을 참조하세요.

COTS 및 사내 애플리케이션 리플랫폼 FAQ

이 섹션에서는 COTS 및 사내 애플리케이션 리플랫폼에 대해 일반적으로 제기되는 질문에 대한 답변을 제공합니다.

언제 리호스팅 대신 리플랫폼을 선택해야 하나요?

OS 공급업체가 보안 업데이트를 중단한 경우 데이터베이스가 아닌 구성 요소를 다시 플랫폼화해야 합니다(예: RHEL 5 및 Windows 2008). 지원되지 않는 OSs 경우 구성 요소를 격리된 프라이빗 서브넷으로 리호스팅하고 프로덕션 환경에 배포하기 전에 업그레이드할 수 있습니다.

데이터베이스 구성 요소의 경우 Amazon RDS의 상용 데이터베이스로 리플랫폼하는 것이 적절한 선택인지 평가해야 합니다. 데이터베이스의 공급업체에서 AWS 클라우드에서 현재 라이선스를 사용하도록 허용하는 경우 Amazon EC2에서 실행되는 자체 관리형 데이터베이스를 사용할 수도 있습니다.

COTS 애플리케이션을 오픈 소스 데이터베이스로 리팩터링할 수 있나요?

애플리케이션의 공급업체가 오픈 소스 데이터베이스를 지원하는 경우 COTS 애플리케이션을 오픈 소스 데이터베이스로 리팩터링할 수 있습니다. 공급업체의 지원을 받지 않고 COTS 애플리케이션에 오픈 소스 데이터베이스를 사용하지 않는 것이 좋습니다.

서버를 AWS 클라우드로 빠르게 리호스팅하는 데 사용할 수 있는 AWS 도구는 무엇입니까?

[MGN](#)을 기본 마이그레이션 서비스로 사용하여 애플리케이션을 리프트 앤드 시프트할 수 있습니다 AWS.

애플리케이션을 리플랫폼하는 데 사용할 수 있는 AWS 도구는 무엇입니까?

AWS 클라우드에는 다음과 같은 몇 가지 도구가 있습니다.

- [Systems Manager 자동 업그레이드](#) - AWS 클라우드에서 Windows 2008 및 SQL Server 2008을 업그레이드합니다.

- [AWS SCT](#) - 데이터베이스 스키마를 상용 데이터베이스에서 오픈 소스 데이터베이스로 변환합니다.
- [AWS DMS](#) - Amazon RDS 데이터베이스 또는 Amazon EC2에서 실행되는 데이터베이스의 온프레미스 데이터베이스에서 AWS 클라우드로 데이터를 복제합니다.

리소스

참조

- [AWS 마이그레이션 백서의 AWS Cloud Adoption Framework](#)
- [CloudWatch 대시보드 생성](#)
- [Microsoft SQL Server DB 인스턴스에 대한 제한](#)
- [자체 관리형 Microsoft SQL Server 데이터베이스를 Amazon RDS의 완전 관리형 데이터베이스로 마이그레이션](#)
- [AWS 워크숍: Amazon FSx의 Microsoft SQL Server 장애 조치 클러스터 인스턴스](#)
- [Amazon EC2 Windows Server 인스턴스 OS 업그레이드](#)

도구

- [AWS CodeBuild](#) 및는 AWS 클라우드에서 지속적 통합 및 지속적 개발(CI/CD) 파이프라인을 구축하는 데 [AWS CodePipeline](#) 도움이 됩니다.
- [AWS DataSync](#) – DataSync는 온프레미스 파일 스토어를 클라우드에 복제하는 데 도움이 됩니다 AWS .
- [AWS DMS](#) – AWS Database Migration Service (AWS DMS)는 온프레미스 데이터베이스의 데이터를 Amazon RDS 데이터베이스 또는 Amazon EC2의 데이터베이스로 복제하는 데 도움이 됩니다.
- [AWS Systems Manager 기반 자동 업그레이드](#)는 클라우드에서 Windows 2008 및 SQL Server 2008을 업그레이드하는 AWS 데 도움이 됩니다.
- [AWS SCT](#) –는 데이터베이스 스키마를 상용 데이터베이스에서 오픈 소스 데이터베이스로 변환하는 데 AWS Schema Conversion Tool 도움이 됩니다.
- [MGN](#) - 애플리케이션을 AWS Transform MGN 로 마이그레이션하는 작업을 간소화, 가속화 및 절감하는 고도로 자동화된 lift-and-shift(리호스팅) 솔루션입니다 AWS.
- [Microsoft SQL Server 데이터베이스용 Windows에서 Linux로의 리플랫폼 어시스턴트](#)는 리플랫폼 도구입니다.

문서 기록

아래 표에 이 가이드의 주요 변경 사항이 설명되어 있습니다. 향후 업데이트에 대한 알림을 받으려면 [RSS 피드](#)를 구독하십시오.

변경 사항	설명	날짜
AWS 서비스 이름 업데이트	CloudEndure 마이그레이션을 Application Migration Service로 업데이트	2022년 5월 12일
=	최초 게시	2021년 3월 12일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.