



AWS 대규모 마이그레이션을 위한 포트폴리오 플레이북

AWS 권장 가이드



AWS 권장 가이드: AWS 대규모 마이그레이션을 위한 포트폴리오 플레이북

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
대규모 마이그레이션을 위한 지침	2
실행서, 도구 및 템플릿 정보	2
1단계: 초기화	4
작업 1: 초기 검색 수행 및 마이그레이션 전략 검증	5
1단계: 검색 데이터 검증	5
2단계: 비즈니스 및 기술 동인 식별	7
3단계: 마이그레이션 전략 검증	9
4단계: 마이그레이션 패턴 검증	11
작업 종료 기준	13
작업 2: 메타데이터 식별, 수집 및 저장 프로세스 정의	14
1단계: 필수 메타데이터 정의	14
2단계: 메타데이터 스토리지 및 수집 프로세스 구축	23
3단계: 실행서에 메타데이터 요구 사항 및 수집 프로세스 문서화	29
작업 종료 기준	30
작업 3: 애플리케이션 우선 순위 지정 프로세스 정의	30
복잡성 점수 기준 정보	31
1단계: 애플리케이션 우선 순위 지정 프로세스 정의	37
2단계: 애플리케이션 우선 순위 지정 규칙 정의	42
3단계: 애플리케이션 우선 순위 지정 프로세스 완료	43
작업 종료 기준	43
작업 4: 애플리케이션 심층 분석 프로세스 정의	43
1단계: 애플리케이션 워크숍 프로세스 정의	44
2단계: 애플리케이션 매핑 프로세스 정의	48
3단계: (선택 사항) 애플리케이션 대상 상태 정의	55
4단계: 애플리케이션 심층 분석 프로세스 완료	59
작업 5: 웨이브 계획 프로세스 정의	59
1단계: 이동 그룹 프로세스 정의	60
2단계: 웨이브 계획 선택 기준 정의	63
3단계: 웨이브 계획 프로세스 완료	65
작업 종료 기준	66
2단계: 구현	67
진행 상황 추적	67
작업 1: 애플리케이션 우선 순위 지정	68

작업 2: 애플리케이션 심층 분석 수행	68
작업 3: 웨이브 계획 및 메타데이터 수집 수행	69
리소스	71
AWS 대규모 마이그레이션	71
추가 참조	71
도구 및 서비스	71
AWS 규범적 지침	71
비디오	71
기여자	72
문서 기록	73
용어집	74
#	74
A	75
B	78
C	79
D	83
E	86
F	88
G	90
H	91
I	93
L	95
M	96
O	100
P	102
Q	105
R	105
S	108
T	112
U	113
V	114
W	114
Z	115
.....	cxvii

AWS 대규모 마이그레이션을 위한 포트폴리오 플레이북

Amazon Web Services([기여자](#))

2024년 7월([문서 기록](#))

Note

애플리케이션 포트폴리오의 초기 개괄적인 검색 및 평가를 수행하는 것은 이 플레이북의 작업을 완료하기 위한 사전 조건입니다. 이 프로세스 완료에 대한 자세한 내용은 [AWS 클라우드 마이그레이션을 위한 애플리케이션 포트폴리오 평가 가이드](#)를 참조하세요.

대규모 마이그레이션에서 포트폴리오 워크스트림은 마이그레이션을 위한 애플리케이션의 파도를 계획하고 마이그레이션 워크스트림은 이러한 파도를 마이그레이션하는 데 중점을 둡니다. 웨이브를 계획할 때 포트폴리오 워크스트림은 포트폴리오를 평가하고, 마이그레이션에 필요한 메타데이터를 수집하고, 애플리케이션의 우선순위를 지정한 다음 애플리케이션을 웨이브에 할당하는 역할을 합니다. 웨이브는 마이그레이션 워크스트림의 용량에 따라 크기가 조정되고 예약되어야 하며 애플리케이션의 복잡성, 종속성 및 예산, 성능 목표, 리소스 가용성 및 기한과 같은 비즈니스 요소를 고려해야 합니다. 코어 및 지원 워크스트림에 대한 자세한 내용은 [대규모 마이그레이션을 위한 Foundation 플레이북의 대규모 마이그레이션의 Workstreams](#)를 참조하세요 AWS .

이 플레이북은 애플리케이션 평가 및 웨이브 계획을 포함하여 대규모 마이그레이션 프로젝트에 대한 자세한 포트폴리오 평가를 수행하는 step-by-step 접근 방식을 제공합니다. 대규모 마이그레이션, 초기화 및 구현의 두 단계를 모두 아우르는 포트폴리오 워크스트림의 작업을 설명합니다.

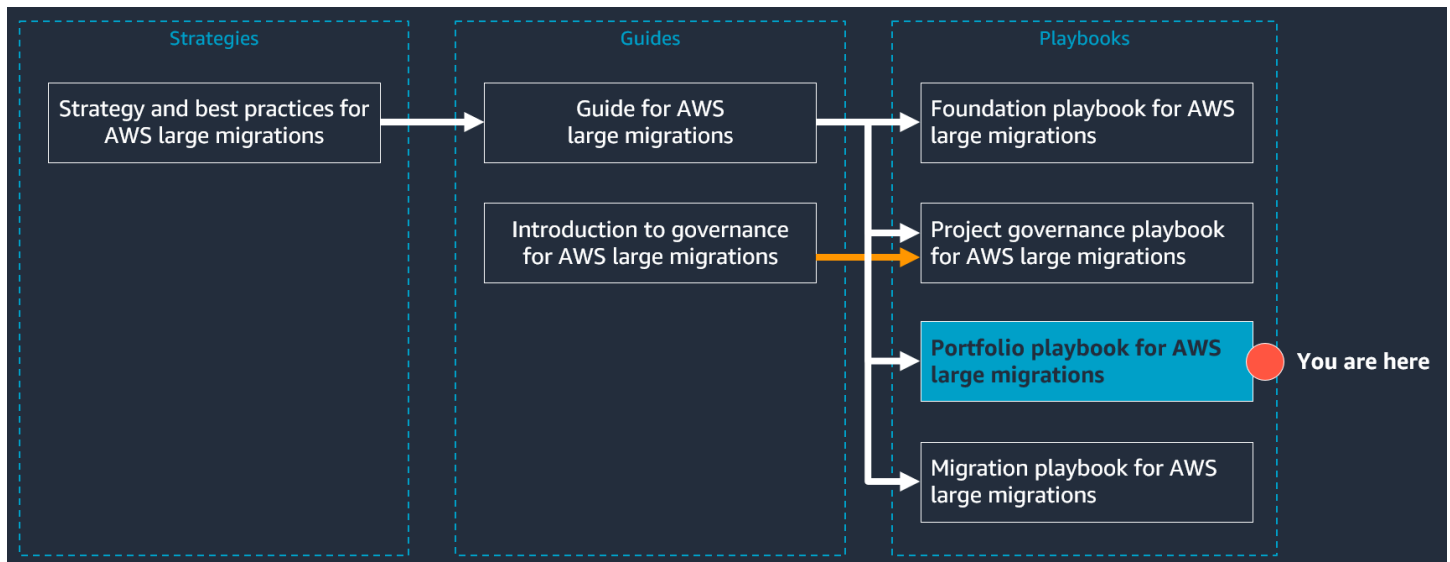
- 1단계, 초기화에서는 초기 포트폴리오 검색 및 마이그레이션 전략을 검증하고 포트폴리오 평가 및 웨이브 계획에 사용되는 프로세스와 규칙을 정의하는 런북을 생성합니다. 1단계가 끝나면 자체 포트폴리오, 프로세스 및 인프라에 맞게 사용자 지정된 포트폴리오 실행서와 추적 도구가 있습니다.
- 2단계에서는 구현하면 이전 단계에서 생성한 런북을 사용하여 포트폴리오 평가 및 웨이브 계획을 완료합니다.

상세한 포트폴리오 평가 및 웨이브 계획은 일회성 작업이 아닙니다. 마이그레이션을 지원하는 지속적인 워크스트림입니다. 마이그레이션 팩토리에서 포트폴리오 평가 및 웨이브 계획은 원재료(서버)를 팩토리에 제공하므로 마이그레이션 프로젝트가 완료될 때까지 이러한 활동을 계속해야 합니다. 마이그레이션 팩토리 모델에 대한 자세한 내용은 [AWS 대규모 마이그레이션 안내서를 참조하세요](#).

대규모 마이그레이션을 위한 지침

300대 이상의 서버를 마이그레이션하는 것은 대규모 마이그레이션으로 간주됩니다. 대규모 마이그레이션 프로젝트의 사람, 프로세스 및 기술 과제는 일반적으로 대부분의 기업에 새로운 과제입니다. 이 문서는 로의 대규모 마이그레이션에 대한 AWS 규범적 지침 시리즈의 일부입니다 AWS 클라우드. 이 시리즈는 처음부터 올바른 전략과 모범 사례를 적용하여 클라우드로의 여정을 간소화하는 데 도움이 되도록 설계되었습니다.

다음 그림은 이 시리즈의 다른 문서를 보여줍니다. 전략을 먼저 검토한 다음 가이드를 검토하고 플레이북으로 진행합니다. 전체 시리즈에 액세스하려면 [로 대규모 마이그레이션을 참조하세요 AWS 클라우](#)
[드](#).



실행서, 도구 및 템플릿 정보

이 플레이북에서는 다음 런북을 생성합니다.

- 애플리케이션 우선 순위 지정 런북
- 메타데이터 관리 런북
- 웨이브 계획 실행서

또한 진행 상황을 추적하거나 결정 및 기타 중요한 정보를 문서화하는 데 사용하는 다음 도구를 생성합니다.

- 애플리케이션 복잡성 점수 시트
- 애플리케이션 대상 상태 워크시트

- 포트폴리오 평가 진행 상황 트래커
- 애플리케이션 소유자를 위한 설문 조사
- 웨이브 계획 및 마이그레이션 대시보드

[포트폴리오 플레이북 템플릿](#)을 사용한 다음 포트폴리오, 프로세스 및 환경에 맞게 사용자 지정하는 것이 좋습니다. 이 플레이북의 지침은 이러한 각 템플릿을 사용자 지정하는 시기와 방법을 설명합니다. 이 플레이북에는 다음 템플릿이 포함되어 있습니다.

- 애플리케이션 대상 상태 워크시트 - 애플리케이션 또는 마이그레이션 전략이 특히 복잡한 경우이 템플릿을 사용하여 AWS 클라우드에서 애플리케이션의 미래 상태를 정의합니다.
- 파도 계획 및 마이그레이션을 위한 대시보드 템플릿 -이 템플릿을 사용하여 중요한 메타데이터를 수집하고, 애플리케이션 포트폴리오를 분석하고, 종속성을 식별하고, 마이그레이션 파도를 계획합니다.
- 포트폴리오 평가를 위한 진행 상황 추적 템플릿 -이 템플릿을 사용하여 포트폴리오 워크스트림을 통해 각 애플리케이션의 진행 상황을 추적할 수 있습니다.
- 애플리케이션 소유자를 위한 설문지 템플릿 - 애플리케이션 소유자로부터 직접 애플리케이션에 대한 정보를 수집하기 위해 애플리케이션 심층 분석 프로세스에서이 템플릿을 사용합니다.
- 애플리케이션 우선 순위 지정을 위한 런북 템플릿 -이 템플릿은 자체 애플리케이션 우선 순위 지정 및 심층 분석 프로세스를 구축하기 위한 출발점입니다.
- 메타데이터 관리를 위한 런북 템플릿 -이 템플릿은 자체 메타데이터 식별 및 수집 프로세스를 구축하기 위한 출발점입니다.
- 웨이브 계획을 위한 런북 템플릿 -이 템플릿은 자체 웨이브 계획 프로세스를 구축하기 위한 출발점입니다.
- 애플리케이션 복잡성에 대한 점수 시트 템플릿 -이 템플릿을 사용하여 각 애플리케이션을 클라우드로 마이그레이션하는 복잡성을 평가한 다음 애플리케이션 우선 순위 지정 프로세스 중에 결과 점수를 사용할 수 있습니다.

1단계: 대규모 마이그레이션 초기화

초기화 단계에서는 구현 단계에서 상세한 포트폴리오 평가 및 웨이브 플랜을 완료하는 데 사용하는 실행서를 정의합니다. 다른 팀원이 대규모 마이그레이션 프로젝트에서 런북을 정의할 책임이 있는 경우 [2단계: 대규모 마이그레이션 구현으로](#) 건너뛰십시오. 여기서 런북을 사용하여 애플리케이션 및 서버의 웨이브를 마이그레이션합니다. 이 단계에서 내린 결정을 문서화하면 실행 가능한 실행서를 생성할 수 있습니다. 예를 들어, 다음 질문에 대한 결정을 내리면 포트폴리오 실행서에 문서화하는 표준 절차가 마련됩니다.

- 필요한 마이그레이션 메타데이터는 무엇이며 어떻게 수집하나요?
- 애플리케이션의 우선순위를 정하고 심층 분석을 수행하려면 어떻게 해야 하나요?
- 웨이브는 어떻게 계획하나요?

1단계에서는 마이그레이션을 지원하기 위해 런북의 활동이 2단계에서 여러 번 반복되므로 규칙을 정의하고 런북을 구축하는 데 많은 시간을 할애합니다.

1단계는 다음 작업과 단계로 구성됩니다.

- [작업 1: 초기 검색 수행 및 마이그레이션 전략 검증](#)
 - [1단계: 검색 데이터 검증](#)
 - [2단계: 비즈니스 및 기술 동인 식별](#)
 - [3단계: 마이그레이션 전략 검증](#)
 - [4단계: 마이그레이션 패턴 검증](#)
- [작업 2: 메타데이터 식별, 수집 및 저장 프로세스 정의](#)
 - [1단계: 필수 메타데이터 정의](#)
 - [2단계: 메타데이터 스토리지 및 수집 프로세스 구축](#)
 - [3단계: 실행서에 메타데이터 요구 사항 및 수집 프로세스 문서화](#)
- [작업 3: 애플리케이션 우선 순위 지정 프로세스 정의](#)
 - [1단계: 애플리케이션 우선 순위 지정 프로세스 정의](#)
 - [2단계: 애플리케이션 우선 순위 지정 규칙 정의](#)
 - [3단계: 애플리케이션 우선 순위 지정 프로세스 완료](#)
- [작업 4: 애플리케이션 심층 분석 프로세스 정의](#)
 - [1단계: 애플리케이션 워크숍 프로세스 정의](#)

- [2단계: 애플리케이션 매핑 프로세스 정의](#)
- [3단계: \(선택 사항\) 애플리케이션 대상 상태 정의](#)
- [4단계: 애플리케이션 심층 분석 프로세스 완료](#)
- [작업 5: 웨이브 계획 프로세스 정의](#)
 - [1단계: 이동 그룹 프로세스 정의](#)
 - [2단계: 웨이브 계획 선택 기준 정의](#)
 - [3단계: 웨이브 계획 프로세스 완료](#)

작업 1: 초기 검색 수행 및 마이그레이션 전략 검증

대규모 마이그레이션 프로젝트에서 포트폴리오 평가의 첫 번째 단계는 현재 보유한 정보, 비즈니스 및 기술 동인, 이미 내린 마이그레이션 전략 결정을 이해하는 것입니다. 포트폴리오 평가의 결과는 마이그레이션 메타데이터, 웨이브 플랜 및 마이그레이션 전략을 마이그레이션 워크스트림에 지속적으로 제공하는 것입니다. 수집된 정보를 기반으로 격차를 분석하고 다음 단계를 결정합니다. 이미 분석 및 작업을 완료한 경우이 플레이북의 일부 섹션을 건너뛸 수 있습니다. 이 작업은 다음 단계로 구성됩니다.

- [1단계: 검색 데이터 검증](#)
- [2단계: 비즈니스 및 기술 동인 식별](#)
- [3단계: 마이그레이션 전략 검증](#)
- [4단계: 마이그레이션 패턴 검증](#)

1단계: 검색 데이터 검증

동원 단계에서는 초기 포트폴리오 평가를 완료했을 수 있으며, 완료했다면 마이그레이션 단계에서 해당 검색 데이터를 재사용할 수 있습니다. 그렇지 않은 경우 걱정하지 마세요. 이 플레이북은 대규모 마이그레이션을 지원하는 데 필요한 사항을 안내합니다.

대규모 마이그레이션에는 일반적으로 많은 데이터가 있습니다. 예를 들어 다음이 있습니다.

- 소스 서버, 애플리케이션 및 데이터베이스에 대한 메타데이터
- 구성 관리 데이터베이스(CMDB)의 IT 포트폴리오에 대한 정보
- 현재 상태 및 종속성을 더 잘 이해하는 데 도움이 되는 검색 도구의 데이터
- 대상 AWS 리소스에 대한 메타데이터

메타데이터 유형 정보

다음은 대규모 마이그레이션을 지원하는 데 필요한 세 가지 기본 메타데이터 유형입니다.

- **소스 포트폴리오 메타데이터** - 소스 포트폴리오 메타데이터는 소스 서버, 애플리케이션 및 데이터베이스에 대한 메타데이터입니다. 기존 CMDB, 검색 도구 또는 애플리케이션 소유자로부터 메타데이터를 가져올 수도 있습니다. 이 메타데이터 유형의 포괄적인 목록은 여기에서 확인할 수 있으며, 다음은 몇 가지 예입니다.
 - [서버 이름]
 - 서버 IP 주소
 - 서버 운영 체제(OS)
 - 서버 스토리지, CPU, 메모리 및 초당 입/출력 작업(IOPS)
 - 애플리케이션 이름
 - 애플리케이션 소유자
 - Application-to-application 종속성
 - 사업부
 - Application-to-server 매핑
 - Application-to-database 매핑
 - 데이터베이스 유형 및 크기
 - 스토리지 유형 및 크기
 - 종속성 메타데이터
 - 성능 및 사용량 데이터
- **대상 환경 메타데이터** - 서버를 대상 환경으로 마이그레이션하는 데 도움이 되는 메타데이터 유형입니다. 대상 환경에 대한 결정을 내려야 합니다. 검색 도구에서 메타데이터 중 일부를 가져올 수 있습니다. 다음은 이 메타데이터 유형의 몇 가지 예입니다.
 - 대상 서브넷
 - 대상 보안 그룹
 - 대상 인스턴스 유형
 - 대상 AWS Identity and Access Management (IAM) 역할
 - 대상 IP 주소
 - 대상 AWS 계정 ID
 - 대상 AWS 리전
 - 대상 AWS 서비스

- 대상 애플리케이션 아키텍처 설계
- 웨이브 계획 메타데이터 - 웨이브 계획 메타데이터는 마이그레이션을 관리하는 데 도움이 되는 메타데이터 유형입니다. 다음은 이 메타데이터 유형의 예입니다.
 - 웨이브 ID
 - 웨이브 시작 시간
 - 웨이브 전환 시간
 - 웨이브 소유자
 - Wave to application/server/database/move 그룹 매핑

검색 데이터 검증

결정을 내리기 전에 현재 검색 데이터를 이해하는 것이 중요합니다. 마이그레이션의 이 단계에서 모든 정보가 있는 것은 아닙니다. 이 플레이북은 메타데이터 요구 사항을 정의하고 메타데이터를 효율적으로 수집하는 데 도움이 됩니다. 현재 사용 가능한 메타데이터와 해당 메타데이터의 위치를 확인하려면 다음 질문을 자문해 보세요.

- 마이그레이션 평가기와 같은 마이그레이션 평가를 수행하는 데 도구를 사용한 적이 있습니까?
- Flexera One Cloud Migration and Modernization과 같은 검색 도구를 환경에 배포한 적이 있습니까?
- IT 포트폴리오에 대한 up-to-date 정보가 있는 CMDB가 있습니까?
- 동원 단계에서 초기 포트폴리오 평가를 완료했나요?
- 초기 웨이브 계획을 완료했나요?
- 초기 대상 환경 설계를 완료했습니까?
- 각 메타데이터 유형의 소스는 무엇입니까?
- 모든 메타데이터에 액세스할 수 있습니까?
- 모든 메타데이터에 액세스하려면 어떻게 해야 하나요?
- 메타데이터에 액세스하는 프로세스를 문서화했나요?

2단계: 비즈니스 및 기술 동인 식별

비즈니스 및 기술 동인은 각 애플리케이션에 대한 높은 수준의 마이그레이션 전략과 패턴을 고려할 때 매우 중요합니다. 마이그레이션에 고유한 드라이버를 이해해야 합니다. 마이그레이션 전략을 검증하고 애플리케이션 매핑 규칙을 정의할 때 이러한 비즈니스 및 기술 동인을 사용합니다.

일반적인 비즈니스 동인

비즈니스 동인은 계약 만료, 빠른 성장 또는 예산과 같이 대규모 마이그레이션을 계획할 때 고려해야 하는 비즈니스 목표 또는 제한과 관련된 요소입니다. 다음은 일반적인 비즈니스 동인입니다.

- 데이터 센터 종료 - 클라우드로 최대한 빨리 마이그레이션해야 합니다. 예를 들어 데이터 센터 계약이 곧 만료됩니다.
- 운영 비용 및 위험 감소 - 온프레미스 환경 운영과 관련된 비용 또는 위험을 줄이고자 합니다.
- 유연성 - 비즈니스의 미래 변화에 대비하려면 전략적 방향으로 클라우드로 전환해야 합니다.
- 비즈니스 성장 - 개발 및 혁신을 빠르게 가속화하거나 빠른 성장을 수용할 수 있어야 합니다.
- 데이터를 지능적으로 사용 - 회사의 성장을 예측하고 고객 행동에 대한 인사이트를 제공할 수 있는 클라우드 기반 인공 지능, 기계 학습 및 사물 인터넷(IoT)을 활용하고자 합니다.
- 보안 및 규정 준수 개선 - AWS 클라우드 인프라에 이미 내장된 규정 준수 프로그램을 활용하거나 데이터에 대한 잠재적 위협을 경고할 수 있는 소프트웨어 기반 보안 도구를 활용해야 합니다.
- 리소스 가용성 - 리소스가 제한되거나 내부 경험에 제한되면 수정 없이 애플리케이션을 이동하는 전략을 선택할 수 있습니다.

일반적인 기술 동인

기술 동인은 현재 아키텍처와 같이 대규모 마이그레이션을 계획할 때 고려해야 하는 기술적 목표 또는 제한과 관련된 요소입니다. 다음은 일반적인 기술 동인입니다.

- 하드웨어 또는 소프트웨어 end-of-support 하드웨어 또는 소프트웨어가 수명 주기 종료에 가까워지므로 공급업체에서 더 이상 지원하지 않으므로 새로 고쳐야 합니다.
- 기술 통합 - 애플리케이션을 빠르고 전략적으로 확장할 수 있는 글로벌 인프라에 액세스할 수 있습니다. 글로벌 서비스 및 인프라를 사용하여 빠르게 글로벌화할 수 있습니다.
- 스토리지 및 컴퓨팅 제한 - 데이터 센터에 더 많은 스토리지 또는 서버를 위한 용량이 없으므로 확장할 다른 위치를 찾아야 합니다.
- 확장성 및 복원력 요구 사항 - 애플리케이션이 과거에 가동 중지를 경험했고 클라우드를 사용하여 복구 시점 목표(RPO) 및 복구 시간 목표(RTO)를 개선하고자 합니다.
- 애플리케이션 아키텍처 현대화 - 클라우드를 활용하고 애플리케이션을 클라우드 네이티브로 변경하려고 합니다.
- 성능 개선 - 성수기에는 애플리케이션 성능이 좋지 않으므로 수요에 맞게 자동으로 확장 및 축소하려고 합니다.

실행서 업데이트

1. [포트폴리오 플레이북 템플릿](#)에서 애플리케이션 우선 순위 지정을 위한 런북 템플릿(Microsoft Word 형식)을 엽니다.
2. 비즈니스 및 기술 드라이버 섹션에서 대규모 마이그레이션 프로젝트에 대해 식별한 드라이버를 기록합니다.
3. 애플리케이션 우선 순위 실행서를 저장합니다.

3단계: 마이그레이션 전략 검증

대규모 마이그레이션에는 마이그레이션 전략을 선택하는 것이 중요합니다. 선택한 마이그레이션 전략이 조직의 기대치, 제한 및 요구 사항을 충족하는지 확인해야 합니다. 사용 가능한 마이그레이션 전략에 대한 자세한 내용은 [AWS 대규모 마이그레이션 안내서를 참조하세요](#).

동원 단계 또는 초기 포트폴리오 평가 중에 마이그레이션 전략을 선택했을 수 있습니다. 이 단계에서는 비즈니스 및 기술 동인을 사용하여 포트폴리오의 마이그레이션 전략을 선택하고 검증합니다.

포트폴리오를 계속 평가하고 마이그레이션을 시작하면 마이그레이션 전략이 변경될 수 있습니다. 이 단계에서 목표는 각 마이그레이션 전략에 대한 포트폴리오의 일반적인 배포를 이해하는 것입니다. 마이그레이션 전략을 선택하는 것은 자세한 마이그레이션 패턴을 검증하는 다음 단계에서 매우 중요합니다.

마이그레이션 전략 선택 및 검증

포트폴리오를 평가하고 다음과 같이 마이그레이션 전략을 선택합니다.

1. 이전 단계에서 식별한 모든 기술 및 비즈니스 동인을 검토하고 비즈니스 요구 사항에 따라 동인의 우선순위를 정합니다.
2. 각 비즈니스 및 기술 동인을 마이그레이션 전략에 매핑합니다. 다음 표는 예제입니다.

우선순위	비즈니스 또는 기술 드라이버	마이그레이션 전략
1	지정된 날짜까지 데이터 센터 종료	가능한 한 많은 애플리케이션을 리호스팅하고 리호스팅이 불가능한 경우에만 리플랫폼 및 리팩터링합니다.

우선순위	비즈니스 또는 기술 드라이버	마이그레이션 전략
2	운영 비용 및 위험 감소	마이그레이션을 가속화하려면 최대한 많은 애플리케이션을 리호스팅합니다.
3	하드웨어 또는 소프트웨어 end-of-support	지원되는 애플리케이션을 리호스팅하고 클라우드의 최신 하드웨어 및 소프트웨어에서 지원되지 않는 애플리케이션을 리플랫폼합니다.
4	리소스 가용성	AWS Managed Services (AMS)로 리호스팅하여 운영 오버헤드를 줄입니다.

3. 각 비즈니스 및 기술 동인을 평가하고 포트폴리오를 개괄적으로 평가하여 각 마이그레이션 전략 간에 애플리케이션을 배포하는 방법을 추정합니다. 드라이버 간의 충돌을 확인하는 것이 일반적입니다. 프로젝트 이해관계자는 협력하여 충돌을 해결하기 위한 최종 결정을 내려야 합니다. 다음은 포트폴리오를 각 마이그레이션 전략에 배포하는 방법의 예입니다.

- 리호스팅 - 60%
- 리플랫폼 - 15%
- 사용 중지 - 10%
- 유지 - 5%
- 재구매 - 5%
- 리팩터링 - 5%

포트폴리오에 대한 상위 수준 마이그레이션 전략을 선택할 때까지 마이그레이션을 진행하지 마십시오.

실행서 업데이트

1. 애플리케이션 우선 순위 실행서를 엽니다.
2. 마이그레이션 전략 섹션에서 애플리케이션 워크로드가 7가지 마이그레이션 전략 간에 어떻게 분산되는지 기록합니다. 예제:
 - 리호스팅 - 60%

- 리플랫폼 - 15%
- 사용 중지 - 10%
- 유지 - 5%
- 재구매 - 5%
- 리팩터링 - 5%

3. 애플리케이션 우선 순위 실행서를 저장합니다.

4단계: 마이그레이션 패턴 검증

마이그레이션 패턴 정보

마이그레이션 패턴은 마이그레이션 전략, 마이그레이션 대상, 사용된 마이그레이션 애플리케이션 또는 서비스를 자세히 설명하는 반복 가능한 마이그레이션 작업입니다. 예를 들어 를 사용하여 Amazon Elastic Compute Cloud(Amazon EC2)로 리호스팅 AWS Transform MGN합니다. 다음 AWS 서비스 및 솔루션은 일반적인 마이그레이션 패턴에서 자주 참조됩니다.

- AWS App2Container
- AWS Transform MGN (AWS MGN)
- AWS CloudFormation
- AWS Database Migration Service (AWS DMS)
- AWS DataSync
- Amazon Elastic Compute Cloud(Amazon EC2)
- Amazon Elastic Container Service(Amazon ECS)
- Amazon Elastic File System(Amazon EFS)
- AWS 클라우드 마이그레이션 팩토리 솔루션
- Amazon Relational Database Service(Amazon RDS)
- AWS Schema Conversion Tool (AWS SCT)
- AWS Transfer Family

마이그레이션 전략을 선택하는 것과 마찬가지로 이전 단계에서 마이그레이션 패턴을 이미 식별했을 수 있습니다. 그러나 이를 검증하고 패턴이 정의되고 문서화되었는지 확인해야 합니다. 다음 표에는 일반적인 마이그레이션 전략과 패턴이 나와 있습니다.

ID	전략	패턴
1	리호스팅	MGN 또는 Cloud Migration Factory를 사용하여 Amazon EC2로 리호스팅
2	리플랫폼	AWS DMS 및를 사용하여 Amazon RDS로 리플랫폼 AWS SCT
3	리플랫폼	를 사용하여 Amazon EC2로 리플랫폼 CloudFormation  Note CloudFormation 템플릿은에서 새 인프라를 빌드합니다 AWS 클라우드.
4	리플랫폼	AWS DataSync 또는를 사용하여 Amazon EFS로 리플랫폼 AWS Transfer Family
5	리플랫폼	AWS App2Container를 사용하여 Amazon ECS로 리플랫폼
6	리플랫폼	에뮬레이터를 사용하여 메인프레임 또는 미드레인지 서버를 Amazon EC2로 리플랫폼
7	리플랫폼	Amazon EC2에서 Windows에서 Linux로 리플랫폼
8	만료	애플리케이션 사용 중지
9	보관	온프레미스에 보관

ID	전략	패턴
10	재구매	SaaS로 재구매 및 업그레이드
11	리팩터링 또는 리아키텍트	애플리케이션 재설계

실행서 업데이트

이때 포트폴리오 수준에서 패턴을 정의합니다. 이 플레이북의 뒷부분에서 각 애플리케이션을 해당 마이그레이션 패턴에 매핑합니다.

1. 애플리케이션 우선 순위 실행서를 엽니다.
2. 마이그레이션 패턴 섹션에서 식별하고 검증한 마이그레이션 패턴을 기록합니다. 각 패턴에 고유한 ID를 할당하고 패턴에 대한 마이그레이션 전략을 기록해 둡니다.
3. 애플리케이션 우선 순위 실행서를 저장합니다.

마이그레이션 패턴이 진행됨에 따라 변경될 수 있습니다. 나중에 새 정보를 찾거나 워크로드 범위를 변경하거나 새 AWS 서비스를 사용하기로 결정할 때 마이그레이션 전략과 패턴을 변경할 수 있습니다.

작업 종료 기준

높은 수준의 포트폴리오 관점에서 마이그레이션 전략과 패턴을 아직 식별하지 못한 경우 다음 작업으로 넘어가기 전에 기술 팀과 협력하여 이를 정의하는 것이 좋습니다. 포트폴리오 평가 및 웨이브 계획은 마이그레이션 전략과 패턴을 이해하는 데 달려 있습니다. 계속하기 전에 마이그레이션 패턴의 포괄적인 목록이 없어도 됩니다. 새 패턴을 추가하고 진행에 따라 전략을 조정할 수 있습니다.

다음을 완료하면 다음 작업을 계속합니다.

- 최신 검색 데이터에 액세스하여 이해할 수 있습니다.
- 마이그레이션의 비즈니스 및 기술 동인을 식별했습니다.
- 비즈니스 및 기술 동인에 따라 마이그레이션 전략을 선택하고 검증했습니다.
- 마이그레이션 패턴을 선택하고 검증했습니다.
- 애플리케이션 우선 순위 실행서에 다음을 문서화했습니다.
 - 비즈니스 및 기술 동인
 - 마이그레이션 전략

- 마이그레이션 패턴

작업 2: 메타데이터 식별, 수집 및 저장 프로세스 정의

이전 작업에서는 초기 검색 데이터, 마이그레이션 전략 및 대규모 마이그레이션을 위한 마이그레이션 패턴을 검증했습니다. 이 작업에서는 필요한 메타데이터를 식별하고 이를 수집하는 방법을 결정합니다. 이 작업은 다음 단계로 구성됩니다.

- [1단계: 필수 메타데이터 정의](#)
- [2단계: 메타데이터 스토리지 및 수집 프로세스 구축](#)
- [3단계: 실행서에 메타데이터 요구 사항 및 수집 프로세스 문서화](#)

이 섹션의 단계를 완료하면 메타데이터 관점에서 전체 마이그레이션 주기를 고려하세요. 포트폴리오 평가, 웨이브 계획, 마이그레이션, 테스트, 전환 후 활동을 고려한 다음 가능한 모든 사용 사례 및 관련 사용 사례를 분석합니다. 전체 마이그레이션 프로세스를 완료하는 데 필요한 정보를 생각하면 해당 패턴의 모든 메타데이터를 식별하는 데 도움이 됩니다.

1단계: 필수 메타데이터 정의

필요한 메타데이터 속성을 확인하려면 먼저 마이그레이션 패턴을 이해해야 합니다. 예를 들어 서버를 Amazon EC2로 마이그레이션하고 데이터베이스를 Amazon RDS로 마이그레이션하려면 서로 다른 메타데이터가 필요합니다. 대부분의 패턴은 많은 작은 작업으로 구성됩니다. 마이그레이션 패턴을 수행하려면 필요한 메타데이터 속성을 파악한 다음 해당 애플리케이션의 메타데이터를 수집해야 합니다. 구현 단계에서 지연 없이 효율적으로 마이그레이션을 수행할 수 있도록 초기화 단계에서 필요한 메타데이터를 결정하고 수집해야 합니다.

메타데이터 속성을 정의하는 사람 또는 팀은 마이그레이션 패턴을 수행하는 데 필요한 단계와 작업을 정의하는 것으로 시작합니다. 태스크는 필요한 메타데이터를 결정하므로 각 태스크를 통해 필요한 메타데이터의 포괄적인 컬렉션을 빌드합니다. 필요한 메타데이터를 결정하는 사람은 일반적으로 마이그레이션 패턴을 완료하는 방법을 포괄적으로 이해해야 합니다. 마이그레이션 실행서를 작성하는 사람과의 조정이 필요할 수 있습니다. 자세한 내용은 [AWS 대규모 마이그레이션을 위한 마이그레이션 플레이북을 참조하세요](#).

대규모 마이그레이션 중에는 메타데이터에 의존하는 여러 프로세스가 모든 워크스트림에 분산되어 있습니다. 적시에 정확한 메타데이터를 확보하면 대규모 마이그레이션의 성공에 광범위하고 상당한 영향을 미칩니다.

이 단계에서는 패턴 또는 작업을 정의한 다음 정의를 사용하여 필요한 메타데이터를 식별합니다.

마이그레이션 패턴 및 지원 작업의 주요 구성 요소 식별

이 단계에서는 각 마이그레이션 패턴 또는 지원 작업에 대해 작업, 소스 객체, 대상 객체 및 사용되는 도구와 같은 주요 구성 요소를 정의합니다. 그런 다음 답변을 기반으로 패턴 또는 작업의 이름을 지정합니다.

지원되는 작업에는 웨이브 계획, 애플리케이션 우선순위 지정, 종속성 분석, 거버넌스, 재해 복구, 성능 테스트 또는 사용자 수락 테스트와 같이 마이그레이션 중에 포트폴리오 및 마이그레이션 워크스트림이 수행해야 하는 운영 활동이 포함됩니다. 이러한 작업을 지원하려면 메타데이터가 필요하므로 마이그레이션 패턴과 지원 작업 모두에 대해 이 단계를 수행합니다.

1. 작업 - 마이그레이션 전략 또는 지원 작업을 식별합니다. 한 작업에는 연결된 다른 작업이 있을 수 있습니다. 예를 들어 마이그레이션 작업을 정의할 수 있습니다. 예제 작업은 다음과 같습니다.

- 리호스팅, 리플랫폼 또는 재배포와 같은 마이그레이션 전략
- 웨이브 계획
- 애플리케이션 우선 순위 및 종속성 분석
- 연산
- 거버넌스
- 재해 복구
- 성능 테스트 또는 사용자 수락 테스트(UAT)와 같은 테스트

2. 소스 객체 - 작업이 수행될 소스 객체를 식별합니다. 소스 객체의 예는 다음과 같습니다.

- 웨이브
- Server
- Database
- 파일 공유
- 애플리케이션

3. 도구 - 작업을 수행하는 데 사용되는 서비스 또는 도구를 식별합니다. 둘 이상의 도구 또는 서비스를 사용할 수 있습니다. 예제 도구는 다음과 같습니다.

- AWS Transform MGN
- AWS DataSync
- AWS Database Migration Service (AWS DMS)

- AWS Backup

- 성능 모니터링 도구
4. 대상 객체 - 작업이 완료될 때 소스가 상주할 대상 객체, 서비스 또는 위치를 식별합니다. 객체, 서비스 또는 위치의 예는 다음과 같습니다.
- Amazon Elastic Compute Cloud(Amazon EC2)
 - Amazon Relational Database Service(Amazon RDS)
 - Amazon Elastic File System(Amazon EFS)
 - Amazon Elastic Container Service(Amazon ECS)
 - 웨이브 풀랜
5. 패턴 이름 - 다음과 같이 이전 단계에 대한 답변을 결합합니다.

<tool>을 사용하여 <action> <source object> on/to <target object>

예를 들어, 다음과 같습니다.

- MGN 또는 Cloud Migration Factory(도구)를 사용하여 Amazon EC2(대상 객체)로 (작업) 웨이브, 애플리케이션 또는 서버(소스 객체) 리호스팅
- DataSync(도구)를 사용하여 (작업) 파일 공유(소스 객체)를 Amazon EFS(대상 객체)로 리플랫폼
- (도구)를 사용하여 AWS DMS (작업) 데이터베이스(소스 객체)를 Amazon RDS(대상 객체)로 리플랫폼
- Amazon CloudWatch(도구)를 사용하여 Amazon EC2(대상 객체)에서 애플리케이션(소스 객체)의 성능 모니터링(작업) Amazon CloudWatch
- 마이그레이션 후 (도구)를 사용하여 AWS Backup Amazon EC2(대상 객체)에 (작업) 서버(소스 객체) 백업
- 웨이브 풀랜(대상 객체)을 생성하기 위한 웨이브 계획(작업) 웨이브, 애플리케이션 또는 서버(소스 객체)

다음은 마이그레이션 패턴 테이블에서 MGN 또는 Cloud Migration Factory를 사용하여 패턴 1: Amazon EC2로 리호스팅을 기록하는 방법의 예입니다. [???<\/a>](#)

Pattern ID	1
Pattern name	MGN 또는 Cloud Migration Factory를 사용하여 Amazon EC2로 리호스팅

Action	리호스팅 마이그레이션
Source object	웨이브, 애플리케이션 또는 서버
Tools	MGN 또는 클라우드 마이그레이션 팩토리
Target object	Amazon EC2

각 패턴 또는 작업에 필요한 메타데이터 확인

이제 패턴 또는 작업을 정의했으므로 소스 객체, 대상 객체, 도구 및 기타 비즈니스 정보에 필요한 메타데이터를 결정합니다. 이 프로세스를 설명하기 위해이 플레이북은 [??? 마이그레이션 패턴 테이블](#)의 MGN 또는 Cloud Migration Factory를 사용하여 패턴 1: Amazon EC2로 리호스팅을 예로 사용합니다. 일부 패턴 또는 작업의 경우 일부 단계가 적용되지 않을 수 있습니다.

1. 대상 객체 분석 - 대상 객체에서 역방향으로 작업하여 객체를 수동으로 생성하고 이를 지원하는 데 필요한 메타데이터를 식별합니다. 다음 표와 같이 메타데이터를 캡처합니다.

예를 들어 EC2 인스턴스를 생성할 때 인스턴스 유형, 스토리지 유형, 스토리지 크기, 서브넷, 보안 그룹 및 태그를 선택해야 합니다. 다음 표에는 대상 객체가 EC2 인스턴스인 경우 필요할 수 있는 메타데이터 속성의 예가 나와 있습니다.

속성 이름	객체 유형	설명 또는 용도
target_subnet	대상 EC2 인스턴스	대상 EC2 인스턴스의 서브넷
target_subnet_test	대상 EC2 인스턴스	대상 EC2 인스턴스의 서브넷 테스트
target_security_group	대상 EC2 인스턴스	대상 EC2 인스턴스의 보안 그룹
target_security_group_test	대상 EC2 인스턴스	대상 EC2 인스턴스의 보안 그룹 테스트
IAM_role	대상 EC2 인스턴스	AWS Identity and Access Management 대상 EC2 인스턴스의 (IAM) 역할

속성 이름	객체 유형	설명 또는 용도
instance_type	대상 EC2 인스턴스	대상 EC2 인스턴스의 인스턴스 유형
AWS_account_ID	대상 EC2 인스턴스	AWS 대상 EC2 인스턴스를 호스팅하기 위한 계정
AWS_Region	대상 EC2 인스턴스	AWS 대상 EC2 인스턴스를 호스팅할 리전

2. 도구 분석 - 도구를 사용하여 대상 객체를 생성하고 차이점을 확인합니다. 다음 표에 표시된 대로 도구별 메타데이터를 캡처하고 마이그레이션 도구에서 지원되지 않는 경우 이전 테이블에서 속성을 제거합니다. 예를 들어 리호스팅 마이그레이션 도구가 like-for-like하기 때문에 MGN의 OS 유형 및 스토리지 크기를 사용자 지정할 수 없습니다. 따라서 이러한 속성이 이전 테이블에 포함된 경우 대상 OS 및 대상 디스크 크기를 제거합니다. 이전 예제 테이블에서는 도구에서 모든 속성을 지원하므로 별도의 작업이 필요하지 않습니다.

다음 표에는 도구에 필요할 수 있는 메타데이터의 예가 나와 있습니다.

속성 이름	객체 유형	설명 또는 용도
AWS_account_ID	도구(MGN)	AWS의 계정 ID AWS Transform MGN
AWS_Region	도구(MGN)	AWS MGN의 리전
replication_server_subnet	도구(MGN)	MGN 복제 서버용 서브넷
replication_server_security_group	도구(MGN)	MGN 복제 서버의 보안 그룹

3. 소스 객체 분석 - 다음과 같이 작업을 평가하여 소스 객체에 필요한 메타데이터를 결정합니다.

- 서버를 마이그레이션하려면 서버에 연결하기 위해 소스 서버 이름과 정규화된 도메인 이름(FQDN)을 알아야 합니다.
- 서버와 함께 애플리케이션을 마이그레이션하려면 애플리케이션 이름, 애플리케이션 환경 및 application-to-server 알아야 합니다.

- 포트폴리오 평가를 수행하거나, 애플리케이션의 우선순위를 지정하거나, 이동 그룹을 정의하려면 application-to-server 매핑, 애플리케이션-application-to-database 매핑, application-to-application 종속성을 알아야 합니다.
- 웨이브를 관리하려면 웨이브 ID와 웨이브의 시작 및 종료 시간을 알아야 합니다.

다음 표에는 소스 객체에 필요할 수 있는 메타데이터의 예가 나와 있습니다.

속성 이름	객체 유형	설명 또는 용도
wave_ID	소스 웨이브	웨이브의 ID(예: 웨이브 10)
wave_start_date	소스 웨이브	웨이브 시작 날짜
wave_cutover_date	소스 웨이브	웨이브의 전환 날짜
wave_owner	소스 웨이브	파도 소유자
app_name	소스 애플리케이션	소스 애플리케이션 이름
app_to_server_mapping	소스 애플리케이션	Application-to-server 관계
app_to_DB_mapping	소스 애플리케이션	Application-to-database 관계
app_to_app_dependencies	소스 애플리케이션	애플리케이션의 외부 종속성
server_name	소스 서버	소스 서버 이름
server_FQDN	소스 서버	소스 서버의 정규화된 도메인 이름
server_OS_family	소스 서버	소스 서버의 운영 체제(OS) 패밀리(예: Windows 또는 Linux)
server_OS_version	소스 서버	소스 서버의 OS 버전(예: Windows Server 2003)
server_environment	소스 서버	소스 서버의 환경(예: 개발, 프로덕션 또는 테스트)

속성 이름	객체 유형	설명 또는 용도
server_tier	소스 서버	소스 서버의 티어(예: 웹, 데이터베이스 또는 애플리케이션)
CPU	소스 서버	소스 서버의 CPUs 수
RAM	소스 서버	소스 서버의 RAM 크기
disk_size	소스 서버	소스 서버의 디스크 크기

4. 다른 속성 고려 - 기본 작업 외에도 대상 객체 또는 애플리케이션과 관련된 다른 작업 및 속성을 고려합니다. 예제 패턴의 경우 패턴 1: MGN 또는 Cloud Migration Factory를 사용하여 Amazon EC2로 리호스팅하면 작업이 리호스팅되고 대상 객체는 Amazon EC2입니다. 이 대상 객체에 대한 다른 관련 작업에는 Amazon EC2에 백업, 마이그레이션 후 EC2 인스턴스 모니터링, 태그를 사용하여 EC2 인스턴스와 관련된 비용을 관리하는 작업이 포함될 수 있습니다. 질문이나 전환 목적으로 문의해야 하는 애플리케이션 소유자와 같이 마이그레이션을 관리하는 데 도움이 되는 다른 애플리케이션 속성을 고려할 수도 있습니다.

다음 표에는 일반적으로 사용되는 추가 메타데이터의 예가 나와 있습니다. 이 테이블에는 대상 EC2 인스턴스에 대한 태그가 포함되어 있습니다. 태그 및 태그 사용 방법에 대한 자세한 내용은 [Amazon EC2 설명서의 Amazon EC2 리소스 태그 지정을 참조하세요](#). Amazon EC2

속성 이름	객체 유형	설명 또는 용도
Name	대상 EC2 인스턴스(태그)	대상 EC2 인스턴스의 이름을 정의하는 태그
app_owner	소스 애플리케이션	소스 애플리케이션의 소유자
business_unit	대상 EC2 인스턴스(태그)	대상 EC2 인스턴스의 사업부를 식별하는 태그(예: HR, 재무 또는 IT)
cost_center	대상 EC2 인스턴스(태그)	대상 EC2 인스턴스의 비용 센터를 식별하는 태그

5. 테이블 생성 - 이전 단계에서 식별된 모든 메타데이터를 단일 테이블로 결합합니다.

속성 이름	객체 유형	설명 또는 용도
wave_ID	소스 웨이브	웨이브의 ID(예: 웨이브 10)
wave_start_date	소스 웨이브	웨이브 시작 날짜
wave_cutover_date	소스 웨이브	웨이브의 전환 날짜
wave_owner	소스 웨이브	파도 소유자
app_name	소스 애플리케이션	소스 애플리케이션 이름
app_to_server_mapping	소스 애플리케이션	Application-to-server 관계
app_to_DB_mapping	소스 애플리케이션	Application-to-database 관계
app_to_app_dependencies	소스 애플리케이션	애플리케이션의 외부 종속성
AWS_account_ID	도구(MGN)	AWS 대상 EC2 인스턴스를 호스팅하기 위한 계정
AWS_Region	도구(MGN)	AWS 대상 EC2 인스턴스를 호스팅할 리전
replication_server_subnet	도구(MGN)	MGN 복제 서버용 서브넷
replication_server_security_group	도구(MGN)	MGN 복제 서버의 보안 그룹
server_name	소스 서버	소스 서버 이름
server_FQDN	소스 서버	소스 서버의 정규화된 도메인 이름
server_OS_family	소스 서버	소스 서버의 운영 체제(OS) 패밀리(예: Windows 또는 Linux)

속성 이름	객체 유형	설명 또는 용도
server_OS_version	소스 서버	소스 서버의 OS 버전(예: Windows Server 2003)
server_environment	소스 서버	소스 서버의 환경(예: 개발, 프로덕션 또는 테스트)
server_tier	소스 서버	소스 서버의 티어(예: 웹, 데이터베이스 또는 애플리케이션)
CPU	소스 서버	소스 서버의 CPUs 수
RAM	소스 서버	소스 서버의 RAM 크기
disk_size	소스 서버	소스 서버의 디스크 크기
target_subnet	대상 서버	대상 EC2 인스턴스의 서브넷
target_subnet_test	대상 서버	대상 EC2 인스턴스의 서브넷 테스트
target_security_group	대상 서버	대상 EC2 인스턴스의 보안 그룹
target_security_group_test	대상 서버	대상 EC2 인스턴스의 보안 그룹 테스트
instance_type	대상 서버	대상 EC2 인스턴스의 인스턴스 유형
IAM_role	대상 서버	AWS Identity and Access Management 대상 EC2 인스턴스의 (IAM) 역할
Name	대상 서버(태그)	대상 EC2 인스턴스의 이름을 정의하는 태그
app_owner	소스 애플리케이션	소스 애플리케이션의 소유자

속성 이름	객체 유형	설명 또는 용도
business_unit	대상 서버(태그)	대상 EC2 인스턴스의 사업부를 식별하는 태그(예: HR, 재무 또는 IT)
cost_center	대상 서버(태그)	대상 EC2 인스턴스의 비용 센터를 식별하는 태그

6. 반복 - 각 패턴에 필요한 메타데이터를 문서화할 때까지 이 프로세스를 반복합니다.

2단계: 메타데이터 스토리지 및 수집 프로세스 구축

이전 단계에서 마이그레이션을 지원하는 데 필요한 메타데이터를 정의했습니다. 이 단계에서는 메타데이터를 수집하고 저장하는 프로세스를 구축합니다. 이 단계는 다음 두 가지 작업으로 구성됩니다.

1. 이전 단계의 필수 메타데이터를 분석하고 소스를 식별합니다.
2. 메타데이터를 효율적으로 저장하고 수집하는 프로세스를 정의합니다.

메타데이터 소스 분석

많은 일반적인 메타데이터 소스가 있습니다. 일반적으로 가장 먼저 액세스할 수 있는 것은 상위 수준 자산 인벤토리로, 일반적으로 구성 관리 데이터베이스(CMDB) 또는 다른 기존 도구에서 내보냅니다. 그러나 자동화된 프로세스와 수동 프로세스를 모두 사용하여 다른 소스에서도 메타데이터를 수집해야 합니다.

다음 표에는 공통 소스, 해당 소스의 표준 수집 프로세스, 해당 소스에서 찾을 것으로 예상되는 공통 메타데이터 유형이 나와 있습니다.

메타데이터 소스	컬렉션 유형	메타데이터 유형
검색 도구	자동	소스 서버
CMDB	자동	소스 서버
용 RVTools 와 같은 다른 도구의 인벤토리 VMware vSphere	자동	소스 서버

메타데이터 소스	컬렉션 유형	메타데이터 유형
애플리케이션 소유자 설문	수동	소스 서버, 대상 서버, 웨이브
애플리케이션 소유자 인터뷰	수동	소스 서버, 대상 서버, 웨이브
애플리케이션 설계 설명서	수동	대상 서버
랜딩 존 설계 설명서	수동	대상 서버, 도구

메타데이터의 가능한 모든 소스를 나열한 후 메타데이터 유형을 분석하고 각 소스를 이전 단계에서 식별한 메타데이터 속성에 매핑합니다.

- 에서 메타데이터 속성의 전체 목록을 가져옵니다. [1단계: 필수 메타데이터 정의](#).
- 각 메타데이터 유형을 분석하고 자동화된 프로세스를 사용하여 검색할 수 없는 유형을 결정합니다. 이는 일반적으로 대상 서버 메타데이터 및 웨이브 메타데이터 유형입니다. 애플리케이션 소유자의 결정이 필요하기 때문입니다. 예를 들어 대상 EC2 인스턴스에 사용할 서브넷 및 보안 그룹은 무엇입니까?
- 각 메타데이터 속성을 분석하여 이전 테이블의 메타데이터 소스에 매핑합니다. 여러 소스를 조합하여 사용하는 것이 일반적입니다. 검색 도구를 사용하여 일부 소스 서버 메타데이터를 수집할 수 있습니다. 검색 도구를 사용하여 메타데이터를 수집하는 방법에 대한 자세한 내용은 AWS 권장 가이드 웹 사이트의 [자동 포트폴리오 검색 시작하기](#)를 참조하세요.
- 테이블을 생성하여 메타데이터 속성을 유형 및 소스에 매핑합니다. 다음 표는 예제입니다.

메타데이터 속성	메타데이터 유형	메타데이터 소스
app_name	소스 애플리케이션	CMDB
app_owner	소스 애플리케이션	CMDB
app_to_server_mapping	소스 애플리케이션	CMDB, 검색 도구 또는 애플리케이션 소유자 설문지
app_to_DB_mapping	소스 애플리케이션	CMDB, 검색 도구 또는 애플리케이션 소유자 설문지

메타데이터 속성	메타데이터 유형	메타데이터 소스
app_to_app_dependencies	소스 애플리케이션	CMDB, 검색 도구 또는 애플리케이션 소유자 설문지
server_name	소스 서버	CMDB
server_FQDN	소스 서버	CMDB
server_OS_family	소스 서버	CMDB
server_IP	소스 서버	검색 도구
disk_size	소스 서버	검색 도구
instance_type	대상 서버	검색 도구
target_subnet	대상 서버	애플리케이션 소유자 설문
target_security_group	대상 서버	애플리케이션 소유자 설문
AWS_Region	대상 서버	애플리케이션 소유자 설문
AWS_account_ID	대상 서버	애플리케이션 소유자 설문
replication_server_subnet	도구(MGN)	랜딩 존 설계 설명서
replication_server_security_group	도구(MGN)	랜딩 존 설계 설명서
Name	대상 서버(태그)	애플리케이션 소유자 설문
business_unit	대상 서버(태그)	애플리케이션 소유자 설문
cost_center	대상 서버(태그)	애플리케이션 소유자 설문
wave_ID	웨이브 계획	애플리케이션 소유자 인터뷰
wave_start_date	웨이브 계획	애플리케이션 소유자 인터뷰

메타데이터 속성	메타데이터 유형	메타데이터 소스
wave_cutover_date	웨이브 계획	애플리케이션 소유자 인터뷰

단일 메타데이터 스토어 정의

각 메타데이터 속성을 소스에 매핑한 후 메타데이터를 저장할 위치를 정의합니다. 메타데이터를 저장하는 방법과 위치에 관계없이 리포지토리를 하나만 선택해야 합니다. 이렇게 하면 단일 정보 소스를 확보할 수 있습니다. 메타데이터를 여러 곳에 저장하는 것은 대규모 마이그레이션에서 흔히 발생하는 실수입니다.

옵션 1: 공유 리포지토리의 스프레드시트에 메타데이터 저장

이 옵션은 매우 수동적인 프로세스처럼 들릴 수 있지만 대규모 마이그레이션을 위한 가장 일반적인 데이터 스토어입니다. 또한 Microsoft SharePoint 사이트와 같은 공유 리포지토리에 스프레드시트를 저장하는 것이 일반적입니다.

Microsoft Excel 스프레드시트는 사용자 지정하기 쉽고 빌드하는 데 시간이 오래 걸리지 않습니다. 단점은 메타데이터가 많으면 매우 복잡해지고 서버, 애플리케이션 및 데이터베이스와 같은 자산 간의 관계를 관리하기 어려울 수 있다는 것입니다. 또 다른 문제는 버전 관리입니다. 쓰기 액세스를 소수의 사용자만으로 제한하거나 자동화된 프로세스를 사용하여 스프레드시트를 업데이트해야 합니다.

[포트폴리오 플레이북 템플릿](#)에서 웨이브 계획 및 마이그레이션(Excel 형식)을 위한 대시보드 템플릿을 자체 데이터 스토어 스프레드시트를 빌드하기 위한 출발점으로 사용할 수 있습니다.

옵션 2: 특별히 구축된 도구에 메타데이터 저장

[TDS Transition Manager](#)(TDS 웹 사이트)와 같은 사전 구축된 도구를 사용하여 데이터를 저장하거나 자체 도구를 구축할 수 있습니다. 자체 도구를 빌드할 때는 옵션 1의 Excel 스프레드시트 탭과 마찬가지로 데이터베이스 테이블이 필요합니다. 예제:

- 서버 테이블
- 애플리케이션 테이블
- 데이터베이스 테이블
- Application-to-server 및 application-to-database 매핑 테이블
- 웨이브 플랜 테이블
- 애플리케이션 소유자 설문지 테이블

메타데이터 수집 프로세스 정의

이전 단계에서는 메타데이터를 소스에 매핑하고 메타데이터를 수집할 데이터 스토어를 정의했습니다. 이 단계에서는 메타데이터를 효과적으로 수집하는 프로세스를 구축합니다. 수동 copy-and-paste 프로세스를 최소화하고 자동화를 사용하여 각 소스에서 메타데이터를 수집해야 합니다. 다음 세 단계가 있습니다.

1. 메타데이터 매핑 테이블을 기반으로 각 메타데이터 소스에 대한 추출, 변환 및 로드(ETL) 스크립트를 빌드합니다.
2. 각 소스에서 메타데이터를 정기적으로 자동으로 가져오는 예약된 작업을 빌드합니다.
3. 내보내기 프로세스를 구축하거나 리포지토리에 저장된 메타데이터에 대한 애플리케이션 프로그래밍 인터페이스(API) 액세스를 제공합니다.

다음 표는 각 ETL 스크립트에서 수집한 메타데이터 속성의 예입니다. 메타데이터는 스프레드시트 또는 특별히 제작된 도구와 같이 이전 섹션에서 정의한 위치에 저장됩니다.

메타데이터 속성	메타데이터 유형	메타데이터 소스	수집 프로세스
app_name	소스 애플리케이션	CMDB	ETL 스크립트 – CMDB
app_owner	소스 애플리케이션	CMDB	ETL 스크립트 – CMDB
app_to_server_mapping	소스 애플리케이션	CMDB	ETL 스크립트 – CMDB
app_to_DB_mapping	소스 애플리케이션	CMDB	ETL 스크립트 – CMDB
app_to_app_dependencies	소스 애플리케이션	검색 도구	ETL 스크립트 - 검색 도구
server_name	소스 서버	CMDB	ETL 스크립트 – CMDB
server_FQDN	소스 서버	CMDB	ETL 스크립트 – CMDB

메타데이터 속성	메타데이터 유형	메타데이터 소스	수집 프로세스
server_OS_family	소스 서버	CMDB	ETL 스크립트 - CMDB
server_OS_version	소스 서버	CMDB	ETL 스크립트 - CMDB
disk_size	소스 서버	검색 도구	ETL 스크립트 - 검색 도구
instance_type	대상 서버	검색 도구	ETL 스크립트 - 검색 도구
target_subnet	대상 서버	애플리케이션 소유자 설문	ETL 스크립트 - 애플리케이션 소유자
target_security_group	대상 서버	애플리케이션 소유자 설문	ETL 스크립트 - 애플리케이션 소유자
AWS_Region	대상 서버	애플리케이션 소유자 설문	ETL 스크립트 - 애플리케이션 소유자
AWS_account_ID	대상 서버	애플리케이션 소유자 설문	ETL 스크립트 - 애플리케이션 소유자
Name	대상 서버(태그)	애플리케이션 소유자 설문	ETL 스크립트 - 애플리케이션 소유자
business_unit	대상 서버(태그)	애플리케이션 소유자 설문	ETL 스크립트 - 애플리케이션 소유자
cost_center	대상 서버(태그)	애플리케이션 소유자 설문	ETL 스크립트 - 애플리케이션 소유자
wave_ID	웨이브 계획	애플리케이션 소유자 설문	ETL 스크립트 - 애플리케이션 소유자
wave_start_date	웨이브 계획	애플리케이션 소유자 설문	ETL 스크립트 - 애플리케이션 소유자

메타데이터 속성	메타데이터 유형	메타데이터 소스	수집 프로세스
wave_cuto ver_date	웨이브 계획	애플리케이션 소유자 설문	ETL 스크립트 - 애플 리케이션 소유자

3단계: 실행서에 메타데이터 요구 사항 및 수집 프로세스 문서화

이 작업에서는 메타데이터 관리 실행서에 결정을 문서화합니다. 마이그레이션 중에 포트폴리오 워크스트림은 메타데이터 수집 및 저장을 위한 표준 절차로 이 런북을 준수합니다.

1. [포트폴리오 플레이북 템플릿](#)에서 메타데이터 관리를 위한 런북 템플릿(Microsoft Word 형식)을 엽니다. 이는 자체 런북을 빌드하기 위한 출발점 역할을 합니다.
2. 메타데이터 속성 섹션에서 각 마이그레이션 패턴에 대한 메타데이터 속성 테이블을 생성하고에 식별된 메타데이터 속성으로 테이블을 채웁니다 [1단계: 필수 메타데이터 정의](#).
3. 소스 위치 섹션에서에서 식별한 소스를 문서화합니다 [메타데이터 소스 분석](#).
4. 소스 위치 액세스 지침 섹션에서 메타데이터 소스 위치에 액세스하기 위해 사용자가 따라야 하는 단계를 문서화합니다.
5. 메타데이터 스토어 섹션에서 사용자가에서 생성한 메타데이터 스토어에 액세스하기 위해 따라야 할 단계를 문서화합니다 [단일 메타데이터 스토어 정의](#).
6. 데이터 수집 유형 섹션에서 각 메타데이터 소스에 사용할 데이터 수집 프로세스를 식별합니다. 자동화 스크립트를 사용하여 모든 메타데이터 수집을 자동화하는 것이 가장 좋습니다.
7. 메타데이터별 데이터 수집 속성 섹션의 각 메타데이터 속성에 대한 지침에 따라 다음을 식별합니다. [메타데이터 수집 프로세스 정의](#)
 - a. 메타데이터 유형
 - b. 메타데이터 소스
 - c. 메타데이터 스토어
 - d. 컬렉션 유형
8. 메타데이터 수집 섹션에서 사용 사례에 따라 프로세스를 업데이트합니다. 이는 구현 단계에서 파도에 대한 메타데이터를 수집할 때 포트폴리오 워크스트림이 따르는 프로세스입니다.
9. 실행서가 완전하고 정확한지 확인합니다. 이 실행서는 마이그레이션 중에 사실의 출처여야 합니다.
10. 메타데이터 관리 런북을 팀과 공유하여 검토합니다.

작업 종료 기준

다음을 완료하면 다음 작업을 계속합니다.

- 수집된 메타데이터를 저장하기 위한 단일 리포지토리를 준비했습니다.
- 메타데이터 관리 실행서에서 다음을 정의하고 문서화했습니다.
 - 각 마이그레이션 패턴에 필요한 메타데이터 속성
 - 메타데이터 소스 및 각 소스에 액세스하는 방법에 대한 자세한 지침
 - 메타데이터 스토어 및 액세스 방법에 대한 자세한 지침
 - 메타데이터를 수집하는 데 사용되는 프로세스
 - 메타데이터 속성을 메타데이터 소스 및 수집 프로세스에 매핑하는 매핑 테이블

작업 3: 애플리케이션 우선 순위 지정 프로세스 정의

애플리케이션 우선 순위 지정은 애플리케이션을 클라우드로 마이그레이션할 순서를 결정하는 프로세스입니다. 애플리케이션을 클라우드로 마이그레이션하는 복잡성과 정의한 규칙에 따라 우선순위를 평가합니다. 애플리케이션 우선 순위에 대해 논의할 때 우선 순위가 높다고 해서 비즈니스에 대한 애플리케이션의 중요성과 반드시 상관관계가 있는 것은 아닙니다. 실제로 비즈니스 크리티컬 애플리케이션은 위험이 더 높기 때문에 일반적으로 비즈니스 크리티컬 애플리케이션은 마이그레이션 우선 순위가 낮습니다. 대규모 마이그레이션에서는 비즈니스에 중요하지 않은 복잡성이 낮은 애플리케이션의 우선 순위를 지정하고, 각 웨이브에서 점점 더 복잡하거나 비즈니스에 중요한 애플리케이션을 마이그레이션합니다.

수백 개의 애플리케이션이 마이그레이션을 위해 준비된 대규모 마이그레이션에서는 모든 애플리케이션의 우선 순위를 정하고 한 번에 모든 애플리케이션을 계획하지 않는 것이 좋습니다. 이는 대규모 마이그레이션 프로젝트에 애플리케이션 우선순위 지정 프로세스를 정의하는 것이 중요한 이유 중 하나입니다. 민첩한 방식으로 마이그레이션에 접근하려면 우선 순위가 가장 높은 애플리케이션(3~10개의 애플리케이션)을 선택하거나 3~5개의 웨이브에 충분한 애플리케이션을 선택할 수 있습니다. 그런 다음 선택한 애플리케이션에 대해서만 애플리케이션 검색 및 웨이브 계획을 완료합니다. 이 접근 방식은 대규모 마이그레이션 과정에서 애플리케이션 우선 순위와 파도가 자주 변경되기 때문에 상당한 시간을 절약합니다.

애플리케이션 우선 순위에 대한 일반적인 오해 중 하나는 우선 순위가 가장 높은 애플리케이션이 첫 번째 웨이브에 있어야 한다는 것입니다. 웨이브 계획을 수행할 때 우선 순위가 가장 높은 10개의 애플리케이션 중 일부만 첫 번째 웨이브에 있을 가능성이 높습니다. 다른 애플리케이션은 준비가 되지 않았기 때문입니다. 이는 종속성, 비즈니스 제약 또는 리소스 가용성과 같은 다양한 유효한 이유 때문일 수 있

습니다. 애플리케이션 우선 순위는 웨이브 계획의 중요한 요소이지만 고려하는 유일한 요소는 아닙니다.

이 작업에서는 애플리케이션 우선 순위 지정 프로세스 및 규칙을 정의합니다. 이 작업은 다음 단계로 구성됩니다.

- [1단계: 애플리케이션 우선 순위 지정 프로세스 정의](#)
- [2단계: 애플리케이션 우선 순위 지정 규칙 정의](#)
- [3단계: 애플리케이션 우선 순위 지정 프로세스 완료](#)

다음 섹션에서는 복잡성 점수에 대해 설명합니다. 이 플레이북은 애플리케이션의 우선 순위를 지정하는 방법에 대한 세 가지 프로세스 옵션을 제공하며, 세 가지 옵션 중 두 가지는 복잡성 점수를 사용합니다. 프로세스 옵션에 대한 자세한 내용은 섹션을 참조하세요 [1단계: 애플리케이션 우선 순위 지정 프로세스 정의](#). 애플리케이션 추천 프로세스를 사용하려는 경우 복잡성 점수 기준을 정의할 필요가 없으므로 바로 진행해야 합니다 [1단계: 애플리케이션 우선 순위 지정 프로세스 정의](#).

복잡성 점수 기준 정보

복잡성 점수는 애플리케이션 마이그레이션의 어려움을 평가하는 데 사용되는 프로세스로, 애플리케이션 우선 순위를 지정할 때 중요한 요소입니다. 복잡성 점수를 매기려면 사용자가 정의한 것과 동일한 비즈니스 및 기술 기준에 따라 모든 애플리케이션을 평가해야 합니다. 애플리케이션을 평가할 때 각 기준에 점수를 할당합니다. 비즈니스 기준 및 기술 기준의 점수를 합산하면 해당 애플리케이션 마이그레이션의 전반적인 복잡성을 반영하는 복잡성 점수를 받게 됩니다. 그런 다음 애플리케이션의 우선순위를 정하고 웨이브를 계획할 때 복잡성 점수를 사용할 수 있습니다.

복잡성 점수 기준에는 두 가지 범주가 있습니다.

- 비즈니스 기준 -이 범주의 기준은 애플리케이션을 사용할 수 없게 되는 경우의 위험, 보안 및 규정 준수 고려 사항, 리소스 가용성 등 애플리케이션 마이그레이션의 비즈니스 복잡성과 관련이 있습니다.
- 기술 기준 -이 범주의 기준은 운영 체제 및 버전, 서버 및 사용자 수, 마이그레이션 전략과 같은 애플리케이션 마이그레이션의 기술적 복잡성과 관련이 있습니다.

사용 사례에 적합한 점수 기준을 결정해야 합니다. 애플리케이션 복잡성을 수동으로 채점하는 경우 [포트폴리오 플레이북 템플릿](#)에서 애플리케이션 복잡성에 대한 점수 시트 템플릿(Microsoft Excel 형식)에는 표준 기준 및 점수 값 세트가 포함됩니다. 이러한 값으로 시작한 다음 사용 사례에 맞게 사용자 지정할 수 있습니다. 애플리케이션 우선 순위 지정에 검색 도구를 사용하는 경우 이러한 도구에는 일반적인

로 표준 기준 세트가 포함되며 기준을 추가, 제거 또는 수정할 수 있고 필요에 따라 가중치를 부여할 수 있습니다. 기준을 설정할 때 다음 두 섹션의 질문을 사용하여 기준을 구체화합니다.

비즈니스 기준

다음은 복잡성 점수에 일반적으로 사용되는 비즈니스 기준입니다.

비즈니스 기준	설명
비즈니스 영향	이 애플리케이션을 사용할 수 없게 되면 비즈니스에 미치는 영향을 평가합니다. - 재정적 영향이 있나요? - 운영에 영향을 미치는가? - 고객 경험에 영향을 미치나요? - 제품 또는 회사 이벤트에 영향을 미치나요?
직원 가용성	마이그레이션 중에 애플리케이션 소유자, 주제 전문가(SME), 네트워크 또는 인프라 관리자, 테스터 및 개발자의 지원이 필요할 수 있습니다. 마이그레이션 중에 도움이 되는 이러한 리소스의 가용성을 평가합니다. - 마이그레이션 중에이 직원이 마이그레이션 팀을 지원하고 지침을 제공할 수 있나요? - 마이그레이션 후이 직원이 애플리케이션을 테스트하고 검증할 수 있나요? - 이 직원이 대상 환경에서 애플리케이션을 실행하는 데 필요한 IP 주소 또는 포트를 프로비저닝할 수 있습니까?
비즈니스 복잡성	상호 의존적이고 상호 연결된 이해관계자, 정보 기술 시스템 및 조직 구조가 많으면 비즈니스 복잡성이 증가할 수 있습니다. 다음과 같이 비즈니스 복잡성을 평가합니다.

비즈니스 기준	설명
	• 방화벽 변경 또는 새 인스턴스 프로비저닝과 같은 인프라 및 네트워크 변경을 승인하는 데 얼마나 걸리나요? • 비즈니스에서 검색 도구와 같은 새 소프트웨어 또는 도구 설치를 승인하는 데 얼마나 걸리나요?
준비 상태	다음과 같이 애플리케이션을 마이그레이션할 준비가 되었는지 평가합니다. • 애플리케이션이 현재 기술 새로 고침을 진행 중이거나 수행할 예정입니까? • 유지 관리가 예약되어 있으며 예약된 마이그레이션과 겹치나요? • 애플리케이션이 폐기될 예정입니까? • 애플리케이션이 현재 업그레이드 중이며 새로운 기능이 개발 또는 통합되고 있습니까?

비즈니스 기준	설명
보안	다음과 같이 애플리케이션의 보안 요구 사항 및 보안 정책의 복잡성을 평가합니다. • 애플리케이션에 액세스하기 위해 IP 주소와 포트를 프로비저닝해야 합니까? • 애플리케이션에 인프라 보호가 필요합니까? • 애플리케이션에 데이터 보호가 필요합니까? • 키 관리가 필요합니까? • 애플리케이션에 특별한 액세스 관리 정책이 필요합니까? • 애플리케이션에 모니터링 또는 로깅이 필요합니까? • 애플리케이션에 인시던트 대응 프로세스 및 자동화가 필요합니까? • 이 애플리케이션에 알림 및 알림이 필요합니까?

비즈니스 기준	설명
규정 준수	주, 비즈니스 산업 또는 회사 정책에서 제공하는 법률, 규정 및 지침과 같은 규정 준수 요구 사항이 애플리케이션에 적용될 수 있습니다. 다음과 같이 애플리케이션 규정 준수 요구 사항의 복잡성을 평가합니다. • 데이터 프라이버시 및 레지던시 요구 사항이 있나요? • 애플리케이션의 저장 데이터를 암호화해야 합니까? • 애플리케이션과 주고받는 전송 중 데이터를 암호화해야 합니까? • 감사 로깅이 필요합니까? • 애플리케이션이 SOC(System and Organization Controls)와 같은 회계 및 재무 표준을 준수해야 합니까? • 애플리케이션이 Payment Card Industry(PCI)와 같은 결제 보안 표준을 준수해야 합니까? • 애플리케이션이 HIPAA(Health Insurance Portability and Accountability Act)와 같은 환자 건강 정보 규정을 준수해야 합니까? • 애플리케이션이 정보 시스템 보안 관리 및 평가 프로그램(ISMAP)과 같은 퍼블릭 클라우드 보안 프로그램을 준수해야 합니까?
애플리케이션 지식	애플리케이션 소유자와 같은 조직 내 누군가가 문제를 유지, 통합, 해결 및 수정할 수 있는 지식, 기술 및 경험을 가지고 있습니까? 그리고 비즈니스 수요에 맞게 애플리케이션을 확장할 수 있나요?
마이그레이션 기술	조직의 직원이 워크로드를 대상 환경으로 마이그레이션할 수 있는 기술을 갖추고 있습니까?

기술 기준

다음은 복잡성 점수에 일반적으로 사용되는 기술 기준입니다.

기술 기준	설명
스토리지	다음과 같이 애플리케이션의 현재 스토리지를 평가합니다. - 애플리케이션은 현재 어디에 저장되어 있습니까? 네트워크 연결 스토리지(NAS), 스토리지 영역 네트워크(SAN) 또는 로컬 드라이브를 예로 들 수 있습니다. - 현재 스토리지의 총 크기는 얼마입니까?
사용자 수	이 애플리케이션에는 몇 명의 사용자가 있습니까? 실제 로그 또는 건적을 사용할 수 있습니다.
서버 수	애플리케이션 스택에 몇 개의 서버가 있습니까?
연결	다음과 같이 이 애플리케이션이 조직의 다른 사용자와 어떻게 연결되어 있는지 평가합니다. - 이 애플리케이션에 종속되는 다른 애플리케이션은 몇 개입니까? - 이 애플리케이션을 사용할 수 없는 경우 다른 애플리케이션에 미치는 영향은 무엇입니까?
애플리케이션 OS 및 버전	다음과 같이 애플리케이션 서버의 운영 체제(OS) 및 버전을 평가합니다. - 서버의 OS 버전이 더 이상 지원되지 않나요? - 서버가 Unix 또는 Linux OS를 실행하고 있습니까? - 서버가 Windows Server OS를 실행하고 있습니까? - 애플리케이션이 메인프레임 또는 미드레인지 서버에 있습니까?

기술 기준	설명
애플리케이션 종속성	이 애플리케이션이 환경의 다른 리소스에 어떻게 의존하는지 평가합니다. - 이 애플리케이션은 어떤 리소스에 의존하나요? 리소스는 다른 애플리케이션, 구성 요소, OS별 서비스(예: 레지스트리 또는 웹 서버) 또는 라이브러리일 수 있습니다. - 이러한 리소스를 하나 이상 사용할 수 없는 경우 이 애플리케이션에 미치는 영향은 무엇입니까?
데이터 마이그레이션	이 애플리케이션의 데이터 또는 파일을 마이그레이션해야 하는지 여부를 평가합니다. - 데이터 마이그레이션은 얼마나 복잡합니까? - 파일 마이그레이션은 얼마나 복잡합니까?
마이그레이션 전략	선택한 마이그레이션 전략의 복잡성을 평가합니다. 마이그레이션 전략에 대한 자세한 내용은 AWS 대규모 마이그레이션 가이드를 참조하세요.
COTS 또는 사용자 지정	애플리케이션이 다음과 같이 사용자 지정으로 만들어졌는지 아니면 상용 off-the-shelf(COTS)인지 평가합니다. - 최신 버전의 소스 코드가 있습니까? - 애플리케이션 공급업체가 지원됩니까? - 애플리케이션이 아웃소싱됩니까?

1단계: 애플리케이션 우선 순위 지정 프로세스 정의

이 플레이북에는 애플리케이션 우선 순위를 지정하기 위한 세 가지 프로세스 옵션이 포함되어 있습니다. 옵션 중 하나를 선택하거나 둘 이상을 결합하여 사용자 지정 프로세스를 빌드하기로 결정할 수 있습니다. 사용 사례를 평가하고 다음 중 환경에 가장 적합한 방법을 결정합니다.

- **옵션 1: 수동 복잡성 점수** - 개인 또는 워크숍 스타일 세션에서 완료할 수 있는 수동 우선 순위 지정 프로세스입니다. 이 프로세스에서는 복잡성 점수 기준을 사용하여 애플리케이션 우선 순위 지정의 중요한 요소인 각 애플리케이션을 마이그레이션하는 데 따른 어려움을 평가합니다. 이 수동 프로세스는 대규모 애플리케이션 포트폴리오의 우선순위를 정하기 위한 일관되고 정량적인 접근 방식을 제공하므로 대규모 마이그레이션에 적합합니다. 그러나 정의된 기준 집합을 기준으로 각 애플리케이션을 평가하면 다른 두 옵션과 비교할 때 속도가 느려질 수 있습니다.
- **옵션 2: 애플리케이션 추천** - 일반적으로 워크숍 스타일 세션을 완료하는 수동 우선 순위 지정 프로세스입니다. 이 프로세스에서 애플리케이션 소유자는 마이그레이션을 위해 애플리케이션을 추천합니다. 이 프로세스를 성공적으로 수행하려면 애플리케이션 소유자가 각 애플리케이션에 대한 포괄적인 지식을 가지고 있어야 합니다. 이 프로세스는 시간이 요인이고 애플리케이션의 우선순위를 빠르게 지정해야 하는 경우에 권장됩니다.
- **옵션 3: 검색 도구** - 자동 우선 순위 지정 프로세스입니다. 환경의 검색 도구에 자동화된 애플리케이션 복잡성 점수 또는 우선순위 지정을 위한 자동화된 기능이 있는 경우 이 기능을 사용하면 시간을 절약하고 애플리케이션 우선순위 지정 프로세스를 가속화할 수 있습니다. 이 프로세스에서는 일반적으로 검색 도구의 파라미터 내에서 기준을 정의한 다음 도구가 애플리케이션을 분석하고 최종 복잡성 점수를 제공합니다. 이 옵션을 선택하기 전에 검색 도구에서 사용할 수 있는 기능을 살펴보고 사용 사례의 요구 사항에 맞게 사용자 지정할 수 있는지 확인합니다.

옵션 1: 수동 복잡성 점수

이 수동 애플리케이션 우선 순위 지정 프로세스에서는 워크시트를 사용하여 정의된 복잡성 점수 기준 세트를 기준으로 애플리케이션을 평가합니다. 워크숍 스타일 세션에서 워크시트를 완료하는 것이 좋습니다. 그렇지 않으면 이해관계자와 협력하여 개인이 워크시트를 완료할 수 있습니다. 그런 다음 최종 복잡성 점수와 애플리케이션 우선 순위 지정 규칙을 사용하여 애플리케이션의 우선 순위를 결정합니다. 수동 프로세스 중에서 애플리케이션 복잡성을 파악하고 해당 정보를 사용하여 애플리케이션의 우선 순위를 지정하는 가장 일관되고 양적인 접근 방식을 제공합니다.

이 프로세스의 복잡성 점수 평가 단계에서는 포트폴리오 플레이북 템플릿에서 사용할 수 있는 애플리케이션 복잡성(Excel 형식)에 대한 점수 시트 템플릿을 사용하는 것이 좋습니다. [samples/portfolio-playbook-templates.zip](https://aws.amazon.com/pt/playbook-templates/) 이 템플릿에는 사전 정의된 비즈니스 및 기술 기준이 포함되어 있습니다. 이러한 기준을 추가, 제거 또는 수정하거나 점수 값을 조정할 수 있습니다. 예를 들어 1~5 대신 1~10의 점수 범위를 선호할 수 있습니다. 제공된 템플릿에 대한 다음 사항에 유의하세요.

- 각 기준 위에 마우스를 올려 놓으면 해당 기준에 대한 설명을 볼 수 있습니다.
- 템플릿에 익숙한 경우 예제를 삭제해야 합니다. 이는 데모용일 뿐입니다.

마이그레이션의 초기화 및 구현 단계 전반에 걸쳐 복잡성 점수 시트를 최신 상태로 유지합니다. 포트폴리오 평가를 진행하면서 점수를 변경할 수 있습니다. 애플리케이션 심층 분석은 팀이 자세히 검토할 때 각 애플리케이션에 대해 자세히 알아보기 때문에 점수 시트를 업데이트하는 일반적인 시간입니다. 마이그레이션 중에 애플리케이션 마이그레이션을 방해하는 검색되지 않은 종속성 및 제한과 같은 문제가 발생하는 경우에도 애플리케이션 우선 순위를 변경할 수 있습니다. 마이그레이션 전반에 걸쳐 점수 시트를 유지 관리하면 애플리케이션의 우선 순위를 더 정확하게 지정할 수 있습니다.

다음과 같이 애플리케이션 우선 순위 지정 프로세스를 문서화합니다.

1. [포트폴리오 플레이북 템플릿](#)에서 애플리케이션 복잡성에 대한 점수 시트 템플릿을 엽니다.
2. 애플리케이션 시트에서 마이그레이션에 적합한 기준을 추가, 수정 또는 제거합니다. 기준을 수정할 때 다음을 수행합니다.
 - 이 플레이북의 [복잡성 점수 기준 정보](#) 섹션에 있는 지침을 검토합니다.
 - 각 기준이 마이그레이션 기간, 리소스 및 비용에 미치는 영향을 고려합니다.
 - 안정적인 복잡성 점수를 얻으려면 조직의 다양한 마이그레이션 복잡성 수준을 나타내는 기준을 포함하세요.
3. 점수 평가 가이드 시트에서 사용 사례에 따라 기본값과 기준을 업데이트합니다.
4. 점수 시트를 저장합니다.
5. 애플리케이션 우선 순위 실행서를 엽니다.
6. 애플리케이션 복잡성 점수 기준 섹션에서 점수 시트의 위치를 반영하도록 섹션을 업데이트합니다.
7. 애플리케이션 우선 순위 지정 프로세스 섹션에서 다음을 수행합니다.
 - a. 옵션 1 유지: 수동 복잡성 점수를 매기고 다른 옵션을 삭제합니다.
 - b. 사용 사례에 따라 프로세스를 수정합니다.
 - c. 이 단원에서 Option이라는 단어가 포함된 제목을 모두 제거합니다. 실행서에 이를 그대로 두면 사용자가 프로세스가 선택 사항이거나 여러 옵션을 사용할 수 있다고 생각하게 될 수 있습니다.
 - d. 애플리케이션 우선 순위 실행서를 저장합니다.

옵션 2: 애플리케이션 추천

이 수동 애플리케이션 우선 순위 지정 프로세스는 애플리케이션 우선 순위를 지정하는 가장 쉽고 빠른 접근 방식입니다. 이 프로세스에서는 애플리케이션 소유자에게 클라우드로 쉽게 마이그레이션할 수 있는 애플리케이션을 추천하도록 요청합니다. 그러면 지정된 애플리케이션에 대한 심층적인 지식이 이미 있으므로 사용자와 애플리케이션 소유자가 애플리케이션의 우선 순위를 빠르게 지정할 수 있습니다. 워크숍 스타일 세션에서 이해관계자와 협력하는 것이 좋지만 이메일, 공유 설명서 및 기타 커뮤니케이션 플랫폼을 통해 협업할 수도 있습니다.

추천 프로세스 중에 [포트폴리오 플레이북](#) 템플릿에 포함된 애플리케이션 복잡성(Excel 형식)에 대한 점수 시트 템플릿에 지정된 애플리케이션을 입력합니다. 이 템플릿의 모든 점수 및 기준 기능을 사용하지는 않지만 이 시트를 사용하여 추천 및 우선순위 결정을 기록하는 것이 좋습니다.

일부 상황에서는 애플리케이션 추천 프로세스를 사용하여 우선 순위를 가속화하므로 점수 시트가 필요하지 않을 수 있습니다. 예를 들어 소수의 애플리케이션만 우선시하거나 애플리케이션 소유자가 애플리케이션에 대해 매우 잘 알고 있는 경우 애플리케이션 소유자와 이해관계자는 지식과 경험을 기반으로 애플리케이션의 우선 순위를 지정할 수 있습니다. 이 경우 점수 시트 사용을 건너뛰고 우선 순위 지정으로 바로 진행할 수 있습니다.

다음과 같이 애플리케이션 우선 순위 지정 프로세스를 문서화합니다.

1. 애플리케이션 우선 순위 실행서를 엽니다.
2. 애플리케이션 복잡성 점수 기준 섹션을 삭제합니다. 이 프로세스는 애플리케이션 복잡성 점수를 사용하지 않습니다.
3. 애플리케이션 우선 순위 지정 프로세스 섹션에서 다음을 수행합니다.
 - a. 옵션 2 유지: 애플리케이션 추천 및 다른 옵션 삭제.
 - b. 사용 사례에 따라 프로세스를 수정합니다.
 - c. 이 단원에서 Option이라는 단어가 포함된 제목을 모두 제거합니다. 실행서에 이를 그대로 두면 사용자가 프로세스가 선택 사항이거나 여러 옵션을 사용할 수 있다고 생각하게 될 수 있습니다.
4. 애플리케이션 우선 순위 실행서를 저장합니다.

옵션 3: 검색 도구

검색 도구에 복잡성 점수 또는 애플리케이션 우선 순위 지정 기능이 있는 경우 이 자동화된 프로세스에는 리소스가 거의 필요하지 않으며 애플리케이션 우선 순위 지정 프로세스를 가속화할 수 있습니다. 검색 도구의 기준을 사용 사례에 맞게 사용자 지정하면 검색 도구가 애플리케이션을 자동으로 분석하고 최종 복잡성 점수를 제공합니다. 도구에는 이미 모든 애플리케이션 메타데이터가 있으므로 입력할 필요가 없습니다.

예를 들어 Flexera One Cloud Migration and Modernization(이전의 Flexera Foundation 및 CloudScape) 검색 도구에는 최적화 점수표라는 복잡성 점수 평가 기능이 있습니다. 이 기능을 사용하면 점수에 포함할 기준을 선택하고 기본 설정에 따라 각 기준을 평가할 수 있습니다. 데이터 검색이 완료되면 검색 도구는 사용자가 제공한 가중치 기반 기준에 따라 데이터를 분석하고 도구의 독점 공식을 사용하여 최종 복잡성 점수를 생성합니다. 자세한 내용은 [Foundation and CloudScape 사용 설명서](#)(Flexera 설명서) 및 [Optimization Scorecard](#)(Flexera 설명서)를 참조하세요.

이 프로세스의 단점은 환경에서 에이전트 없는 검색 도구에 대한 스캔 어플라이언스를 설정하거나 모든 범위 내 워크로드에 에이전트를 설치하는 데 시간이 걸린다는 것입니다(4~8주). 검색 도구에서 점수 기능을 사용하려면 먼저 애플리케이션 워크로드를 스캔하고 애플리케이션 스택 분석을 수행하여 검색 도구가 메타데이터를 수집할 수 있도록 추가 시간(4~12주)을 허용해야 합니다. 그러나 메타데이터 수집 및 애플리케이션 우선 순위 지정에 필요한 시간과 리소스를 줄여 검색 도구를 구성하는 데 필요한 추가 시간을 복구할 수 있습니다. 예를 들어 검색 도구의 데이터가 아직 최신 상태인 경우 포트폴리오 워크스트림은 검색 도구와 동원 단계의 데이터를 재사용하여 파일럿 애플리케이션을 식별할 수 있습니다.

Note

검색 도구 프로세스를 사용하는 경우에도 애플리케이션 복잡성에 대한 수동 점수 시트 템플릿을 사용하여 다른 기준 집합을 기준으로 애플리케이션을 분석할 수 있습니다. 이 추가 정보는 애플리케이션 우선 순위를 구체화하는 데 도움이 될 수 있습니다.

다음과 같이 애플리케이션 우선 순위 지정 프로세스를 문서화합니다.

1. 아직 설정하지 않은 경우 환경에서 검색 도구를 설정합니다. 자세한 내용은 AWS 권장 가이드 웹 사이트의 [자동 포트폴리오 검색 시작하기](#)를 참조하세요.
2. 도구의 지침에 따라 검색 도구에서 복잡성 점수 또는 애플리케이션 우선 순위 기준을 사용자 지정합니다. 기준 선택에 대한 자세한 내용은 섹션을 참조하세요 [복잡성 점수 기준 정보](#).
3. 애플리케이션 우선 순위 실행서를 엽니다.
4. 애플리케이션 복잡성 점수 기준 섹션에서 검색 도구에 점수 기준이 정의되어 있음을 반영하도록 섹션을 업데이트합니다. 예: 복잡성 점수 기준은 <검색 도구>에 정의되어 있습니다.
5. 애플리케이션 우선 순위 지정 프로세스 섹션에서 다음을 수행합니다.
 - a. 옵션 3: 검색 도구를 유지하고 다른 옵션을 삭제합니다.
 - b. 사용 사례에 따라 프로세스를 수정합니다. 복잡성 점수가 있는 보고서를 생성하는 방법에 대한 step-by-step 지침을 포함하는 것이 중요합니다. 사용 가능한 경우 사용 설명서 링크를 포함할 수 있습니다.
 - c. 이 섹션에서 Option이라는 단어가 포함된 제목을 모두 제거합니다. 실행서에 이를 그대로 두면 사용자가 프로세스가 선택 사항이거나 여러 옵션을 사용할 수 있다고 생각하게 될 수 있습니다.
6. 애플리케이션 우선 순위 실행서를 저장합니다.

2단계: 애플리케이션 우선 순위 지정 규칙 정의

이 단계에서는 애플리케이션의 마이그레이션 순서를 결정하는 데 도움이 되는 애플리케이션 우선 순위 지정 규칙을 정의합니다. 애플리케이션의 복잡성 점수는 애플리케이션의 우선순위를 정하고 웨이브를 계획하는 데 중요한 요소이지만 비즈니스 및 기술 요소도 고려해야 합니다. 각 애플리케이션의 우선순위를 평가하고 적절한 웨이브에서 애플리케이션을 예약하는 데 도움이 되는 규칙을 생성합니다. 일반적인 비즈니스 및 기술 규칙은 다음과 같습니다.

- 데이터 센터 마이그레이션 순서 및 일정 지정
- 사업부 우선 순위 지정
- 중요한 비즈니스 애플리케이션의 기한 캡처

다음과 같이 애플리케이션 우선 순위 지정 규칙을 정의합니다.

1. 애플리케이션 우선 순위 실행서를 엽니다.
2. 애플리케이션 우선 순위 지정 규칙 섹션에서 마이그레이션을 위한 사용자 지정 규칙을 추가합니다.
3. 애플리케이션 우선 순위 실행서를 저장합니다.
4. 애플리케이션 우선 순위 실행서의 규칙을 유지 관리합니다. 마이그레이션 진행 상황, 범위 또는 일정 이 변경되면 규칙이 변경될 수 있습니다.

다음은 애플리케이션 우선 순위 지정 규칙 세트의 예입니다.

우선순위	규칙
1	뉴욕 데이터 센터의 애플리케이션은 항상 텍사스 데이터 센터의 애플리케이션보다 우선 순위가 더 높아야 합니다.
2	IT 부서의 애플리케이션은 항상 마케팅 부서의 애플리케이션보다 우선 순위가 더 높아야 합니다.
3	복잡성 점수가 높은 애플리케이션은 우선 순위가 더 높아야 합니다.

우선순위	규칙
4	SAP 애플리케이션은 연말 이전에 마이그레이션해야 합니다.

3단계: 애플리케이션 우선 순위 지정 프로세스 완료

이제 포트폴리오 워크스트림이 규칙과 프로세스를 사용하여 애플리케이션의 우선순위를 지정하는 방법을 정의합니다. 이는 포트폴리오 워크스트림이 마이그레이션의 구현 단계에서 참조하는 프로세스입니다.

애플리케이션 우선 순위 실행서에서 다음과 같이 이 프로세스를 사용자 지정합니다.

1. 애플리케이션 우선 순위 실행서를 엽니다.
2. 2단계: 애플리케이션 우선순위 지정 섹션에서 사용 사례 및 환경에 맞게 프로세스를 수정합니다.
3. 애플리케이션 우선 순위 실행서를 저장합니다.

작업 종료 기준

다음을 완료하면 다음 작업을 계속합니다.

- 사용 가능한 옵션에서 애플리케이션 우선 순위 지정 프로세스를 선택했습니다.
- 애플리케이션 우선 순위 실행서에 다음을 문서화했습니다.
 - 애플리케이션 복잡성 점수 기준(해당하는 경우)
 - 애플리케이션 우선 순위 지정 프로세스
 - 애플리케이션 우선 순위 지정 규칙
- 애플리케이션 실행서의 2단계: 애플리케이션 우선순위 지정 섹션을 업데이트했습니다.

작업 4: 애플리케이션 심층 분석 프로세스 정의

이제 애플리케이션 우선 순위 지정을 위한 규칙 및 프로세스 설정을 완료했으므로 각 애플리케이션의 우선 순위를 구체화하는 데 도움이 되는 애플리케이션 심층 분석을 수행합니다. 가장 높은 우선 순위부터 가장 낮은 우선 순위까지 한 번에 한 애플리케이션에 대해 애플리케이션 심층 분석을 수행합니다. 포트폴리오 팀이 여러 개인 프로젝트의 경우 각 팀은 동시에 다른 애플리케이션에 대한 애플리케이션 심층 분석을 수행할 수 있습니다.

심층 분석 중에는 애플리케이션 마이그레이션의 복잡성에 영향을 미치는 종속성과 같은 예상치 못한 문제가 발생할 수 있습니다. 이 경우 이전 단계에서 정의한 복잡성 점수 기준을 수정하고 그에 따라 점수 시트를 업데이트하여 향후 애플리케이션에 대한 보다 정확한 복잡성 점수를 얻어야 합니다. 그런 다음 새로운 복잡성 점수를 사용하여 애플리케이션 우선 순위를 업데이트할 수 있습니다.

이 작업은 다음 단계로 구성됩니다.

- [1단계: 애플리케이션 워크숍 프로세스 정의](#)
- [2단계: 애플리케이션 매핑 프로세스 정의](#)
- [3단계: \(선택 사항\) 애플리케이션 대상 상태 정의](#)
- [4단계: 애플리케이션 심층 분석 프로세스 완료](#)

1단계: 애플리케이션 워크숍 프로세스 정의

워크숍 프로세스는 애플리케이션 심층 분석을 위한 가장 효율적인 접근 방식 중 하나입니다. 이 프로세스를 사용하면 이해관계자, 애플리케이션 소유자 및 포트폴리오 팀과 협력하여 애플리케이션을 평가하고 분석할 수 있습니다. 목표는 아키텍처, 비즈니스 목적, 종속성 및 환경을 포함하여 애플리케이션의 현재 상태를 명확하게 이해하는 것입니다. 그런 다음 애플리케이션의 크기와 복잡성에 대한 세부 정보를 사용하여 애플리케이션의 대상 상태를 설계합니다.

각 워크숍은 일반적으로 1~8시간 동안 지속되지만 복잡성이 높은 애플리케이션에는 추가 시간이 필요할 수 있습니다. 또한 리소스 가용성, 기본 설정, 애플리케이션의 크기 및 복잡성에 따라 워크숍을 여러 회의로 나눌 수 있습니다.

예상 결과 식별

애플리케이션 워크숍을 수행하기 전에 의제를 설정하고 워크숍의 예상 결과 또는 워크숍에서 수집해야 하는 정보를 정의해야 합니다. 이렇게 하면 워크숍 참가자가 워크숍을 준비하고, 회의를 대상으로 유지하며, 워크숍이 끝날 때까지 애플리케이션의 우선 순위 지정, 계획 수립 및 마이그레이션에 필요한 모든 정보를 확보할 수 있습니다.

표준 예상 결과 세트를 정의하고 애플리케이션 우선 순위 런북에 문서화하는 것이 좋습니다. 워크숍을 준비할 때 표준 예상 결과를 사용하고 특정 애플리케이션에 대해 새 결과를 추가합니다.

다음과 같이 애플리케이션 워크숍에 대한 표준 예상 결과 세트를 정의합니다.

1. 애플리케이션 우선 순위 실행서를 엽니다.
2. 애플리케이션 워크숍 예상 결과 섹션에서 애플리케이션 워크숍에 대한 표준 예상 결과 세트를 설정합니다. 워크숍을 준비할 때 애플리케이션의 특정 요구 사항에 맞게 사용자 지정할 수 있습니다.

3. 애플리케이션 우선 순위 실행서를 저장합니다.
4. 애플리케이션 우선 순위 실행서에서 예상 결과를 유지합니다. 애플리케이션 워크숍을 수행하고 포트폴리오 평가 및 웨이브 계획을 계속 진행하면서 새로운 예상 결과를 식별하거나 기존 결과를 구체화할 수 있습니다.

다음은 애플리케이션 워크숍에서 예상되는 결과의 예입니다.

우선순위	애플리케이션 워크숍의 예상 결과
1	연결된 서버, 종속성, 환경 및 애플리케이션 계층을 포함하여 애플리케이션의 현재 아키텍처에 대한 명확한 이해.
2	팀은 대상 아키텍처 설계를 지원하기 위해 메타데이터를 수집했습니다. 다음 메타데이터가 필요합니다. • 대상 AWS 계정 ID • 대상 AWS 리전 • 대상 서브넷 • 대상 보안 그룹
3	애플리케이션 소유자 설문이 완료되고 모든 주요 질문에 답변합니다.
4	팀은 사용 설명서, 애플리케이션 아키텍처 설명서, 테스트 설명서, 설계 설명서, 애플리케이션 프로그래밍 인터페이스(API) 설명서와 같은 모든 애플리케이션 설명서를 수집했습니다.

애플리케이션 워크숍 규칙 정의

애플리케이션 워크숍을 수행하기 전에 워크숍을 관리하는 규칙을 정의해야 합니다. 일반적인 규칙으로는 워크숍 기간, 워크숍에 필요할 수 있는 도구, 고려해야 할 일정 고려 사항 또는 기한 등이 있습니다. 이렇게 하면 회의를 대상으로 유지하고 워크숍에서 내린 결정이 마이그레이션 일정에 부합하도록 할 수 있습니다.

애플리케이션 우선 순위 실행서에 애플리케이션 워크숍 규칙을 문서화하는 것이 좋습니다.

다음과 같이 애플리케이션 워크숍 규칙을 정의합니다.

1. 애플리케이션 우선 순위 실행서를 엽니다.
2. 애플리케이션 워크숍 규칙 섹션에서 워크숍을 관리하는 규칙을 정의합니다.
3. 애플리케이션 우선 순위 실행서를 저장합니다.
4. 애플리케이션 우선 순위 실행서의 규칙을 유지 관리합니다. 애플리케이션 워크숍을 수행하고 포트폴리오 평가 및 웨이브 계획을 계속 진행하면서 새 규칙을 식별하거나 기존 규칙을 세분화할 수 있습니다.

다음은 애플리케이션 워크숍 규칙의 예입니다.

우선순위	워크숍 규칙
1	워크숍은 화요일과 목요일에 세션당 최대 2시간으로 예약해야 합니다.
2	인프라가 12월 1일부터 1월 15일까지 동결될 예정입니다.
3	마이그레이션 도구에 대한 실습 워크숍이 있습니다.
4	데이터 센터 계약은 3월 31일에 만료됩니다. 과태료 및 비용이 많이 드는 계약 연장을 방지하려면 3월 31일까지 워크로드를 제거해야 합니다.
5	생체 인식 애플리케이션과 시간 및 참석 애플리케이션은 유지됩니다.

애플리케이션 워크숍 프로세스 정의

애플리케이션 워크숍을 수행하기 위한 표준 프로세스를 정의하는 것이 중요합니다. 이렇게 하면 일관된 경험을 보장하고 워크숍 참가자에 대한 기대치를 설정하여 워크숍의 효율성을 개선할 수 있습니다.

애플리케이션 워크숍 프로세스에는 세 단계가 있습니다.

- 워크숍 준비 - 워크숍을 준비하면 세션이 원활하게 진행되고 참가자가 기대하는 바를 알 수 있습니다. 워크숍을 준비하기 위해 의제를 수립하고 예상 결과를 정의하고, 워크숍에 필요한 참가자, 도구 및 정보를 식별하고, 워크숍을 예약합니다. 최소 1주일 전에 워크숍을 예약하면 팀이 일정을 차단하고 워크숍을 준비하며 유용한 정보를 수집할 수 있습니다.
- 워크숍 수행 - 워크숍을 수행할 때 토론을 의제의 항목으로 제한하고 예상 결과를 충족하는지 확인합니다. 유용하다고 생각하지만 의제에 포함되지 않은 주제를 기록해 둡니다. 워크숍을 기록하는 것이 도움이 될 수 있습니다.
- 워크숍 결과 마무리 - 워크숍이 끝나면 팀은 애플리케이션의 현재 상태와 우선순위 지정 및 마이그레이션에 영향을 미칠 수 있는 잠재적 문제점, 위험, 과제 및 장애물을 명확하게 이해해야 합니다. 워크숍 이후의 일반적인 작업에는 애플리케이션의 우선 순위 다시 지정, 애플리케이션의 미래 상태 초안 작성, 다음 워크숍에서 유용할 수 있는 예상 결과, 규칙 또는 프로세스 변경 사항으로 런북 업데이트가 포함됩니다.

애플리케이션 우선순위 지정을 위한 런북 템플릿에는 애플리케이션 워크숍을 준비, 수행 및 마무리하기 위한 표준 step-by-step 프로세스가 포함되어 있습니다. 다음과 같이 애플리케이션 워크숍 프로세스를 정의합니다.

1. 애플리케이션 우선 순위 실행서를 엽니다.
2. 애플리케이션 워크숍 프로세스 섹션에서 사용 사례의 요구 사항에 맞게 표준 프로세스를 수정합니다.
3. 애플리케이션 우선 순위 실행서를 저장합니다.
4. 애플리케이션 우선 순위 실행서에서 프로세스를 유지 관리합니다. 애플리케이션 워크숍을 수행할 때 프로세스에 적용하려는 변경 사항을 식별할 수 있습니다.

단계 종료 기준

- 워크숍 의제와 워크숍을 지원하는 데 필요한 리소스 및 아티팩트를 정의했습니다.
- 워크숍의 예상 결과를 정의하고 워크숍에서 수집해야 할 정보를 식별했습니다.
- 워크숍 프로세스를 시험해 보고 애플리케이션 매핑을 지원하고 대상 상태를 설계하는 데 필요한 정보를 확보했습니다.
- 애플리케이션 우선 순위 실행서에 다음을 문서화했습니다.
 - 애플리케이션 워크숍 예상 결과
 - 애플리케이션 워크숍 규칙
 - 애플리케이션 워크숍 프로세스

2단계: 애플리케이션 매핑 프로세스 정의

애플리케이션 매핑은에서 식별하고 검증한 마이그레이션 패턴에 각 애플리케이션을 할당하는 프로세스입니다. [4단계: 마이그레이션 패턴 검증](#). 이 단계에서는 애플리케이션을 평가하는 데 사용할 규칙을 정의합니다. 그런 다음 각 애플리케이션을 평가하는 데 사용할 프로세스를 정의합니다. 각 애플리케이션을 마이그레이션 패턴의 사용 사례에 매핑하면 마이그레이션 도구, 마이그레이션을 완료하는 데 필요한 기술 및 마이그레이션 패턴의 복잡성을 식별하는 데 도움이 됩니다.

애플리케이션을 항상 단일 패턴으로 마이그레이션하는 것은 아닙니다. 동일한 애플리케이션에 여러 패턴이 필요할 수 있는 경우에 대한 자세한 내용은 이 섹션 [애플리케이션 매핑 프로세스 정의](#) 뒷부분의 섹션을 참조하세요.

애플리케이션 매핑 규칙

애플리케이션 매핑 규칙은 애플리케이션을 평가하고 적절한 마이그레이션 패턴을 식별하는 데 도움이 됩니다. 각 규칙은 애플리케이션을 평가하고 패턴의 기준을 일치시키는 데 사용해야 하는 모든 정보로 구성됩니다.

[포트폴리오 플레이북 템플릿](#)에서 애플리케이션 우선순위 지정에 위한 런북 템플릿에는 애플리케이션 매핑 규칙을 문서화하기 위한 섹션이 포함되어 있습니다. 다음과 같이 프로세스를 정의합니다.

1. 애플리케이션 우선 순위 실행서를 엽니다.
2. 애플리케이션 매핑 규칙 섹션에서 애플리케이션 매핑 규칙을 정의합니다.
3. 애플리케이션 우선 순위 실행서를 저장합니다.
4. 애플리케이션 우선 순위 실행서의 규칙을 유지 관리합니다.

다음 표에는 애플리케이션 매핑 규칙의 예가 나와 있습니다.

Note

이 테이블의 패턴 IDs와 이름은의 샘플 패턴에 해당합니다. [4단계: 마이그레이션 패턴 검증](#). 애플리케이션 우선 순위 실행서에 정의한 패턴 IDs와 이름을 사용합니다.

우선순위	매핑 규칙
1	사용률 데이터 또는 모니터링 도구를 사용하여 애플리케이션이 좀비 애플리케이션인지 유휴 애플리케이션인지 확인

우선순위	매핑 규칙
	플리케이션인지 식별합니다. 애플리케이션이 좀비 또는 유휴 애플리케이션인 경우 패턴 8: 애플리케이션 사용 중지를 선택한 다음 애플리케이션 스택의 서버를 종료합니다.
2	이 애플리케이션을 클라우드로 마이그레이션하는 것이 비즈니스 가치를 제공하는지 확인합니다. 온프레미스에서만 사용되고 시간 및 근태 애플리케이션과 같이 브랜치 또는 지리적 위치에서 공유되지 않는 애플리케이션은 일반적으로 클라우드로 마이그레이션할 필요가 없습니다. 이 애플리케이션을 마이그레이션해도 비즈니스 가치가 제공되지 않는 경우 패턴 9: 온프레미스 보존을 선택합니다.
3	애플리케이션의 운영 체제(OS)가 AWS 마이그레이션 서비스 AWS, 공급업체 또는 리호스팅 마이그레이션 도구에서 지원되는지 확인한 다음 다음을 수행합니다. • OS가 지원되는 경우 패턴 1: MGN 또는 Cloud Migration Factory를 사용하여 Amazon EC2로 리호스팅을 선택합니다. • OS가 지원되지 않는 경우 패턴 3:를 사용하여 Amazon EC2로 리플랫폼 CloudFormation을 선택합니다.
4	애플리케이션에 서비스형 소프트웨어(SaaS) 버전 또는 이에 상응하는 버전이 있는지 확인한 다음이 새 플랫폼으로 이전할 때의 이점과 비용을 평가합니다. 이러한 기준이 충족되면 패턴 10: 재구매 및 SaaS로 업그레이드를 선택합니다.

우선순위	매핑 규칙
5	온프레미스 Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션하거나 온프레미스 MySQL 데이터베이스를 Amazon Aurora MySQL-Compatible Edition 데이터베이스로 마이그레이션하는 등 애플리케이션의 온프레미스 데이터베이스를 클라우드의 동종 서비스로 마이그레이션할 수 있는지 확인합니다. 이러한 기준이 충족되면 패턴 2: AWS DMS 및를 사용하여 Amazon RDS로 리플랫폼 AWS SCT을 사용합니다.
6	애플리케이션이 Microsoft .NET Core(.NET 5 또는 .NET 6), Java, PHP 또는 다른 오픈 소스 프로그래밍 언어를 사용하는지 여부와 애플리케이션이 Microsoft Windows Server에서 호스팅되는지 여부를 식별합니다. 리플랫폼 비용이 타당한지 평가합니다. 이러한 기준이 충족되면 Amazon EC2에서 패턴 7: Windows에서 Linux로 리플랫폼을 선택합니다.
7	애플리케이션이 의존하는 온프레미스 로컬 및 공유 파일 스토리지를 식별한 다음 마이그레이션에 포함해야 하는지 여부를 결정합니다. 로컬 및 공유 파일 스토리지를 마이그레이션해야 하는 경우 패턴 4: AWS DataSync 또는를 사용하여 Amazon EFS로 리플랫폼 AWS Transfer Family을 선택합니다.
8	애플리케이션의 서버가 IBM AS/400 또는 Apache Spark와 같은 메인프레임 서버인지 미드레인지 서버인지 식별하고 애플리케이션이 에뮬레이터와 호환되는지 확인합니다. 이러한 기준이 충족되면 패턴 6: 에뮬레이터를 사용하여 메인프레임 또는 미드레인지 서버를 Amazon EC2로 리플랫폼합니다.

우선순위	매핑 규칙
9	제한으로 인해 더 이상 비즈니스 수요를 충족할 수 없는 레거시, 모놀리식 또는 메인프레임 기반 애플리케이션인지 식별합니다. 예를 들어 애플리케이션이 규모를 조정할 수 있는지, 관련 애플리케이션과 통합할 수 있는지, 비용이 많이 들고 유지 관리하기 어려운지 확인합니다. 애플리케이션이 이러한 기준 중 하나를 충족하는 경우 패턴 11: 애플리케이션 재설계를 선택합니다.

애플리케이션 매핑 프로세스 정의

애플리케이션을 마이그레이션 패턴에 매핑할 때는 검색 팀에 애플리케이션에 대한 검색 데이터를 요청하는 것이 좋습니다. 이 데이터에는 일반적으로 권장 마이그레이션 패턴(R 패턴 또는 R 점수라고도 함), 사용자 정보, 애플리케이션 종속성 및 평가에 사용할 수 있는 기타 정보와 같은 정보가 포함됩니다. 이 애플리케이션을 자세히 살펴보면서 이전에 식별된 마이그레이션 패턴을 변경하기로 결정할 수 있습니다.

데이터가 있으면 애플리케이션에서 식별한 비즈니스 및 기술 기준과 비교할 수 있습니다. [2단계: 비즈니스 및 기술 동인 식별](#). 애플리케이션 우선 순위 실행서에 드라이버를 기록했습니다. 기준을 기준으로 애플리케이션을 평가하면 애플리케이션에 대한 여러 마이그레이션 패턴을 선택하거나 비용, 일정 또는 기타 제한 사항에 따라 패턴을 제거할 수 있습니다.

다음은 단일 애플리케이션에서 여러 마이그레이션 패턴을 사용해야 하는 비즈니스 드라이버의 예입니다. 온프레미스 SQL Server 데이터베이스를 Amazon DynamoDB로 마이그레이션하려고 하지만 데이터 센터 계약이 만료되기 때문에 기업은 제안된 일정보다 빨리 마이그레이션하여 다시 플랫폼화하려고 합니다. 이 비즈니스 동인을 해결하려면 애플리케이션의 마이그레이션 계획을 두 가지 패턴으로 수정합니다. 먼저 애플리케이션을 클라우드로 리호스팅하여 데이터 센터에서 제거합니다. 나중에 애플리케이션이 클라우드에 있으면 제안된 일정에 따라 애플리케이션을 리플랫폼할 수 있습니다.

또한 애플리케이션이 여러 계층으로 구성된 애플리케이션인 n-티어 애플리케이션인지도 고려해야 합니다. 애플리케이션 계층은 애플리케이션의 가로 계층을 호스팅하는 물리적 서버 그룹입니다. N 티어 애플리케이션은 각 티어마다 다른 전략이 필요할 수 있으며 애플리케이션 티어를 다른 웨이브로 마이그레이션하도록 선택할 수 있기 때문에 더 복잡합니다. 예를 들어 프레젠테이션, 비즈니스 서비스 및 데이터베이스 계층으로 구성된 애플리케이션이 있는 경우 각 계층에 대해 다른 패턴을 매핑할 수 있습니다.

그런 다음 애플리케이션 우선 순위 런북에 정의한 애플리케이션 매핑 규칙을 기준으로 애플리케이션을 평가합니다. 자세한 내용은 이 섹션 앞부분에 있는 [애플리케이션 매핑 규칙](#) 내용을 참조하세요.

애플리케이션을 하나 이상의 패턴에 매핑한 후 애플리케이션 소유자와 함께이 결정을 검토하고 확인합니다. 애플리케이션 소유자는 선택한 패턴을 확인해야 하며 마이그레이션을 계획하고 수행하는 데 도움이 되어야 합니다. 현재 애플리케이션 소유자는 경험을 기반으로 인사이트를 제공하거나 마이그레이션과 관련하여 예상되는 문제를 공유할 수도 있습니다.

애플리케이션을 하나 이상의 마이그레이션 패턴에 매핑하고 애플리케이션 소유자와 패턴을 확인한 경우 애플리케이션 우선 순위 실행서의 애플리케이션 매핑 테이블에 애플리케이션, 패턴 ID, 패턴 이름 및 관련 드라이버를 기록합니다.

[포트폴리오 플레이북 템플릿](#)에서 애플리케이션 우선순위 지정을 위한 런북 템플릿에는 애플리케이션 매핑을 위한 표준 step-by-step 프로세스가 포함되어 있습니다. 다음과 같이 프로세스를 정의합니다.

1. 애플리케이션 우선 순위 실행서를 엽니다.
2. 애플리케이션 워크숍 프로세스 섹션에서 사용 사례의 요구 사항에 맞게 표준 프로세스를 수정합니다.
3. 애플리케이션 우선 순위 실행서를 저장합니다.
4. 애플리케이션 우선 순위 실행서에서 프로세스를 유지 관리합니다.

다음 표는 샘플 애플리케이션 매핑 테이블입니다. 애플리케이션 우선순위 지정을 위해 제공된 런북 템플릿에는 애플리케이션 매핑 프로세스의 결과를 기록할 수 있는 빈 테이블이 포함되어 있습니다.

Note

이 테이블의 패턴 IDs와 이름은의 샘플 패턴에 해당합니다. [4단계: 마이그레이션 패턴 검증](#). 애플리케이션 우선 순위 실행서에 정의한 패턴 IDs와 이름을 사용합니다.

애플리케이션 이름	패턴 ID	패턴 이름	비즈니스 및 기술 동인 해결
기업 웹 사이트	1	MGN 또는 Cloud Migration Factory를 사용하여 Amazon EC2로 리호스팅	• 데이터 센터 종료 • 운영 비용 절감

애플리케이션 이름	패턴 ID	패턴 이름	비즈니스 및 기술 동인 해결
HR 시스템	8	애플리케이션 사용 중지	• 운영 비용 절감
시간 및 출석 신청	9	온프레미스에 보관	• 운영 비용 절감 • 위험 및 영향 감소
PO 시스템	3	를 사용하여 Amazon EC2로 리플랫폼 CloudFormation	• 기술 통합 • 스토리지 및 컴퓨팅 제한 • 하드웨어 및 소프트웨어 end-of-life 지원 • 보안 및 규정 준수 개선
CRM 시스템	10	SaaS로 재구매 및 업그레이드	• 운영 비용 절감 • 기술 통합 • 하드웨어 및 소프트웨어 end-of-life 지원 • 개발, 혁신 및 성장 가속화
고정 자산 시스템	7	Amazon EC2에서 Windows에서 Linux로 리플랫폼	• 운영 비용 절감
ERP 파일 스토리지	4	AWS DataSync 또는 를 사용하여 Amazon EFS로 리플랫폼 AWS Transfer Family	• 스토리지 및 컴퓨팅 제한

애플리케이션 이름	패턴 ID	패턴 이름	비즈니스 및 기술 동인 해결
원장 시스템	6	에뮬레이터를 사용하여 메인프레임 또는 미드레인지 서버를 Amazon EC2로 리호스팅	• 데이터 센터 종료 • 기술 통합 • 보안 및 규정 준수 개선 • 하드웨어 및 소프트웨어 end-of-life 지원 • 스토리지 및 컴퓨팅 제한 • 애플리케이션 아키텍처 현대화
총계정원장	11	애플리케이션 재설계	• 운영 비용 절감 • 기술 통합 • 보안 및 규정 준수 개선 • 하드웨어 및 소프트웨어 end-of-life 지원 • 스토리지 및 컴퓨팅 제한 • 애플리케이션 아키텍처 현대화 • 확장성 및 복원력 • 개발, 혁신 및 성장 가속화

단계 종료 기준

- 애플리케이션 우선 순위 실행서에 다음을 문서화했습니다.
 - 애플리케이션 매핑 규칙

- 애플리케이션 매핑 프로세스
- 하나 이상의 proof-of-concept(POC) 애플리케이션을 사용하여 매핑 규칙 및 프로세스를 검증했습니다.

3단계: (선택 사항) 애플리케이션 대상 상태 정의

이 단계에서는 애플리케이션의 대상 상태 또는 예정 상태를 문서화하는 데 사용하는 속성과 프로세스를 정의합니다. 대상 상태는 마이그레이션 후 애플리케이션이 대상 클라우드 환경에서 작동하는 방식입니다. 대상 환경은 대상 플랫폼 또는 서비스 및 비즈니스 요구 사항에 따라 달라집니다. 대상 환경은 AWS 클라우드 또는 AWS Managed Services (AMS)일 수 있습니다.

대상 상태를 정의하면 프로젝트 관리자, 마이그레이션 컨설턴트, 아키텍트, 애플리케이션 소유자 및 이해관계자가 효과적으로 협업할 수 있습니다. 팀은 이 프로세스를 사용하여 문제를 미리 식별 및 해결하고 대상 상태 환경을 보다 효율적으로 구현할 수 있습니다.

일부 애플리케이션의 경우 이 단계는 선택 사항입니다. 마이그레이션하는 애플리케이션이 독립 실행형이거나 복잡성이 낮은 경우 이 단계를 건너뛸 수 있습니다. 리호스팅과 같이 애플리케이션을 수정하지 않는 마이그레이션 전략에는 이 단계가 필요하지 않을 수 있습니다. 그러나 리플랫폼 및 리아키텍트와 같은 보다 복잡한 마이그레이션 전략의 경우 마이그레이션을 시작하기 전에 대상 상태를 정의해야 합니다.

워크숍은 애플리케이션의 현재 상태를 심층적으로 이해하므로 워크숍을 완료한 후 대상 상태를 초안으로 작성하는 것이 좋습니다. 또한 애플리케이션을 마이그레이션 패턴에 매핑하면 추가 인사이트를 얻을 수 있으며 대상 상태를 정의하는 것이 필요한지 여부를 식별하는 데 도움이 됩니다. 예를 들어 MGN 또는 Cloud Migration Factory를 사용하여 애플리케이션을 Amazon EC2에 리호스팅 패턴에 매핑하는 경우 전략이 리호스팅되었음을 확인했으므로 이 애플리케이션의 대상 상태를 정의할 필요가 없을 수 있습니다.

대상 상태 속성

대상 상태를 정의하고 애플리케이션에 대한 결정을 내릴 때 다음 대상 상태 속성을 고려하는 것이 좋습니다.

- AWS Well-Architected Tool - AWS Well-Architected Framework와 비교하여 애플리케이션 대상 상태를 검토하여 클라우드에서 애플리케이션의 보안, 성능 및 복원력을 개선합니다.
- 대상 랜딩 존 - 일반적으로 [동원 단계가](#) 끝날 때까지 파일럿 애플리케이션을 실행할 준비가 된 완전한 랜딩 존을 구축했어야 합니다. 랜딩 존은 이미 다중 계정 아키텍처, ID 및 액세스 관리, 거버넌스,

데이터 보안, 네트워크 설계 및 로깅으로 구성되어야 합니다. 파일럿 애플리케이션을 사용하여 랜딩 존이 완료되었는지 확인합니다. 기존 대상 랜딩 존에서 파일럿 애플리케이션을 시작하고 실행할 수 있는지 확인합니다. 애플리케이션의 랜딩 존을 수정해야 하는 경우 랜딩 존 팀에 요구 사항을 알립니다. 예를 들어 애플리케이션에 별도의 계정에서 호스팅되는 서비스에 대한 액세스가 필요하거나 애플리케이션에 Virtual Private Cloud(VPC)로의 특수 라우팅이 필요할 수 있습니다.

- **종속성** - 애플리케이션이 제대로 작동하기 위해 의존하는 애플리케이션을 식별합니다. 예를 들어 애플리케이션은 데이터베이스, 스토리지 또는 결제 게이트웨이나 외부 웹 서비스와 같은 타사 서비스에 종속되거나 환경 내의 다른 애플리케이션에 종속될 수 있습니다. 이러한 종속성에 액세스하려면 인증을 위해 디렉터리 서비스에 연결하는 것과 같은 특별한 라우팅 또는 구성이 필요할 수 있습니다.
- **종속 애플리케이션** - 제대로 작동하기 위해 애플리케이션에 의존하는 애플리케이션을 식별합니다. 마이그레이션 중에 가동 중지가 발생하지 않도록 이러한 애플리케이션을 재구성하고 업데이트해야 할 수 있습니다.
- **보안 및 규정 준수** - 보안 및 규정 준수 팀과 대상 환경을 검토하고 격차를 식별합니다. 애플리케이션은 여러 구성 요소, 논리적 계층 또는 여러 계층으로 구성될 수 있습니다. 규정 준수 요구 사항에 따라 이러한 구성 요소 중 일부를 대상 환경으로 마이그레이션하지 못하거나 워크로드를 마이그레이션할 때 추가 보안 조치가 필요할 수 있습니다. 일반적인 보안 및 규정 준수 요구 사항은 데이터 레지던시, 전송 중 암호화, 저장 중 암호화입니다. 이를 위해서는 대상 환경에서 추가 구성이 필요합니다. 예를 들어 통신을 보호하기 위해 인증서를 구성해야 하거나 저장 데이터를 보호하기 위해 암호화 키가 필요할 수 있습니다. 또한 일부 애플리케이션 계층이 온프레미스에 남아 있고 다른 계층이 클라우드로 마이그레이션되도록 애플리케이션에 대해 여러 마이그레이션 패턴을 선택해야 할 수도 있습니다.
- **스토리지 종속성** - 애플리케이션 스토리지 종속성을 검토하고 애플리케이션을 대상 환경으로 마이그레이션하는 것이 이러한 종속성에 미치는 영향을 결정합니다. 예를 들어 애플리케이션이 네트워크 연결 스토리지(NAS) 또는 스토리지 영역 네트워크(SAN)와 같은 네트워크 스토리지에 의존하는 경우 Amazon Simple Storage Service(Amazon S3) 또는 Amazon FSx와 같은 클라우드의 스토리지 서비스를 계획해야 합니다. 또한 애플리케이션을 계속 실행하려면 데이터를 대상 클라우드 환경으로 마이그레이션하는 방법을 계획해야 합니다.
- **데이터베이스** - 애플리케이션이 사용하는 모든 데이터베이스의 마이그레이션 전략을 검토합니다. Amazon Relational Database Service(RDS), Amazon Aurora 또는 Amazon DynamoDB와 같은 새 데이터베이스 서비스로 리플랫폼하시겠습니까? DynamoDB 대상 환경에서 데이터베이스를 리호스팅하시겠습니까? 경우에 따라 특히 모놀리식 데이터베이스의 경우 1초 미만의 지연 시간과 같은 기술적 요구 사항을 해결하거나 특정 유형의 데이터베이스의 기능을 활용하기 위해 AWS 데이터베이스를 리팩터링해야 합니다. 데이터 레지던시 규정 준수 요구 사항과 마찬가지로 온프레미스에서 보존해야 하는 데이터와 클라우드로 마이그레이션해야 하는 데이터를 식별해야 합니다. 예를 들어 고객 정보를 위해 온프레미스 데이터베이스 테이블을 유지해야 할 수 있으며 나머지 데이터베이스를 클라우드로 마이그레이션할 수 있습니다.

- 애플리케이션 구성 요소 - 애플리케이션이 의존하는 구성 요소를 검토합니다. 애플리케이션이 대상 환경에서 지원하지 않는 구성 요소에 의존하는지 확인합니다. 대상 환경이 모든 애플리케이션 구성 요소를 지원하지 않는 경우 문제를 완화하려면 애플리케이션을 리팩터링해야 합니다. 예를 들어 Linux 운영 체제에서 애플리케이션을 리플랫폼하려면 Component Object Model(COM) Interop, COM+ 또는 Windows 레지스트리와 같은 Windows 전용 구성 요소에 의존하는 .NET Framework 애플리케이션이 있는 경우 애플리케이션을 .NET Core로 리팩터링해야 합니다.
- 애플리케이션 계층 - 애플리케이션의 계층 수를 식별합니다. 애플리케이션이 n-티어, 2-티어 또는 독립 실행형입니까? 각 계층의 마이그레이션 패턴을 이해하고 있는지 확인합니다. 예를 들어 애플리케이션에는 사용자 인터페이스를 호스팅하는 프레젠테이션(또는 웹) 계층, 비즈니스 서비스를 호스팅하는 애플리케이션 계층, 데이터베이스를 호스팅하는 데이터베이스 계층이 있을 수 있으며 각 계층에는 다른 마이그레이션 패턴이 필요할 수 있습니다.
- 재해 복구 - 복구 시점 목표(RPO) 및 복구 시간 목표(RTO)를 포함하여 애플리케이션의 현재 및 미래 상태 재해 복구(DR) 계획을 식별합니다. 기존 온프레미스 DR 계획을 사용할지 아니면 클라우드에서 새 DR 전략을 탐색할지 결정합니다. 자세한 내용은 [클라우드의 재해 복구 옵션](#) 및 클라우드의 재해 워크로드 복구 AWS백서의 복구 [목표\(RTO 및 RPO\)](#) 섹션을 참조하세요.

대상 상태 프로세스 정의

애플리케이션 대상 상태를 정의하려면 [포트폴리오 플레이북](#) 템플릿에서 사용할 수 있는 제공된 템플릿인 애플리케이션 대상 상태 워크시트(Excel 형식)를 사용하는 것이 좋습니다. 템플릿에는 사용하거나 수정할 수 있는 표준 속성이 포함되어 있습니다. 다음과 같이 애플리케이션 대상 상태를 문서화하는 프로세스를 정의합니다.

1. 애플리케이션 대상 상태 워크시트를 엽니다.
2. 기본 속성을 검토하고 사용 사례에 맞게 변경합니다.
3. 워크시트를 저장합니다.
4. 애플리케이션 우선 순위 실행서를 엽니다.
5. 대상 애플리케이션 상태 섹션에서 다음을 수행합니다.
 - a. 이 프로세스를 완료해야 하는 경우 섹션에서 포트폴리오 팀이 애플리케이션의 대상 상태를 정의해야 하는지 여부를 결정할 수 있는 기준을 설정합니다.
 - b. 필요에 따라 속성 섹션을 업데이트합니다.
 - c. 사용 사례에 따라 프로세스 섹션을 업데이트합니다.
6. 애플리케이션 우선 순위 실행서를 저장합니다.

애플리케이션 대상 상태의 샘플

다음 표는 애플리케이션 대상 상태 워크시트를 사용하여 애플리케이션의 대상 상태를 문서화하는 방법의 예를 보여줍니다.

애플리케이션	예제
대상 플랫폼	AWS 클라우드
랜딩 존	온프레미스 디렉터리 서비스에 대한 액세스가 필요합니다. 조직 전체에서 여러 계정 및 서비스의 관리를 AWS Control Tower 중앙 집중화해야 함
종속성	Active Directory, 결제 게이트웨이, 인벤토리 시스템
종속 애플리케이션	없음
보안	저장 데이터 및 전송 데이터 암호화
규정 준수	PCI, SOC
스토리지 종속성	부팅 드라이브 연결됨, NAS, 네트워크 공유
데이터베이스	현재: Oracle DB 클라우드: Amazon RDS for Oracle
애플리케이션 구성 요소	.NET Framework 4.5
애플리케이션 티어	N-계층 프론트엔드, 비즈니스 서비스, 데이터 서비스 및 에이전트, 데이터베이스
재해 복구	RPO: 1분, RTO: 5분 현재 DR 전략은 웜 스탠바이입니다.

애플리케이션	예제
	모든 미국 리전의 DR

단계 종료 기준

- 애플리케이션 대상 상태 워크시트에서 대상 상태 프로세스에 포함할 속성을 정의했습니다.
- 애플리케이션 우선 순위 실행서에서 다음을 수행했습니다.
 - 포트폴리오 팀이 애플리케이션의 대상 상태를 정의할 것으로 예상되는 시점에 대한 기준을 설정했습니다.
 - 사용 사례에 따라 대상 상태를 정의하는 프로세스를 업데이트했습니다.

4단계: 애플리케이션 심층 분석 프로세스 완료

이제 포트폴리오 워크스트림이 작업에서 설정한 워크숍, 규칙 및 프로세스를 사용하여 애플리케이션에 대한 심층 분석을 수행하는 방법을 정의합니다. 이는 포트폴리오 워크스트림이 마이그레이션의 구현 단계에서 참조하는 프로세스입니다.

애플리케이션 우선 순위 실행서에서 다음과 같이이 프로세스를 사용자 지정합니다.

1. 애플리케이션 우선 순위 실행서를 엽니다.
2. 2단계: 애플리케이션 심층 분석 수행 섹션에서 사용 사례 및 환경에 맞게 프로세스를 수정합니다.
3. 애플리케이션 우선 순위 실행서를 저장합니다.
4. 검토를 위해 애플리케이션 우선 순위 런북을 팀과 공유합니다.

작업 5: 웨이브 계획 프로세스 정의

웨이브 계획은 대규모 마이그레이션의 주요 마일스톤입니다. 웨이브 플랜에서는 인프라 및 애플리케이션 종속성(예: 공유 데이터베이스), 애플리케이션의 우선 순위, 애플리케이션 아키텍처의 유사성 및 비즈니스 기능을 고려하여 유사한 애플리케이션을 그룹화합니다. 그런 다음 애플리케이션 및 인프라 팀과 함께 웨이브 플랜을 검토하여 지정된 마이그레이션 및 전환 기간 동안 가용성을 확인합니다.

다음은 다양한 AWS 고객의 실제 배포를 기반으로 하는 웨이브 계획 모범 사례입니다.

- 마이그레이션 웨이브를 최소 4~5회 미리 계획합니다. 이렇게 하면 마이그레이션 워크스트림에 충분한 서버가 항상 확보됩니다.

- 빠른 실패. 복잡성이 낮은 몇 가지 애플리케이션으로 시작하여 학습한 내용을 이후 웨이브에 적용해야 합니다.
- 초기 파도(파도 1~5)에서는 더 적은 수의 서버(10개 미만), 복잡성이 낮은 애플리케이션, 개발 또는 테스트 환경과 같은 하위 환경의 애플리케이션을 선택합니다. 진행하면서 점점 더 복잡해지고 파도에 더 많은 서버가 도입됩니다.
- 웨이브 계획은 일회성 작업이 아닌 지속적인 프로세스입니다. 모든 웨이브를 한 번에 계획하지 마세요.
- 포트폴리오 검색 도구를 사용하고 있고 복잡성 점수 평가 기능이 있는 경우 웨이브 계획에서 사용하세요. 복잡성이 가장 낮은 애플리케이션을 먼저 마이그레이션합니다.

이 작업은 다음 단계로 구성됩니다.

- [1단계: 이동 그룹 프로세스 정의](#)
- [2단계: 웨이브 계획 선택 기준 정의](#)
- [3단계: 웨이브 계획 프로세스 완료](#)

1단계: 이동 그룹 프로세스 정의

이 단계에서는 application-to-server 종속성을 식별하고 이동 그룹으로 함께 이동해야 하는 서버를 결정하는 데 사용할 규칙을 정의합니다. 이동 그룹은 그룹에서 함께 이동해야 하는 서버 또는 애플리케이션의 블록입니다. 이는 마이그레이션 웨이브의 구성 요소로, 각 웨이브는 각 이동 그룹의 서버 수에 따라 하나 이상의 이동 그룹으로 구성됩니다.

애플리케이션 종속성 식별

다음은 이동 그룹에서 상호 종속적인 애플리케이션을 그룹화할 때 고려해야 할 주요 사항입니다.

- 다음과 같은 인프라 종속성을 고려합니다.
 - 애플리케이션에는 여러 데이터베이스가 있을 수 있으며 해당 데이터베이스는 다른 애플리케이션에서 공유할 수 있습니다.
 - 애플리케이션은 다른 애플리케이션에 종속될 수 있습니다.
 - 서버는 여러 애플리케이션에 대한 데이터베이스를 호스팅할 수 있습니다.
- 다음과 같은 비즈니스 및 운영 종속성을 고려합니다.
 - 비즈니스 영향 또는 운영 일정(예: 백업 또는 패치 적용)으로 인해 애플리케이션은 특정 기간 동안에만 마이그레이션할 수 있습니다.

- 애플리케이션 소유자는 하나의 마이그레이션 전환 기간에만 사용할 수 있으므로 모든 소유자의 애플리케이션은 동일한 이동 그룹에 있어야 합니다.

애플리케이션 워크숍 프로세스에서 또는 대상 상태를 정의할 때 인프라 종속성을 식별했습니다. 자동 또는 수동 프로세스를 통해 인프라 종속성을 식별할 수 있습니다. 인프라 종속성 식별을 자동화하려면 Flexera One Cloud Migration and Modernization 또는 TDS TransitionManager와 같은 검색 도구를 사용할 수 있습니다. 수동 프로세스의 경우 애플리케이션 및 인프라 팀과 CMDB 정보를 검증합니다.

애플리케이션 워크숍 프로세스에서 비즈니스 및 운영 종속성을 식별했습니다.

자체 웨이브 계획 실행서를 빌드하기 위한 출발점으로 포트폴리오 플레이북 템플릿에 포함된 웨이브 계획용 실행서 템플릿(Microsoft Word 형식)을 사용하는 것이 좋습니다. [samples/portfolio-playbook-templates.zip](https://samples.portfolio-playbook-templates.zip) 다음과 같이 마이그레이션에 대한 종속성을 문서화합니다.

1. 웨이브 계획 실행서를 엽니다.
2. 애플리케이션 종속성 섹션에서 종속성을 기록합니다. 유형(인프라, 비즈니스 또는 운영), 종속성 및 종속성에 대한 간략한 설명을 식별합니다.
3. 웨이브 계획 실행서를 저장합니다.
4. 웨이브 계획 실행서에서 종속성을 유지 관리합니다. 진행하면서 새로운 종속성을 식별할 수 있습니다.

다음 표에는 종속성의 예가 나와 있습니다.

유형	종속성	설명
인프라	Database	데이터베이스가 다른 애플리케이션과 공유됨
인프라	파일 스토어	애플리케이션은 분리할 수 있는 중앙 파일 스토어를 사용하거나 연결된 모든 애플리케이션을 함께 마이그레이션해야 합니다.
인프라	애플리케이션	애플리케이션은 추출, 변환 및 로드(ETL) 작업과 같이 작동하

유형	종속성	설명
		는 하나 이상의 다른 애플리케이션에 의존합니다.
업무	비즈니스 중단	애플리케이션에 대해 승인된 특정 중단 기간
Operational(작동)	패치 창	마이그레이션 전환에 영향을 미칠 수 있는 패치 적용과 같은 예약된 운영 작업

이동 그룹 규칙 정의

웨이브 계획 실행서에 종속성을 기록한 후에는 해당 종속성을 기반으로 이동 그룹 규칙을 빌드해야 합니다. 이러한 규칙은 서버를 이동 그룹으로 그룹화하는 방법을 제어합니다. 다음 단계에 따라 규칙을 빌드합니다.

1. 이전 섹션에서 정의한 종속성을 검토합니다.
2. 이동 그룹에서 애플리케이션을 함께 이동해야 하는지 여부에 영향을 미치는 종속성을 선택합니다. 모든 종속성에서 애플리케이션을 함께 마이그레이션해야 하는 것은 아닙니다. 예를 들어, 이동 그룹을 정의할 때 Microsoft Active Directory에 대한 인프라 종속성을 고려해서는 안 됩니다. 이는 모든 애플리케이션에 대한 일반적인 종속성이기 때문입니다. 애플리케이션을 마이그레이션하기 전에 클라우드에 도메인 컨트롤러를 빌드해야 합니다.
3. 애플리케이션을 함께 이동해야 하는 종속성을 이동 그룹 규칙으로 변환합니다.

애플리케이션이 규칙과 일치하는 경우 연결된 모든 서버를 동일한 이동 그룹에 배치하여 함께 마이그레이션해야 합니다.

마이그레이션을 위한 이동 그룹 규칙을 다음과 같이 문서화합니다.

1. 웨이브 계획 실행서를 엽니다.
2. 그룹 규칙 이동 섹션에서 우선순위에 따라 이동 그룹 규칙을 기록합니다.
3. 웨이브 계획 실행서를 저장합니다.
4. 웨이브 계획 실행서의 규칙을 유지 관리합니다. 진행하면서 새 규칙을 식별할 수 있습니다.

다음 표에는 이동 그룹 규칙의 예가 나와 있습니다.

규칙	그룹 규칙 이동
1	공유 데이터베이스가 있는 애플리케이션은 함께 마이그레이션해야 합니다.
2	애플리케이션 소유자가 동일한 애플리케이션은 함께 마이그레이션해야 합니다.
3	패치 기간이 동일한 애플리케이션은 함께 마이그레이션해야 합니다.

2단계: 웨이브 계획 선택 기준 정의

이동 그룹을 설정한 후에는 마이그레이션 웨이브를 구성하기 위해 유사한 이동 그룹을 함께 수집해야 합니다. 이 단계에서는 각 웨이브에 대해 하나 이상의 이동 그룹을 선택하는 데 사용하는 기준을 정의합니다.

각 이동 그룹의 크기를 이해하는 것은 성공적인 웨이브 계획에 매우 중요합니다. 목표는 마이그레이션이 민첩성을 유지하고 정상적인 서버 파이프라인을 유지하도록 각 웨이브의 크기를 조정하는 것입니다. 너무 큰 파도는 마이그레이션 계획의 변화에 적응하기 어려울 수 있으며, 너무 작은 파도는 원하는 마이그레이션 속도를 달성하기에 충분한 서버를 제공하지 못할 수 있습니다.

웨이브 크기를 조정할 때는 다음 기준을 고려하는 것이 좋습니다.

- 작은 첫 번째 파도 - 초기 파도는 10개 미만의 서버로 작아야 하며, 그런 다음 각 파도의 서버 수를 점진적으로 늘릴 수 있습니다. 이를 통해 빠르게 실패하고 학습한 교훈을 기반으로 구축할 수 있습니다. 예를 들어 서버가 20개인 애플리케이션을 마이그레이션하기 전에 서버가 3개인 애플리케이션을 마이그레이션합니다.
- 리소스 - 마이그레이션 팀이 단일 웨이브로 마이그레이션할 수 있는 서버 수를 식별합니다. 표준 조치는 4명의 아키텍트로 구성된 마이그레이션 팀이 리호스팅 패턴을 위해 일주일에 최대 50개의 서버를 마이그레이션할 수 있다는 것입니다. 이동 그룹을 결합하여 마이그레이션 팀의 용량을 초과하지 않는 마이그레이션 웨이브를 형성합니다.
- 민첩성 - 웨이브는 마이그레이션 계획의 변경 사항에 맞게 조정되어야 합니다. 서버를 다시 예약해야 하는 경우 영향을 받는 서버의 전체 이동 그룹을 다시 예약할 수 있어야 합니다.

- 스토리지 크기 - 더 작은 애플리케이션을 먼저 마이그레이션합니다. 예를 들어 2TB 애플리케이션보다 먼저 100GB 애플리케이션을 마이그레이션합니다.
- 애플리케이션 환경 - 개발 또는 테스트 환경과 같은 하위 환경에서 애플리케이션을 프로덕션 환경의 애플리케이션보다 먼저 마이그레이션합니다.
- 애플리케이션 복잡성 - 외부 종속성이 적은 덜 복잡한 애플리케이션을 먼저 마이그레이션합니다.
- 애플리케이션의 중요도 - 미션 크리티컬 애플리케이션보다 중요하지 않은 애플리케이션을 마이그레이션합니다.
- 사용자 기반 - 사용자 기반이 작은 애플리케이션을 먼저 마이그레이션합니다. 예를 들어 사용자가 10,000명인 애플리케이션 앞에 사용자가 10명인 애플리케이션을 마이그레이션합니다.
- 네트워크 대역폭 - 파도의 크기가 네트워크 대역폭을 초과해서는 안 됩니다. 자세한 내용은 [AWS 대규모 마이그레이션을 위한 Foundation 플레이북의 지침에 따라 정의한 마이그레이션 원칙을 참조하세요](#).

다음과 같이 웨이브 계획의 선택 기준을 문서화합니다.

1. 웨이브 계획 실행서를 엽니다.
2. 웨이브 계획 선택 기준 섹션에서 마이그레이션에 사용할 기준을 기록합니다.
3. 웨이브 계획 실행서를 저장합니다.
4. 웨이브 계획 실행서에서 기준을 유지 관리합니다. 진행하면서 기준을 조정하거나 새 기준을 추가해야 할 수 있습니다.

다음 표에는 웨이브 계획 선택 기준의 예가 나와 있습니다.

기준	설명
가장 복잡하지 않은 애플리케이션 식별	이동 그룹에서 복잡성 점수가 높은 애플리케이션을 식별합니다.
먼저 하위 환경	개발 또는 테스트 환경과 같은 하위 환경 내의 중요하지 않은 애플리케이션은 먼저 마이그레이션해야 합니다. 수익을 창출하는 애플리케이션과 같이 프로덕션 환경 내의 중요한 애플리케이션은 마지막으로 마이그레이션해야 합니다.
빠른 실패	서버가 10개 미만인 초기 웨이브를 구성합니다.

기준	설명
마이그레이션 팀 강점	각 마이그레이션 팀이 전환할 수 있는 서버 수를 식별합니다.
유사한 이동 그룹 결합	공통성을 기반으로 이동 그룹을 결합합니다. 예를 들어 이동 그룹은 동일한 애플리케이션 소유자, 소스 데이터 센터 또는 대상 AWS 계정을 공유할 수 있습니다.
웨이브 크기	파도는 총 50개의 서버를 초과해서는 안 됩니다.

단계 종료 기준

- 사용 사례에 대한 웨이브 계획 기준을 식별하여 웨이브 계획 실행서에 문서화했습니다.

3단계: 웨이브 계획 프로세스 완료

이제 이동 그룹을 생성하는 방법을 정의하고 이동 그룹을 마이그레이션 웨이브로 결합하는 데 사용한 기준을 설정했으므로 웨이브 계획 프로세스를 정의해야 합니다. 이 단계에서는 웨이브 계획 실행서를 업데이트하여 전체 웨이브 계획 프로세스를 기록하고 팀이 웨이브 정보를 기록하는 데 사용할 수 있는 대시보드 도구가 있는지 확인합니다.

이 단계에서는 제공된 대시보드 템플릿을 사용하여 웨이브 계획 및 마이그레이션을 수행하는 것이 좋습니다. 이 템플릿은 [포트폴리오 플레이북 템플릿](#)에서 사용할 수 있습니다. 이 템플릿은 포트폴리오 팀을 지원하도록 설계되었으며 데이터를 수집하고, 애플리케이션 포트폴리오를 분석하고, application-to-server 종속성을 식별하고, 결국 마이그레이션 웨이브를 계획하는 데 도움이 됩니다. 환경에 따라 이 템플릿을 수정할 수 있습니다.

다음과 같이 웨이브 계획 프로세스를 문서화합니다.

1. 웨이브 계획 및 마이그레이션을 위한 대시보드 템플릿을 엽니다.
2. 사용 사례에 따라 대시보드를 수정합니다. 예를 들어, VLOOKUP 함수를 사용하여 서버 인벤토리를 추출하기 위한 워크시트를 추가하거나, 새 피벗 테이블 또는 차트를 추가하거나, 소스 정보를 가져올 수 있습니다.
3. 대시보드 템플릿을 저장합니다.
4. 웨이브 계획 실행서를 엽니다.

5. 2단계: 웨이브 계획 수행 섹션에서 사용 사례의 요구 사항에 맞게 제공된 표준 프로세스를 수정합니다.
6. 웨이브 계획 실행서를 저장합니다.
7. 웨이브 계획 실행서를 팀과 공유하여 검토합니다.
8. 웨이브 계획 실행서에서 프로세스를 유지 관리합니다. 이 프로세스는 대규모 마이그레이션을 위한 웨이브를 계획하기 위한 표준 운영 절차 역할을 합니다.

작업 종료 기준

- 웨이브 계획 실행서에 다음을 문서화했습니다.
 - 애플리케이션 종속성
 - 우선 순위에 따라 나열된 애플리케이션 이동 그룹 규칙
 - 웨이브 계획 선택 기준
 - 웨이브 계획 프로세스

2단계: 대규모 마이그레이션 구현

대규모 마이그레이션을 초기화하는 1단계에서는 포트폴리오 평가 및 웨이브 계획 프로세스를 정의하고 이를 런북에 문서화했습니다. 대규모 마이그레이션을 구현하는 2단계에서는 이러한 프로세스를 완료하고 마이그레이션이 완료될 때까지 각 스프린트에 대해 프로세스를 반복합니다.

포트폴리오 팀은 2단계에서 다음 포트폴리오 평가 및 웨이브 계획 작업을 완료합니다.

- [작업 1: 애플리케이션 우선 순위 지정](#)
- [작업 2: 애플리케이션 심층 분석 수행](#)
- [작업 3: 웨이브 계획 및 메타데이터 수집 수행](#)

Note

포트폴리오 평가 및 웨이브 계획은 일회성 작업이 아닙니다. 마이그레이션을 지원하는 연속 작업입니다. 마이그레이션이 완료될 때까지 이 단계의 모든 작업을 여러 번 반복합니다.

포트폴리오 평가 및 웨이브 계획 프로세스는 일반적으로 각 웨이브마다 1~2주가 걸립니다. 포트폴리오 워크스트림은 일반적으로 마이그레이션 워크스트림을 위한 서버의 정상 파이프라인을 유지하기 위해 4~5개의 웨이브를 미리 계획합니다. 포트폴리오 워크스트림은 초기화 단계(1단계) 종료 시 웨이브 계획을 시작하고, 구현 단계(2단계)는 마이그레이션 워크스트림이 애플리케이션의 첫 번째 웨이브 마이그레이션을 시작할 때 시작됩니다. 파도 일정의 예는 [대규모 마이그레이션을 위한 가이드의 2단계: 대규모 마이그레이션 구현](#)을 참조하세요 AWS .

진행 상황 추적

마이그레이션을 위한 웨이브 준비를 시작할 때 포트폴리오 평가 프로세스를 통해 각 애플리케이션의 상태를 추적하는 것이 좋습니다. [포트폴리오 플레이북 템플릿](#)에서 포트폴리오 평가에 진행 상황 추적 템플릿(Microsoft Excel 형식)을 사용할 수 있습니다. 이 템플릿을 사용하면 각 애플리케이션에 대해 복잡성 점수, 대상 웨이브, 애플리케이션 소유자, 기본 작업(애플리케이션 우선순위 지정, 심층 분석, 웨이브 계획 및 데이터 수집)의 대상 완료 날짜, 마이그레이션을 위한 애플리케이션의 전체 준비 상태를 추적할 수 있습니다. 이 플레이북의 지침에는 진행 상황 추적 시트를 업데이트하는 시기에 대한 지침이 포함되어 있습니다.

작업 1: 애플리케이션 우선 순위 지정

이 작업에서는 포트폴리오의 마이그레이션되지 않은 애플리케이션 목록을 검토하고 나머지 애플리케이션의 하위 집합에 대해 애플리케이션 복잡성 점수와 우선 순위를 할당합니다. 마이그레이션 프로젝트 전반에 걸쳐이 프로세스를 여러 번 반복합니다.

이 작업을 완료하려면 다음 정보가 필요합니다.

입력	소스
마이그레이션하려는 포트폴리오의 전체 애플리케이션 목록	검색 도구 또는 구성 관리 데이터베이스(CMDB)
높은 수준의 대상 마이그레이션 전략 및 패턴	애플리케이션 우선 순위 런북의 마이그레이션 전략 및 마이그레이션 패턴
파도에 포함할 애플리케이션 수	웨이브 계획 실행서의 웨이브 계획 선택 기준

2단계: 애플리케이션 우선 순위 지정 섹션의 애플리케이션 우선 순위 지정 실행서의 지침을 따릅니다. 이 플레이북의에서이 프로세스를 정의했습니다[3단계: 애플리케이션 우선 순위 지정 프로세스 완료](#).

이 작업이 끝나면 다음을 완료했습니다.

출력	설명
우선 순위가 지정된 애플리케이션 목록	웨이브에 포함하려는 애플리케이션 수의 2~3배의 우선 순위를 지정했으며 진행률 트래커에 이러한 애플리케이션을 입력했습니다.

작업 2: 애플리케이션 심층 분석 수행

이 작업에서는 이전 작업에서 우선 순위를 지정한 각 애플리케이션에 대해 자세히 살펴봅니다. 여기에는 일반적으로 애플리케이션 소유자에게 설문지를 보내고, 애플리케이션 종속성을 분석하고, 애플리케이션 워크숍을 예약하는 것이 포함됩니다.

이 작업을 완료하려면 다음 정보가 필요합니다.

입력	소스
우선 순위가 지정된 애플리케이션 목록	의 구현 단계 초반에 생성된 작업 1: 애플리케이션 우선 순위 지정
높은 수준의 대상 마이그레이션 전략 및 패턴	애플리케이션 우선 순위 실행서의 마이그레이션 전략 및 마이그레이션 패턴

애플리케이션 우선 순위 런북의 2단계: 애플리케이션 심층 분석 수행 섹션의 지침을 따릅니다. 이 플레이북의에서이 프로세스를 정의했습니다[4단계: 애플리케이션 심층 분석 프로세스 완료](#).

이 작업이 끝나면 다음을 완료했습니다.

출력	설명
마이그레이션 패턴 매핑	각 애플리케이션을 마이그레이션 패턴에 매핑했습니다.
애플리케이션 대상 상태(해당하는 경우)	애플리케이션에 해당하는 경우 클라우드에서 애플리케이션의 미래 상태를 정의했습니다.

작업 3: 웨이브 계획 및 메타데이터 수집 수행

이는 포트폴리오 평가 및 웨이브 계획을 위한 마지막 작업입니다. 이 작업에서는 애플리케이션 정보와 대상 마이그레이션 패턴을 사용하여 이동 그룹을 구축하고, 이동 그룹을 웨이브에 할당하고, 마이그레이션을 지원하는 데 필요한 모든 메타데이터를 수집합니다. 마지막으로 마이그레이션 워크스트림에 웨이브가 준비되었음을 알립니다.

이 작업을 완료하려면 다음 정보가 필요합니다.

Input	소스
우선 순위가 지정된 애플리케이션 목록	의 구현 단계 초반에 생성된 작업 1: 애플리케이션 우선 순위 지정
마이그레이션 패턴 매핑	의 구현 단계 초반에 생성된 작업 2: 애플리케이션 심층 분석 수행

Input	소스
애플리케이션 대상 상태(해당하는 경우)	에서도 생성됨 작업 2: 애플리케이션 심층 분석 수행

해결 방법:

1. 2단계: 웨이브 계획 수행 섹션의 웨이브 계획 실행서의 지침을 따릅니다. 이 플레이북의에서이 프로세스를 정의했습니다[3단계: 웨이브 계획 프로세스 완료](#).
2. 2단계: 메타데이터 수집 섹션의 메타데이터 관리 실행서의 지침을 따릅니다. 이 플레이북의에서이 프로세스를 정의했습니다[3단계: 실행서에 메타데이터 요구 사항 및 수집 프로세스 문서화](#).
3. 웨이브 플랜이 완료되고 메타데이터가 준비되었음을 마이그레이션 워크스트림에 알립니다. 이 커뮤니케이션은 [AWS 대규모 마이그레이션을 위해 프로젝트 거버넌스 플레이북에 따라 정의한 거버넌스를 준수해야 합니다](#).

이 작업이 끝나면 다음을 완료했습니다.

출력	설명
웨이브 플랜	웨이브를 계획하고, 해당 웨이브에서 서버, 애플리케이션 및 데이터베이스를 식별하고, 시작 날짜와 전환 날짜 및 시간을 정의했습니다.
소스 인프라 메타데이터	서버 이름 및 운영 체제와 같은 소스 인프라 메타데이터를 수집했습니다.
대상 인프라 메타데이터	대상 서브넷, 보안 그룹 및 AWS 계정과 같은 대상 인프라 메타데이터를 수집했습니다.
알림 완료	웨이브 플랜과 메타데이터가 준비되었음을 마이그레이션 워크스트림에 알렸습니다.

포트폴리오 팀은 마이그레이션 프로젝트가 완료될 때까지 각 스프린트에 대해이 단계의 세 작업을 모두 반복합니다.

리소스

AWS 대규모 마이그레이션

대규모 마이그레이션을 위한 전체 AWS 규범적 지침 시리즈에 액세스하려면 [로의 대규모 마이그레이션을 참조하세요 AWS 클라우드](#).

추가 참조

도구 및 서비스

- [AWS 클라우드 마이그레이션 팩토리 솔루션](#)
- [의 무료 클라우드 마이그레이션 서비스 AWS](#)
- [AWS Database Migration Service](#)
- [를 사용하여 마이그레이션 AWS](#)
- [Flexera One Cloud 마이그레이션 및 현대화](#)(Flexera 웹 사이트)
- [TDS TransitionManager](#)(TDS 웹 사이트)

AWS 규범적 지침

- [클라우드 마이그레이션 팩토리를 통한 대규모 서버 마이그레이션 자동화](#)
- [로 마이그레이션하는 동안 사용 중지할 애플리케이션을 평가하는 모범 사례 AWS 클라우드](#)
- [마이그레이션 준비 상태 평가](#)
- [자동화된 포트폴리오 검색 시작하기](#)
- [Mobilize your organization to accelerate large-scale migrations](#)
- [Migration strategy for relational databases](#)
- [AWS 클라우드 마이그레이션을 위한 애플리케이션 포트폴리오 평가 가이드](#)

비디오

- [로 대규모 마이그레이션 실행 AWS](#)(AWS re:Invent 2020)
- [CloudEndure Migration Factory 모범 사례](#)(AWS re:Invent 2020)

기여자

다음 개인이 이 문서에 기여했습니다.

- Pratik Chunawala, Amazon Web Services의 보안 주체 클라우드 아키텍트
- Dwayne Bordelon, Amazon Web Services의 Senior Cloud Application Architect
- Rodolfo Jr. Cerrada, Amazon Web Services의 Senior Application Architect
- Wally Lu, Amazon Web Services의 보안 주체 컨설턴트

문서 기록

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

변경 사항	설명	날짜
에서 VMware Cloud 제거 AWS	에서 VMware Cloud에 대한 참조를 제거하고 일반적인 마이그레이션 전략 및 패턴 목록을 AWS 업데이트했습니다.	2024년 7월 5일
AWS 솔루션 이름 업데이트	참조된 AWS 솔루션의 이름을 CloudEndure Migration Factory에서 Cloud Migration Factory로 업데이트했습니다.	2022년 5월 2일
최초 게시	—	2022년 2월 28일

AWS 권장 가이드 용어집

다음은 AWS 권장 가이드에서 제공하는 전략, 가이드 및 패턴에서 일반적으로 사용되는 용어입니다. 용어집 항목을 제안하려면 용어집 끝에 있는 피드백 제공 링크를 사용하십시오.

숫자

7가지 전략

애플리케이션을 클라우드로 이전하기 위한 7가지 일반적인 마이그레이션 전략 이러한 전략은 Gartner가 2011년에 파악한 5가지 전략을 기반으로 하며 다음으로 구성됩니다.

- 리팩터링/리아키텍트 - 클라우드 네이티브 기능을 최대한 활용하여 애플리케이션을 이동하고 해당 아키텍처를 수정함으로써 민첩성, 성능 및 확장성을 개선합니다. 여기에는 일반적으로 운영 체제와 데이터베이스 이식이 포함됩니다. 예: 온프레미스 Oracle 데이터베이스를 Amazon Aurora PostgreSQL 호환 에디션으로 마이그레이션합니다.
- 리플랫폼(리프트 앤드 리세이프) - 애플리케이션을 클라우드로 이동하고 일정 수준의 최적화를 도입하여 클라우드 기능을 활용합니다. 예: 온프레미스 Oracle 데이터베이스를 AWS 클라우드의 Amazon Relational Database Service(Amazon RDS) for Oracle로 마이그레이션합니다.
- 재구매(드롭 앤드 쇼) - 일반적으로 기존 라이선스에서 SaaS 모델로 전환하여 다른 제품으로 전환합니다. 예: 고객 관계 관리(CRM) 시스템을 Salesforce.com으로 마이그레이션합니다.
- 리호스팅(리프트 앤드 시프트) - 애플리케이션을 변경하지 않고 클라우드로 이동하여 클라우드 기능을 활용합니다. 예: 온프레미스 Oracle 데이터베이스를 AWS 클라우드클라우드의 EC2 인스턴스에 있는 Oracle로 마이그레이션합니다.
- 재배포(하이퍼바이저 수준의 리프트 앤 시프트) - 새 하드웨어를 구매하거나, 애플리케이션을 다시 작성하거나, 기존 운영을 수정하지 않고도 인프라를 클라우드로 이동합니다. 온프레미스 플랫폼에서 동일한 플랫폼의 클라우드 서비스로 서버를 마이그레이션합니다. 예: Microsoft Hyper-V 애플리케이션을 로 마이그레이션합니다 AWS.
- 유지(보관) - 소스 환경에 애플리케이션을 유지합니다. 대규모 리팩터링이 필요하고 해당 작업을 나중에 연기하려는 애플리케이션과 비즈니스 차원에서 마이그레이션할 이유가 없어 유지하려는 레거시 애플리케이션이 여기에 포함될 수 있습니다.
- 사용 중지 - 소스 환경에서 더 이상 필요하지 않은 애플리케이션을 폐기하거나 제거합니다.

A

A2A(Agent-to-Agent)

작업 위임 및 상태 전송 agent-to-agent 공동 작업을 위한 상태 저장 프로토콜입니다.

ABAC

[속성 기반 액세스 제어](#)를 참조하세요.

추상화된 서비스

[관리형 서비스](#)를 참조하세요.

ACID

[원자성, 일관성, 격리성, 내구성](#)을 참조하세요.

능동-능동 마이그레이션

양방향 복제 도구 또는 이중 쓰기 작업을 사용하여 소스 데이터베이스와 대상 데이터베이스가 동기화된 상태로 유지되고, 두 데이터베이스 모두 마이그레이션 중 연결 애플리케이션의 트랜잭션을 처리하는 데이터베이스 마이그레이션 방법입니다. 이 방법은 일회성 전환이 필요한 대신 소규모의 제어된 배치로 마이그레이션을 지원합니다. 더 유연하지만 [액티브 패시브 마이그레이션](#)보다 더 많은 작업이 필요합니다.

능동-수동 마이그레이션

소스 데이터베이스와 대상 데이터베이스가 동기화된 상태로 유지되지만 소스 데이터베이스만 연결 애플리케이션의 트랜잭션을 처리하고 데이터는 대상 데이터베이스로 복제되는 데이터베이스 마이그레이션 방법입니다. 대상 데이터베이스는 마이그레이션 중 어떤 트랜잭션도 허용하지 않습니다.

에이전트

목표를 달성하기 위한 도구를 사용하여 자율적으로 추론, 계획 및 조치를 취할 수 있는 AI 시스템입니다.

에이전트 운영

대규모 프로덕션 환경에서 AI 에이전트를 구축, 테스트, 배포 및 실행하기 위한 운영 사례입니다.

집계 함수

행 그룹에서 작동하고 그룹에 대한 단일 반환 값을 계산하는 SQL 함수입니다. 집계 함수의 예로 SUM 및 MAX가 있습니다.

AI

[인공 지능](#)을 참조하세요.

AIOps

[인공 지능 운영](#)을 참조하세요.

익명화

데이터세트에서 개인 정보를 영구적으로 삭제하는 프로세스입니다. 익명화는 개인 정보 보호에 도움이 될 수 있습니다. 익명화된 데이터는 더 이상 개인 데이터로 간주되지 않습니다.

안티 패턴

솔루션이 다른 솔루션보다 비생산적이거나 비효율적이거나 덜 효과적이어서 반복되는 문제에 자주 사용되는 솔루션입니다.

애플리케이션 제어

맬웨어로부터 시스템을 보호하기 위해 승인된 애플리케이션만 사용하도록 허용하는 보안 접근 방식입니다.

애플리케이션 포트폴리오

애플리케이션 구축 및 유지 관리 비용과 애플리케이션의 비즈니스 가치를 비롯하여 조직에서 사용하는 각 애플리케이션에 대한 세부 정보 모음입니다. 이 정보는 [포트폴리오 탐색 및 분석 프로세스](#)의 핵심이며 마이그레이션, 현대화 및 최적화할 애플리케이션을 식별하고 우선순위를 정하는 데 도움이 됩니다.

인공 지능

컴퓨터 기술을 사용하여 학습, 문제 해결, 패턴 인식 등 일반적으로 인간과 관련된 인지 기능을 수행하는 것을 전문으로 하는 컴퓨터 과학 분야입니다. 자세한 내용은 [What is Artificial Intelligence?](#)를 참조하십시오.

인공 지능 운영(AIOps)

기계 학습 기법을 사용하여 운영 문제를 해결하고, 운영 인시던트 및 사용자 개입을 줄이고, 서비스 품질을 높이는 프로세스입니다. AWS 마이그레이션 전략에서 AIOps가 사용되는 방법에 대한 자세한 내용은 [운영 통합 가이드](#)를 참조하십시오.

비대칭 암호화

한 쌍의 키, 즉 암호화를 위한 퍼블릭 키와 복호화를 위한 프라이빗 키를 사용하는 암호화 알고리즘입니다. 퍼블릭 키는 복호화에 사용되지 않으므로 공유할 수 있지만 프라이빗 키에 대한 액세스는 엄격히 제한되어야 합니다.

원자성, 일관성, 격리성, 내구성(ACID)

오류, 정전 또는 기타 문제가 발생한 경우에도 데이터베이스의 데이터 유효성과 운영 신뢰성을 보장하는 소프트웨어 속성 세트입니다.

ABAC(속성 기반 액세스 제어)

부서, 직무, 팀 이름 등의 사용자 속성을 기반으로 세분화된 권한을 생성하는 방식입니다. 자세한 내용은 AWS Identity and Access Management (IAM) 설명서의 [용 ABAC AWS](#)를 참조하세요.

신뢰할 수 있는 데이터 소스

가장 신뢰할 수 있는 정보 소스로 간주되는 기본 버전의 데이터를 저장하는 위치입니다. 익명화, 편집 또는 가명화와 같은 데이터 처리 또는 수정의 목적으로 신뢰할 수 있는 데이터 소스의 데이터를 다른 위치로 복사할 수 있습니다.

가용 영역

다른 가용 영역의 장애로부터 격리 AWS 리전 되고 동일한 리전의 다른 가용 영역에 저렴하고 지연 시간이 짧은 네트워크 연결을 제공하는 내의 고유한 위치입니다.

AWS 클라우드 채택 프레임워크(AWS CAF)

조직이 클라우드로 성공적으로 전환하기 위한 효율적이고 효과적인 계획을 개발하는 AWS 데 도움이 되는 지침 및 모범 사례 프레임워크입니다. AWS CAF는 지침을 비즈니스, 사람, 거버넌스, 플랫폼, 보안 및 운영이라는 6가지 중점 영역으로 구성합니다. 비즈니스, 사람 및 거버넌스 관점은 비즈니스 기술과 프로세스에 초점을 맞추고, 플랫폼, 보안 및 운영 관점은 전문 기술과 프로세스에 중점을 둡니다. 예를 들어, 사람 관점은 인사(HR), 직원 배치 기능 및 인력 관리를 담당하는 이해관계자를 대상으로 합니다. 이러한 관점에서 AWS CAF는 성공적인 클라우드 채택을 위해 조직을 준비하는 데 도움이 되는 인력 개발, 교육 및 커뮤니케이션에 대한 지침을 제공합니다. 자세한 내용은 [AWS CAF 웹사이트](#)와 [AWS CAF 백서](#)를 참조하세요.

AWS 워크로드 검증 프레임워크(AWS WQF)

데이터베이스 마이그레이션 워크로드를 평가하고, 마이그레이션 전략을 권장하고, 작업 견적을 제공하는 도구입니다. AWS WQF는 AWS Schema Conversion Tool (AWS SCT)에 포함되어 있습니다. 데이터베이스 스키마 및 코드 객체, 애플리케이션 코드, 종속성 및 성능 특성을 분석하고 평가 보고서를 제공합니다.

B

악성 봇

개인 또는 조직을 방해하거나 해를 입히기 위한 [봇](#)입니다.

BCP

[비즈니스 연속성 계획](#)을 참조하세요.

동작 그래프

리소스 동작과 시간 경과에 따른 상호 작용에 대한 통합된 대화형 뷰입니다. Amazon Detective에서 동작 그래프를 사용하여 실패한 로그인 시도, 의심스러운 API 직접 호출 및 유사한 작업을 검사할 수 있습니다. 자세한 내용은 Detective 설명서의 [Data in a behavior graph](#)를 참조하십시오.

빅 엔디안 시스템

가장 중요한 바이트를 먼저 저장하는 시스템입니다. [엔디안](#)도 참조하세요.

바이너리 분류

바이너리 결과(가능한 두 클래스 중 하나)를 예측하는 프로세스입니다. 예를 들어, ML 모델이 “이 이메일이 스팸인가요, 스팸이 아닌가요?”, ‘이 제품은 책임가요, 자동차인가요?’ 등의 문제를 예측해야 할 수 있습니다.

블룸 필터

요소가 세트의 멤버인지 여부를 테스트하는 데 사용되는 메모리 효율성이 높은 확률론적 데이터 구조입니다.

블루/그린(Blue/Green) 배포

동일하지만 별개의 두 환경을 생성하는 배포 전략입니다. 하나의 환경(파란색)에서 현재 애플리케이션 버전을 실행하고 새 애플리케이션 버전은 다른 환경(녹색)에서 실행합니다. 이 전략을 사용하면 영향을 최소화하면서 신속하게 롤백할 수 있습니다.

bot

인터넷을 통해 자동화된 태스크를 실행하고 인적 활동이나 상호 작용을 시뮬레이션하는 소프트웨어 애플리케이션입니다. 인터넷에서 정보를 인덱싱하는 웹 크롤러와 같이 유용하거나 이로운 봇도 있습니다. 악성 봇이라고 하는 다른 일부 봇은 개인 또는 조직을 방해하거나 해를 입히기 위한 봇입니다.

봇넷

[맬웨어](#)에 감염되고 봇 허더 또는 봇 운영자와 같은 단일 당사자가 제어하는 [봇](#) 네트워크입니다. 봇넷은 봇의 규모와 봇의 영향 범위를 확대하는 가장 잘 알려진 메커니즘입니다.

브랜치

코드 리포지토리의 포함된 영역입니다. 리포지토리에 생성되는 첫 번째 브랜치가 기본 브랜치입니다. 기존 브랜치에서 새 브랜치를 생성한 다음 새 브랜치에서 기능을 개발하거나 버그를 수정할 수 있습니다. 기능을 구축하기 위해 생성하는 브랜치를 일반적으로 기능 브랜치라고 합니다. 기능을 출시할 준비가 되면 기능 브랜치를 기본 브랜치에 다시 병합합니다. 자세한 내용은 [About branches](#)(GitHub 설명서)를 참조하십시오.

긴급 액세스 권한

예외적인 상황에서 승인된 프로세스를 통해 사용자가 일반적으로 액세스할 권한이 없는데 액세스할 수 있는 빠른 방법입니다. 자세한 내용은 AWS Well-Architected 지침의 [Implement break-glass procedures](#) 지표를 참조하세요.

브라운필드 전략

사용자 환경의 기존 인프라 시스템 아키텍처에 브라운필드 전략을 채택할 때는 현재 시스템 및 인프라의 제약 조건을 중심으로 아키텍처를 설계합니다. 기존 인프라를 확장하는 경우 브라운필드 전략과 [그린필드](#) 전략을 혼합할 수 있습니다.

버퍼 캐시

가장 자주 액세스하는 데이터가 저장되는 메모리 영역입니다.

사업 역량

기업이 가치를 창출하기 위해 하는 일(예: 영업, 고객 서비스 또는 마케팅)입니다. 마이크로서비스 아키텍처 및 개발 결정은 비즈니스 역량에 따라 이루어질 수 있습니다. 자세한 내용은 백서의 [AWS에서 컨테이너화된 마이크로서비스 실행의 비즈니스 역량 중심의 구성화](#) 섹션을 참조하십시오.

비즈니스 연속성 계획(BCP)

대규모 마이그레이션과 같은 중단 이벤트가 운영에 미치는 잠재적 영향을 해결하고 비즈니스가 신속하게 운영을 재개할 수 있도록 지원하는 계획입니다.

C

CAF

[AWS Cloud Adoption Framework](#)를 참조하세요.

카나리 배포

최종 사용자에게 제공하는 느린 증분 릴리스 버전입니다. 확신이 들면 새 버전을 배포하고 현재 버전을 완전히 교체합니다.

CCoE

[클라우드 혁신 센터](#)를 참조하세요.

CDC

[데이터 캡처 변경](#)을 참조하세요.

변경 데이터 캡처(CDC)

데이터베이스 테이블과 같은 데이터 소스의 변경 내용을 추적하고 변경 사항에 대한 메타데이터를 기록하는 프로세스입니다. 대상 시스템의 변경 내용을 감사하거나 복제하여 동기화를 유지하는 등의 다양한 용도로 CDC를 사용할 수 있습니다.

카오스 엔지니어링

시스템의 복원력을 테스트하기 위해 의도적으로 장애나 중단 이벤트를 도입합니다. [AWS Fault Injection Service \(AWS FIS\)](#)를 사용하여 AWS 워크로드에 스트레스를 주고 응답을 평가하는 실험을 수행할 수 있습니다.

CI/CD

[지속적 통합 및 지속적 전송](#)을 참조하세요.

분류

예측을 생성하는 데 도움이 되는 분류 프로세스입니다. 분류 문제에 대한 ML 모델은 이산 값을 예측합니다. 이산 값은 항상 서로 다릅니다. 예를 들어, 모델이 이미지에 자동차가 있는지 여부를 평가해야 할 수 있습니다.

시민 개발자

전문 기술 없이 노코드/로우코드 플랫폼을 사용하여 AI 애플리케이션을 생성하는 비즈니스 사용자입니다.

클라이언트측 암호화

대상이 데이터를 AWS 서비스 수신하기 전에 로컬에서 데이터를 암호화합니다.

클라우드 혁신 센터(CCoE)

클라우드 모범 사례 개발, 리소스 동원, 마이그레이션 타임라인 설정, 대규모 혁신을 통한 조직 선도 등 조직 전체에서 클라우드 채택 노력을 추진하는 다분야 팀입니다. 자세한 내용은 AWS 클라우드 엔터프라이즈 전략 블로그의 [CCoE 게시물](#)을 참조하세요.

클라우드 컴퓨팅

원격 데이터 스토리지와 IoT 디바이스 관리에 일반적으로 사용되는 클라우드 기술 클라우드 컴퓨팅은 일반적으로 [엣지 컴퓨팅](#) 기술에 연결되어 있습니다.

클라우드 운영 모델

IT 조직에서 하나 이상의 클라우드 환경을 구축, 성숙화 및 최적화하는 데 사용되는 운영 모델입니다. 자세한 내용은 [클라우드 운영 모델 구축](#)을 참조하십시오.

클라우드 채택 단계

조직이 AWS 클라우드로 마이그레이션할 때 일반적으로 거치는 4단계는 다음과 같습니다.

- 프로젝트 - 개념 증명 및 학습 목적으로 몇 가지 클라우드 관련 프로젝트 실행
- 기반 - 클라우드 채택 확장을 위한 기초 투자(예: 랜딩 존 생성, CCoE 정의, 운영 모델 구축)
- 마이그레이션 - 개별 애플리케이션 마이그레이션
- Re-invention - 제품 및 서비스 최적화와 클라우드 혁신

이러한 단계는 Stephen Orban이 블로그 게시물 [The Journey Toward Cloud-First and the Stages of Adoption](#) on the AWS 클라우드 Enterprise Strategy 블로그에서 정의했습니다. AWS 마이그레이션 전략과 어떤 관련이 있는지에 대한 자세한 내용은 [마이그레이션 준비 가이드](#)를 참조하세요.

CMDB

[구성 관리 데이터베이스](#)를 참조하세요.

코드 리포지토리

소스 코드와 설명서, 샘플, 스크립트 등의 기타 자산이 버전 관리 프로세스를 통해 저장되고 업데이트되는 위치입니다. 일반적인 클라우드 리포지토리로 GitHub 또는 Bitbucket Cloud가 포함됩니다. 코드의 각 버전을 브랜치라고 합니다. 마이크로서비스 구조에서 각 리포지토리는 단일 기능 전용입니다. 단일 CI/CD 파이프라인은 여러 리포지토리를 사용할 수 있습니다.

콜드 캐시

비어 있거나, 제대로 채워지지 않았거나, 오래되었거나 관련 없는 데이터를 포함하는 버퍼 캐시입니다. 주 메모리나 디스크에서 데이터베이스 인스턴스를 읽어야 하기 때문에 성능에 영향을 미치며, 이는 버퍼 캐시에서 읽는 것보다 느립니다.

콜드 데이터

거의 액세스되지 않고 일반적으로 과거 데이터인 데이터. 이런 종류의 데이터를 쿼리할 때는 일반적으로 느린 쿼리가 허용됩니다. 이 데이터를 성능이 낮고 비용이 저렴한 스토리지 계층 또는 클래스로 옮기면 비용을 절감할 수 있습니다.

컴퓨터 비전(CV)

기계 학습을 사용하여 디지털 이미지 및 비디오와 같은 시각적 형식에서 정보를 분석하고 추출하는 [AI](#) 필드입니다. 예를 들어 Amazon SageMaker AI는 CV에 대한 이미지 처리 알고리즘을 제공합니다.

구성 드리프트

워크로드의 경우 구성이 예상되는 상태에서 변경됩니다. 이로 인해 워크로드가 규정을 준수하지 않을 수 있으며, 이는 일반적으로 점진적이고 의도되지 않은 작업입니다.

구성 관리 데이터베이스(CMDB)

하드웨어 및 소프트웨어 구성 요소와 해당 구성을 포함하여 데이터베이스와 해당 IT 환경에 대한 정보를 저장하고 관리하는 리포지토리입니다. 일반적으로 마이그레이션의 포트폴리오 탐색 및 분석 단계에서 CMDB의 데이터를 사용합니다.

규정 준수 팩

규정 준수 및 보안 검사를 사용자 지정하기 위해 조합할 수 있는 AWS Config 규칙 및 문제 해결 작업의 모음입니다. YAML 템플릿을 사용하여 적합성 팩을 AWS 계정 및 리전 또는 조직 전체에 단일 엔터티로 배포할 수 있습니다. 자세한 내용은 AWS Config 설명서의 [적합성 팩](#)을 참조하세요.

지속적 통합 및 지속적 전달(CI/CD)

소프트웨어 릴리스 프로세스의 소스, 빌드, 테스트, 스테이징 및 프로덕션 단계를 자동화하는 프로세스입니다. CI/CD는 일반적으로 파이프라인으로 설명됩니다. CI/CD를 통해 프로세스를 자동화하고, 생산성을 높이고, 코드 품질을 개선하고, 더 빠르게 제공할 수 있습니다. 자세한 내용은 [지속적 전달의 이점](#)을 참조하십시오. CD는 지속적 배포를 의미하기도 합니다. 자세한 내용은 [지속적 전달\(Continuous Delivery\)](#)과 [지속적인 개발](#)을 참조하십시오.

CV

[컴퓨터 비전](#)을 참조하세요.

D

저장 데이터

스토리지에 있는 데이터와 같이 네트워크에 고정되어 있는 데이터입니다.

데이터 분류

중요도와 민감도를 기준으로 네트워크의 데이터를 식별하고 분류하는 프로세스입니다. 이 프로세스는 데이터에 대한 적절한 보호 및 보존 제어를 결정하는 데 도움이 되므로 사이버 보안 위험 관리 전략의 중요한 구성 요소입니다. 데이터 분류는 AWS Well-Architected Framework의 보안 원칙 구성 요소입니다. 자세한 내용은 [데이터 분류](#)를 참조하십시오.

데이터 드리프트

프로덕션 데이터와 ML 모델 학습에 사용된 데이터 간의 상당한 차이 또는 시간 경과에 따른 입력 데이터의 의미 있는 변화. 데이터 드리프트는 ML 모델 예측의 전반적인 품질, 정확성 및 공정성을 저하시킬 수 있습니다.

전송 중 데이터

네트워크를 통과하고 있는 데이터입니다. 네트워크 리소스 사이를 이동 중인 데이터를 예로 들 수 있습니다.

데이터 메시

중앙 집중식 관리 및 거버넌스를 통해 분산되고 탈중앙화된 데이터 소유권을 제공하는 아키텍처 프레임워크입니다.

데이터 최소화

꼭 필요한 데이터만 수집하고 처리하는 원칙입니다. 에서 데이터를 최소화하면 개인 정보 보호 위험, 비용 및 분석 탄소 발자국을 줄일 AWS 클라우드 수 있습니다.

데이터 경계

신뢰할 수 있는 자격 증명만 예상 네트워크에서 신뢰할 수 있는 리소스에 액세스하도록 하는 데 도움이 되는 AWS 환경의 예방 가드레일 세트입니다. 자세한 내용은 [데이터 경계 구축을 참조하세요 AWS](#).

데이터 사전 처리

원시 데이터를 ML 모델이 쉽게 구문 분석할 수 있는 형식으로 변환하는 것입니다. 데이터를 사전 처리한다는 것은 특정 열이나 행을 제거하고 누락된 값, 일관성이 없는 값 또는 중복 값을 처리함을 의미할 수 있습니다.

데이터 출처

라이프사이클 전반에 걸쳐 데이터의 출처와 기록을 추적하는 프로세스(예: 데이터 생성, 전송, 저장 방법).

데이터 주체

데이터를 수집 및 처리하는 개인입니다.

데이터 웨어하우스

분석과 같은 비즈니스 인텔리전스를 지원하는 데이터 관리 시스템입니다. 데이터 웨어하우스에는 보통 많은 양의 기록 데이터가 포함되며 일반적으로 쿼리 및 분석에 사용됩니다.

데이터 정의 언어(DDL)

데이터베이스에서 테이블 및 객체의 구조를 만들거나 수정하기 위한 명령문 또는 명령입니다.

데이터베이스 조작 언어(DML)

데이터베이스에서 정보를 수정(삽입, 업데이트 및 삭제)하기 위한 명령문 또는 명령입니다.

DDL

[데이터 정의 언어](#)를 참조하세요.

딥 앙상블

예측을 위해 여러 딥 러닝 모델을 결합하는 것입니다. 딥 앙상블을 사용하여 더 정확한 예측을 얻거나 예측의 불확실성을 추정할 수 있습니다.

딥 러닝

여러 계층의 인공 신경망을 사용하여 입력 데이터와 관심 대상 변수 간의 매핑을 식별하는 ML 하위 분야입니다.

심층 방어

네트워크와 그 안의 데이터 기밀성, 무결성 및 가용성을 보호하기 위해 컴퓨터 네트워크 전체에 일련의 보안 메커니즘과 제어를 신중하게 계층화하는 정보 보안 접근 방식입니다. 이 전략을 채택하면 AWS Organizations 구조의 여러 계층에 여러 컨트롤을 AWS 추가하여 리소스를 보호할 수 있습니다. 예를 들어, 심층 방어 접근 방식은 다단계 인증, 네트워크 세분화 및 암호화를 결합할 수 있습니다.

위임된 관리자

에서 AWS Organizations 호환되는 서비스는 AWS 멤버 계정을 등록하여 조직의 계정을 관리하고 해당 서비스에 대한 권한을 관리할 수 있습니다. 이러한 계정을 해당 서비스의 위임된 관리자라고

합니다. 자세한 내용과 호환되는 서비스 목록은 AWS Organizations 설명서의 [AWS Organizations와 함께 사용할 수 있는 AWS 서비스](#)를 참조하십시오.

배포

대상 환경에서 애플리케이션, 새 기능 또는 코드 수정 사항을 사용할 수 있도록 하는 프로세스입니다. 배포에는 코드 베이스의 변경 사항을 구현한 다음 애플리케이션 환경에서 해당 코드베이스를 구축하고 실행하는 작업이 포함됩니다.

개발 환경

[환경](#)을 참조하세요.

탐지 제어

이벤트 발생 후 탐지, 기록 및 알림을 수행하도록 설계된 보안 제어입니다. 이러한 제어는 기존의 예방적 제어를 우회한 보안 이벤트를 알리는 2차 방어선입니다. 자세한 내용은 AWS에서 보안 제어 구현의 [탐지 제어](#)를 참조하세요.

개발 가치 흐름 매핑 (DVSM)

소프트웨어 개발 라이프사이클에서 속도와 품질에 부정적인 영향을 미치는 제약 조건을 식별하고 우선 순위를 지정하는 데 사용되는 프로세스입니다. DVSM은 원래 린 제조 방식을 위해 설계된 가치 흐름 매핑 프로세스를 확장합니다. 소프트웨어 개발 프로세스를 통해 가치를 창출하고 이동하는 데 필요한 단계와 팀에 중점을 둡니다.

디지털 트윈

건물, 공장, 산업 장비 또는 생산 라인과 같은 실제 시스템을 가상으로 표현한 것입니다. 디지털 트윈은 예측 유지 보수, 원격 모니터링, 생산 최적화를 지원합니다.

차원 테이블

[스타 스키마](#)에서 팩트 테이블의 정량적 데이터에 대한 데이터 속성을 포함하는 더 작은 테이블을 말합니다. 차원 테이블 속성은 일반적으로 텍스트 필드나 텍스트처럼 동작하는 개별 숫자입니다. 이러한 속성은 보통 쿼리 제약, 필터링 및 결과 세트 레이블 지정에 사용됩니다.

재해

워크로드 또는 시스템이 기본 배포 위치에서 비즈니스 목표를 달성하지 못하게 방해하는 이벤트입니다. 이러한 이벤트는 자연재해, 기술적 오류, 의도하지 않은 구성 오류 또는 멀웨어 공격과 같은 사람의 행동으로 인한 결과일 수 있습니다.

재해 복구(DR)

[재해](#)로 인한 가동 중지 시간 및 데이터 손실을 최소화하기 위해 사용하는 전략 및 프로세스입니다. 자세한 내용은 AWS Well-Architected Framework의 [Disaster Recovery of Workloads on AWS: Recovery in the Cloud](#)를 참조하세요.

DML

[데이터베이스 조작 언어](#)를 참조하세요.

도메인 기반 설계

구성 요소를 각 구성 요소가 제공하는 진화하는 도메인 또는 핵심 비즈니스 목표에 연결하여 복잡한 소프트웨어 시스템을 개발하는 접근 방식입니다. 이 개념은 에릭 에반스에 의해 그의 저서인 도메인 기반 디자인: 소프트웨어 중심의 복잡성 해결(Boston: Addison-Wesley Professional, 2003)에서 소개되었습니다. Strangler Fig 패턴과 함께 도메인 기반 설계를 사용하는 방법에 대한 자세한 내용은 [컨테이너 및 Amazon API Gateway를 사용하여 기존의 Microsoft ASP.NET\(ASMX\) 웹 서비스를 점진적으로 현대화하는 방법](#)을 참조하십시오.

DR

[재해 복구](#)를 참조하세요.

드리프트 감지

기준이 되는 구성과의 편차 추적을 말합니다. 예를 들어 AWS CloudFormation 를 사용하여 [시스템 리소스의 드리프트를 감지](#)하거나 사용하여 AWS Control Tower 거버넌스 요구 사항 준수에 영향을 미칠 수 있는 [랜딩 존의 변경 사항을 감지](#)할 수 있습니다.

DVSM

[개발 가치 흐름 매핑](#)을 참조하세요.

E

EDA

[탐색 데이터 분석](#)을 참조하세요.

EDI

[전자 데이터 교환](#)을 참조하세요.

엣지 컴퓨팅

IoT 네트워크의 엣지에서 스마트 디바이스의 컴퓨팅 성능을 개선하는 기술 엣지 컴퓨팅은 [클라우드 컴퓨팅](#)에 비해 보다 통신 지연 시간을 줄이고 응답 시간을 개선할 수 있습니다.

전자 데이터 교환(EDI)

조직 간 비즈니스 문서의 자동화된 교환을 나타냅니다. 자세한 내용은 [전자 데이터 교환\(EDI\)이란 무엇인가요?](#)를 참조하세요.

암호화

사람이 읽을 수 있는 일반 텍스트 데이터를 사이퍼텍스트로 변환하는 컴퓨팅 프로세스입니다.

암호화 키

암호화 알고리즘에 의해 생성되는 무작위 비트의 암호화 문자열입니다. 키의 길이는 다양할 수 있으며 각 키는 예측할 수 없고 고유하게 설계되었습니다.

엔디안

컴퓨터 메모리에 바이트가 저장되는 순서입니다. 빅 엔디안 시스템은 가장 중요한 바이트를 먼저 저장합니다. 리틀 엔디안 시스템은 가장 덜 중요한 바이트를 먼저 저장합니다.

엔드포인트

[서비스 엔드포인트](#)를 참조하세요.

엔드포인트 서비스

Virtual Private Cloud(VPC)에서 호스팅하여 다른 사용자와 공유할 수 있는 서비스입니다. 를 사용하여 엔드포인트 서비스를 생성하고 다른 AWS 계정 또는 AWS Identity and Access Management (IAM) 보안 주체에 권한을 AWS PrivateLink 부여할 수 있습니다. 이러한 계정 또는 보안 주체는 인터페이스 VPC 엔드포인트를 생성하여 엔드포인트 서비스에 비공개로 연결할 수 있습니다. 자세한 내용은 Amazon Virtual Private Cloud(VPC) 설명서의 [엔드포인트 서비스 생성](#)을 참조하십시오.

엔터프라이즈 리소스 계획(ERP)

엔터프라이즈의 주요 비즈니스 프로세스(예: 회계, [MES](#), 프로젝트 관리)를 자동화하고 관리하는 시스템입니다.

봉투 암호화

암호화 키를 다른 암호화 키로 암호화하는 프로세스입니다. 자세한 내용은 AWS Key Management Service (AWS KMS) 설명서의 [봉투 암호화](#)를 참조하세요.

환경

실행 중인 애플리케이션의 인스턴스입니다. 다음은 클라우드 컴퓨팅의 일반적인 환경 유형입니다.

- 개발 환경 - 애플리케이션 유지 관리를 담당하는 핵심 팀만 사용할 수 있는 실행 중인 애플리케이션의 인스턴스입니다. 개발 환경은 변경 사항을 상위 환경으로 승격하기 전에 테스트하는 데 사용됩니다. 이러한 유형의 환경을 테스트 환경이라고도 합니다.
- 하위 환경 - 초기 빌드 및 테스트에 사용되는 환경을 비롯한 애플리케이션의 모든 개발 환경입니다.
- 프로덕션 환경 - 최종 사용자가 액세스할 수 있는 실행 중인 애플리케이션의 인스턴스입니다. CI/CD 파이프라인에서 프로덕션 환경이 마지막 배포 환경입니다.
- 상위 환경 - 핵심 개발 팀 이외의 사용자가 액세스할 수 있는 모든 환경입니다. 프로덕션 환경, 프로덕션 이전 환경 및 사용자 수용 테스트를 위한 환경이 여기에 포함될 수 있습니다.

에픽

애자일 방법론에서 작업을 구성하고 우선순위를 정하는 데 도움이 되는 기능적 범주입니다. 에픽은 요구 사항 및 구현 작업에 대한 개괄적인 설명을 제공합니다. 예를 들어, AWS CAF 보안 에픽에는 ID 및 액세스 관리, 탐지 제어, 인프라 보안, 데이터 보호 및 인시던트 대응이 포함됩니다. AWS 마이그레이션 전략의 에픽에 대한 자세한 내용은 [프로그램 구현 가이드](#)를 참조하십시오.

ERP

[엔터프라이즈 리소스 계획](#)을 참조하세요.

탐색 데이터 분석(EDA)

데이터 세트를 분석하여 주요 특성을 파악하는 프로세스입니다. 데이터를 수집 또는 집계한 다음 초기 조사를 수행하여 패턴을 찾고, 이상을 탐지하고, 가정을 확인합니다. EDA는 요약 통계를 계산하고 데이터 시각화를 생성하여 수행됩니다.

F

팩트 테이블

[스타 스키마](#)의 중앙 테이블입니다. 비즈니스 운영에 대한 정량적 데이터를 저장합니다. 일반적으로 팩트 테이블은 측정값이 있는 열 및 차원 테이블에 대한 외래 키가 있는 열과 같이 두 가지 열 유형을 포함합니다.

빠른 실패

개발 수명 주기를 줄이기 위해 빈번한 증분 테스트를 사용하는 철학입니다. 애자일 접근 방식의 핵심입니다.

장애 격리 경계

에서 장애의 영향을 제한하고 워크로드의 복원력을 개선하는 데 도움이 되는 가용 영역, AWS 리전 컨트롤 플레인 또는 데이터 플레인과 같은 AWS 클라우드경계입니다. 자세한 내용은 [AWS 장애 격리 경계](#)를 참조하세요.

기능 브랜치

[브랜치](#)를 참조하세요.

기능

예측에 사용하는 입력 데이터입니다. 예를 들어, 제조 환경에서 기능은 제조 라인에서 주기적으로 캡처되는 이미지일 수 있습니다.

기능 중요도

모델의 예측에 특성이 얼마나 중요한지를 나타냅니다. 이는 일반적으로 SHAP(Shapley Additive Descriptions) 및 통합 그래디언트와 같은 다양한 기법을 통해 계산할 수 있는 수치 점수로 표현됩니다. 자세한 내용은 [기계 학습 모델 해석 가능성을 참조하세요 AWS](#).

기능 변환

추가 소스로 데이터를 보강하거나, 값을 조정하거나, 단일 데이터 필드에서 여러 정보 세트를 추출하는 등 ML 프로세스를 위해 데이터를 최적화하는 것입니다. 이를 통해 ML 모델이 데이터를 활용할 수 있습니다. 예를 들어, 날짜 '2021-05-27 00:15:37'을 '2021년', '5월', '목', '15일'로 분류하면 학습 알고리즘이 다양한 데이터 구성 요소와 관련된 미묘한 패턴을 학습하는 데 도움이 됩니다.

퓨샷 프롬프팅

유사한 태스크를 수행하도록 요청하기 전에 [LLM](#)에 태스크와 원하는 출력을 보여주는 몇 가지 예제를 제공합니다. 이 기법은 모델이 프롬프트에 포함된 예제(샷)에서 학습하는 컨텍스트 내 학습을 적용합니다. 퓨샷 프롬프팅은 특정 형식 지정, 추론 또는 분야별 지식이 필요한 태스크에 효과적일 수 있습니다. [제로샷 프롬프팅](#)도 참조하세요.

FGAC

[세분화된 액세스 제어](#)를 참조하세요.

세분화된 액세스 제어(FGAC)

여러 조건을 사용하여 액세스 요청을 허용하거나 거부합니다.

플래시컷 마이그레이션

단계적 접근 방식을 사용하는 대신 [변경 데이터 캡처](#)를 통해 지속적 데이터 복제를 사용하여 최단 시간에 데이터를 마이그레이션하는 데이터베이스 마이그레이션 방법입니다. 목표는 가동 중지 시간을 최소화하는 것입니다.

FM

[파운데이션 모델](#)을 참조하세요.

파운데이션 모델(FM)

일반화되고 레이블이 지정되지 않은 데이터의 대규모 데이터세트에서 훈련된 대규모 딥 러닝 신경망입니다. FM은 언어 이해, 텍스트 및 이미지 생성, 자연어 대화와 같은 다양한 일반 태스크를 수행할 수 있습니다. 자세한 내용은 [파운데이션 모델이란?](#)을 참조하세요.

FM 게이트웨이

[파운데이션 모델](#)에 대한 액세스를 제어하고 정규화하는 중앙 집중식 중개자입니다. LLM 게이트웨이이라고도 합니다.

G

생성형 AI

대량의 데이터에서 훈련되었으며 간단한 텍스트 프롬프트를 사용하여 이미지, 비디오, 텍스트, 오디오와 같은 새 콘텐츠와 아티팩트를 생성할 수 있는 [AI](#) 모델의 하위 세트입니다. 자세한 내용은 [생성형 AI란 무엇인가요?](#)를 참조하세요.

지리적 차단

[지리적 제한](#)을 참조하세요.

지리적 제한(지리적 차단)

Amazon CloudFront에서 특정 국가의 사용자가 콘텐츠 배포에 액세스하지 못하도록 하는 옵션입니다. 허용 목록 또는 차단 목록을 사용하여 승인된 국가와 차단된 국가를 지정할 수 있습니다. 자세한 내용은 CloudFront 설명서의 [콘텐츠의 지리적 배포 제한](#)을 참조하십시오.

Gitflow 워크플로

하위 환경과 상위 환경이 소스 코드 리포지토리의 서로 다른 브랜치를 사용하는 방식입니다. Gitflow 워크플로는 레거시로 간주되며 [트렁크 기반 워크플로](#)는 선호되는 현대적 접근 방식입니다.

골든 이미지

시스템 또는 소프트웨어의 새 인스턴스를 배포하기 위한 템플릿으로 사용되는 해당 시스템 또는 소프트웨어의 스냅샷입니다. 예를 들어 제조 분야에서는 골든 이미지를 사용하여 여러 디바이스에서 소프트웨어를 프로비저닝할 수 있으며 이를 통해 딥이스 제조 작업의 속도, 확장성 및 생산성을 개선할 수 있습니다.

브라운필드 전략

새로운 환경에서 기존 인프라의 부재 시스템 아키텍처에 대한 그린필드 전략을 채택할 때 [브라운필드](#)라고도 하는 기존 인프라와의 호환성 제한 없이 모든 새로운 기술을 선택할 수 있습니다. 기존 인프라를 확장하는 경우 브라운필드 전략과 그린필드 전략을 혼합할 수 있습니다.

가드레일

조직 단위(OU) 전체에서 리소스, 정책 및 규정 준수를 관리하는 데 도움이 되는 중요 규칙입니다. 예방 가드레일은 규정 준수 표준에 부합하도록 정책을 시행하며, 서비스 제어 정책과 IAM 권한 경계를 사용하여 구현됩니다. 탐지 가드레일은 정책 위반 및 규정 준수 문제를 감지하고 해결을 위한 알림을 생성하며, 이는 AWS Config Amazon GuardDuty AWS Security Hub CSPM, , AWS Trusted Advisor Amazon Inspector 및 사용자 지정 AWS Lambda 검사를 사용하여 구현됩니다.

가드레일(AI)

책임감 있고 안전한 AI 동작을 보장하기 위해 [에이전트](#) 입력 및 출력을 필터링, 검증 및 제약하는 안전 메커니즘입니다.

H

HA

[고가용성](#)을 참조하세요.

이기종 데이터베이스 마이그레이션

다른 데이터베이스 엔진을 사용하는 대상 데이터베이스로 소스 데이터베이스 마이그레이션(예: Oracle에서 Amazon Aurora로) 이기종 마이그레이션은 일반적으로 리아키텍트 작업의 일부이며 스키마를 변환하는 것은 복잡한 작업일 수 있습니다. AWS 는 스키마 변환에 도움이 되는 [AWS SCT를](#) [제공](#)합니다.

높은 가용성(HA)

문제나 재해 발생 시 개입 없이 지속적으로 운영할 수 있는 워크로드의 능력. HA 시스템은 자동으로 장애 조치되고, 지속적으로 고품질 성능을 제공하고, 성능에 미치는 영향을 최소화하면서 다양한 부하와 장애를 처리하도록 설계되었습니다.

히스토리언 현대화

제조 산업의 요구 사항을 더 잘 충족하도록 운영 기술(OT) 시스템을 현대화하고 업그레이드하는 데 사용되는 접근 방식입니다. 히스토리언은 공장의 다양한 출처에서 데이터를 수집하고 저장하는 데 사용되는 일종의 데이터베이스입니다.

홀드아웃 데이터

[기계 학습](#) 모델을 훈련하는 데 사용되는 데이터세트에서 보류되는 레이블이 지정된 기록 데이터의 일부입니다. 홀드아웃 데이터를 사용하여 모델 예측을 홀드아웃 데이터와 비교해 모델 성능을 평가할 수 있습니다.

human-in-the-loop(HitL)

중요한 결정 시점에서 인적 검토 및 승인을 위해 [에이전트](#) 실행이 일시 중지되는 워크플로 패턴입니다.

동종 데이터베이스 마이그레이션

동일한 데이터베이스 엔진을 공유하는 대상 데이터베이스로 소스 데이터베이스 마이그레이션(예: Microsoft SQL Server에서 Amazon RDS for SQL Server로) 동종 마이그레이션은 일반적으로 리호스팅 또는 리플랫폼 작업의 일부입니다. 네이티브 데이터베이스 유틸리티를 사용하여 스키마를 마이그레이션할 수 있습니다.

핫 데이터

자주 액세스하는 데이터(예: 실시간 데이터 또는 최근 번역 데이터). 일반적으로 이 데이터에는 빠른 쿼리 응답을 제공하기 위한 고성능 스토리지 계층 또는 클래스가 필요합니다.

핫픽스

프로덕션 환경의 중요한 문제를 해결하기 위한 긴급 수정입니다. 핫픽스는 긴급하기 때문에 일반적인 DevOps 릴리스 워크플로 외부에서 실행됩니다.

하이퍼케어 기간

전환 직후 마이그레이션 팀이 문제를 해결하기 위해 클라우드에서 마이그레이션된 애플리케이션을 관리하고 모니터링하는 기간입니다. 일반적으로 이 기간은 1~4일입니다. 하이퍼케어 기간이 끝나면 마이그레이션 팀은 일반적으로 애플리케이션에 대한 책임을 클라우드 운영 팀에 넘깁니다.

I

IaC

[코드형 인프라](#)를 참조하세요.

자격 증명 기반 정책

AWS 클라우드 환경 내에서 권한을 정의하는 하나 이상의 IAM 보안 주체에 연결된 정책입니다.

유휴 애플리케이션

90일 동안 평균 CPU 및 메모리 사용량이 5~20%인 애플리케이션입니다. 마이그레이션 프로젝트에서는 이러한 애플리케이션을 사용 중지하거나 온프레미스에 유지하는 것이 일반적입니다.

IIoT

[산업용 사물 인터넷](#)을 참조하세요.

변경 불가능한 인프라

기존 인프라를 업데이트, 패치 또는 수정하는 대신 프로덕션 워크로드에 대한 새 인프라를 배포하는 모델입니다. 변경 불가능한 인프라는 [변경 가능한 인프라](#)보다 본질적으로 더 일관되고 안정적이며 예측 가능합니다. 자세한 내용은 AWS Well-Architected Framework의 [변경 불가능한 인프라를 사용하여 배포](#) 모범 사례를 참조하세요.

인바운드(수신) VPC

AWS 다중 계정 아키텍처에서 애플리케이션 외부에서 네트워크 연결을 수락, 검사 및 라우팅하는 VPC입니다. [AWS Security Reference Architecture](#)에서는 애플리케이션과 더 넓은 인터넷 간의 양방향 인터페이스를 보호하기 위해 인바운드, 아웃바운드 및 검사 VPC로 네트워크 계정을 설정할 것을 권장합니다.

증분 마이그레이션

한 번에 전체 전환을 수행하는 대신 애플리케이션을 조금씩 마이그레이션하는 전환 전략입니다. 예를 들어, 처음에는 소수의 마이크로서비스나 사용자만 새 시스템으로 이동할 수 있습니다. 모든 것이 제대로 작동하는지 확인한 후에는 레거시 시스템을 폐기할 수 있을 때까지 추가 마이크로서비스 또는 사용자를 점진적으로 이동할 수 있습니다. 이 전략을 사용하면 대규모 마이그레이션과 관련된 위험을 줄일 수 있습니다.

Industry 4.0

연결성, 실시간 데이터, 자동화, 분석 및 AI/ML의 발전을 통해 제조 프로세스의 현대화를 나타내기 위해 2016년에 [Klaus Schwab](#)에서 도입한 용어입니다.

인프라

애플리케이션의 환경 내에 포함된 모든 리소스와 자산입니다.

코드형 인프라(IaC)

구성 파일 세트를 통해 애플리케이션의 인프라를 프로비저닝하고 관리하는 프로세스입니다. IaC는 새로운 환경의 반복 가능성, 신뢰성 및 일관성을 위해 인프라 관리를 중앙 집중화하고, 리소스를 표준화하고, 빠르게 확장할 수 있도록 설계되었습니다.

산업용 사물 인터넷(IIoT)

제조, 에너지, 자동차, 의료, 생명과학, 농업 등의 산업 부문에서 인터넷에 연결된 센서 및 디바이스의 사용 자세한 내용은 [산업용 사물 인터넷\(IoT\) 디지털 트랜스포메이션 전략 구축](#)을 참조하십시오.

검사 VPC

AWS 다중 계정 아키텍처에서는 VPC(동일하거나 다른 AWS 리전), 인터넷 및 온프레미스 네트워크 간의 네트워크 트래픽 검사를 관리하는 중앙 집중식 VPCs입니다. [AWS Security Reference Architecture](#)에서는 애플리케이션과 더 넓은 인터넷 간의 양방향 인터페이스를 보호하기 위해 인바운드, 아웃바운드 및 검사 VPC로 네트워크 계정을 설정할 것을 권장합니다.

사물 인터넷(IoT)

인터넷이나 로컬 통신 네트워크를 통해 다른 디바이스 및 시스템과 통신하는 센서 또는 프로세서가 내장된 연결된 물리적 객체의 네트워크 자세한 내용은 [IoT란?](#)을 참조하십시오.

해석력

모델의 예측이 입력에 따라 어떻게 달라지는지를 사람이 이해할 수 있는 정도를 설명하는 기계 학습 모델의 특성입니다. 자세한 내용은 [기계 학습 모델 해석 가능성을 참조하세요 AWS](#).

IoT

[사물 인터넷](#)을 참조하세요.

IT 정보 라이브러리(ITIL)

IT 서비스를 제공하고 이러한 서비스를 비즈니스 요구 사항에 맞게 조정하기 위한 일련의 모범 사례 ITIL은 ITSM의 기반을 제공합니다.

IT 서비스 관리(ITSM)

조직의 IT 서비스 설계, 구현, 관리 및 지원과 관련된 활동 클라우드 운영을 ITSM 도구와 통합하는 방법에 대한 자세한 내용은 [운영 통합 가이드](#)를 참조하십시오.

ITIL

[IT 정보 라이브러리](#)를 참조하세요.

ITSM

[IT 서비스 관리](#)를 참조하세요.

L

레이블 기반 액세스 제어(LBAC)

사용자 및 데이터 자체에 각각 보안 레이블 값을 명시적으로 할당하는 필수 액세스 제어(MAC)를 구현한 것입니다. 사용자 보안 레이블과 데이터 보안 레이블 간의 교차 부분에 따라 사용자가 볼 수 있는 행과 열이 결정됩니다.

랜딩 존

랜딩 존은 확장 가능하고 안전한 잘 설계된 다중 계정 AWS 환경입니다. 조직은 여기에서부터 보안 및 인프라 환경에 대한 확신을 가지고 워크로드와 애플리케이션을 신속하게 시작하고 배포할 수 있습니다. 랜딩 존에 대한 자세한 내용은 [안전하고 확장 가능한 다중 계정 AWS 환경 설정](#)을 참조하십시오.

대규모 언어 모델(LLM)

방대한 양의 데이터에서 사전 훈련된 딥 러닝 [AI](#) 모델입니다. LLM은 질문에 대한 답변, 문서 요약, 텍스트를 다른 언어로 번역, 문장 완성과 같은 여러 태스크를 수행할 수 있습니다. 자세한 내용은 [대규모 언어 모델\(LLM\)이란 무엇인가요?](#)를 참조하세요.

대규모 마이그레이션

300대 이상의 서버 마이그레이션입니다.

LBAC

[레이블 기반 액세스 제어](#)를 참조하세요.

최소 권한

작업을 수행하는 데 필요한 최소 권한을 부여하는 보안 모범 사례입니다. 자세한 내용은 IAM 설명서의 [최소 권한 적용](#)을 참조하십시오.

리프트 앤드 시프트

[7R](#)을 참조하세요.

리틀 엔디안 시스템

가장 덜 중요한 바이트를 먼저 저장하는 시스템입니다. [엔디안](#)도 참조하세요.

LLM

[대규모 언어 모델](#)을 참조하세요.

하위 환경

[환경](#)을 참조하세요.

M

기계 학습(ML)

패턴 인식 및 학습에 알고리즘과 기법을 사용하는 인공 지능의 한 유형입니다. ML은 사물 인터넷 (IoT) 데이터와 같은 기록된 데이터를 분석하고 학습하여 패턴을 기반으로 통계 모델을 생성합니다. 자세한 내용은 [기계 학습](#)을 참조하십시오.

기본 브랜치

[브랜치](#)를 참조하세요.

맬웨어

컴퓨터 보안 또는 프라이버시를 위협하도록 설계된 소프트웨어입니다. 맬웨어는 컴퓨터 시스템을 방해하거나 민감한 정보를 유출하거나 무단 액세스 권한을 확보할 수 있습니다. 맬웨어의 예로 바이러스, 웜, 랜섬웨어, 트로이 목마, 스파이웨어, 키로거 등이 있습니다.

관리형 서비스

AWS 서비스는 인프라 계층, 운영 체제 및 플랫폼을 AWS 운영하며 사용자는 엔드포인트에 액세스하여 데이터를 저장하고 검색합니다. 관리형 서비스의 예로 Amazon Simple Storage Service(Amazon S3) 및 Amazon DynamoDB가 있습니다. 이를 추상화된 서비스라고도 합니다.

제조 실행 시스템(MES)

원자재를 생산 현장에서 완제품으로 변환하는 생산 프로세스를 추적, 모니터링, 문서화 및 제어하기 위한 소프트웨어 시스템입니다.

MAP

[Migration Acceleration Program](#)을 참조하세요.

MCP

[모델 컨텍스트 프로토콜](#)을 참조하세요.

Model Context Protocol(MCP)

[에이전트](#) 간??? 통신을 위한 상태 비저장 프로토콜입니다.

MCP 서버

[모델 컨텍스트 프로토콜](#)을 통해 하나 이상의 [도구](#)를 노출하는 서비스입니다.

메커니즘

도구를 생성하고 도구 채택을 유도한 다음 조정을 위해 결과를 검사하는 전체 프로세스입니다. 메커니즘은 작동 시 자체적으로 강화하고 개선하는 주기입니다. 자세한 내용은 AWS Well-Architected Framework의 [메커니즘 구축](#)을 참조하세요.

멤버 계정

조직의 일부인 관리 계정을 AWS 계정 제외한 모든 계정. AWS Organizations하나의 계정은 한 번에 하나의 조직 멤버만 될 수 있습니다.

MES

[제조 실행 시스템](#)을 참조하세요.

메시지 큐 원격 분석 전송(MQTT)

리소스 제약이 있는 [IoT](#) 디바이스에 대한 [게시 및 구독](#) 패턴을 기반으로 하는 경량 Machine-to-Machine(M2M) 통신 프로토콜입니다.

마이크로서비스

잘 정의된 API를 통해 통신하고 일반적으로 소규모 자체 팀이 소유하는 소규모 독립 서비스입니다. 예를 들어, 보험 시스템에는 영업, 마케팅 등의 비즈니스 역량이나 구매, 청구, 분석 등의 하위 영역에 매핑되는 마이크로 서비스가 포함될 수 있습니다. 마이크로서비스의 이점으로 민첩성, 유연한 확장, 손쉬운 배포, 재사용 가능한 코드, 복원력 등이 있습니다. 자세한 내용은 [AWS 서버리스 서비스를 사용하여 마이크로서비스 통합을 참조하세요](#).

마이크로서비스 아키텍처

각 애플리케이션 프로세스를 마이크로서비스로 실행하는 독립 구성 요소를 사용하여 애플리케이션을 구축하는 접근 방식입니다. 이러한 마이크로서비스는 경량 API를 사용하여 잘 정의된 인터페이스를 통해 통신합니다. 애플리케이션의 특정 기능에 대한 수요에 맞게 이 아키텍처의 각 마이크로 서비스를 업데이트, 배포 및 조정할 수 있습니다. 자세한 내용은 [에서 마이크로서비스 구현을 참조하세요 AWS](#).

Migration Acceleration Program(MAP)

조직이 클라우드로 전환하기 위한 강력한 운영 기반을 구축하고 초기 마이그레이션 비용을 상쇄하는 데 도움이 되는 컨설팅 지원, 교육 및 서비스를 제공하는 AWS 프로그램입니다. MAP에는 레거시 마이그레이션을 체계적인 방식으로 실행하기 위한 마이그레이션 방법론과 일반적인 마이그레이션 시나리오를 자동화하고 가속화하는 도구 세트가 포함되어 있습니다.

대규모 마이그레이션

애플리케이션 포트폴리오의 대다수를 웨이브를 통해 클라우드로 이동하는 프로세스로, 각 웨이브에서 더 많은 애플리케이션이 더 빠른 속도로 이동합니다. 이 단계에서는 이전 단계에서 배운 모범 사례와 교훈을 사용하여 팀, 도구 및 프로세스의 마이그레이션 팩토리를 구현하여 자동화 및 민첩한 제공을 통해 워크로드 마이그레이션을 간소화합니다. 이것은 [AWS 마이그레이션 전략](#)의 세 번째 단계입니다.

마이그레이션 팩토리

자동화되고 민첩한 접근 방식을 통해 워크로드 마이그레이션을 간소화하는 다기능 팀입니다. 마이그레이션 팩토리 팀에는 일반적으로 스프린트에서 일하는 운영, 비즈니스 분석가 및 소유자, 마이그레이션 엔지니어, 개발자, DevOps 전문가가 포함됩니다. 엔터프라이즈 애플리케이션 포트폴리오의 20~50%는 공장 접근 방식으로 최적화할 수 있는 반복되는 패턴으로 구성되어 있습니다. 자세한 내용은 이 콘텐츠 세트의 [클라우드 마이그레이션 팩토리 가이드](#)와 [마이그레이션 팩토리에 대한 설명](#)을 참조하십시오.

마이그레이션 메타데이터

마이그레이션을 완료하는 데 필요한 애플리케이션 및 서버에 대한 정보 각 마이그레이션 패턴에는 서로 다른 마이그레이션 메타데이터 세트가 필요합니다. 마이그레이션 메타데이터의 예로는 대상 서브넷, 보안 그룹 및 AWS 계정이 있습니다.

마이그레이션 패턴

사용되는 마이그레이션 전략, 마이그레이션 대상, 마이그레이션 애플리케이션 또는 서비스를 자세히 설명하는 반복 가능한 마이그레이션 작업입니다. 예: AWS Application Migration Service를 사용하여 Amazon EC2로 마이그레이션을 리호스팅합니다.

Migration Portfolio Assessment(MPA)

AWS 클라우드로 마이그레이션하는 비즈니스 사례를 검증하기 위한 정보를 제공하는 온라인 도구입니다. MPA는 상세한 포트폴리오 평가(서버 적정 규모 조정, 가격 책정, TCO 비교, 마이그레이션 비용 분석)와 마이그레이션 계획(애플리케이션 데이터 분석 및 데이터 수집, 애플리케이션 그룹화, 마이그레이션 우선순위 지정, 웨이브 계획)을 제공합니다. [MPA 도구](#)(로그인 필요)는 모든 AWS 컨설턴트와 APN 파트너 컨설턴트가 무료로 사용할 수 있습니다.

마이그레이션 준비 상태 평가(MRA)

AWS CAF를 사용하여 조직의 클라우드 준비 상태에 대한 인사이트를 얻고, 강점과 약점을 식별하고, 식별된 격차를 해소하기 위한 행동 계획을 수립하는 프로세스입니다. 자세한 내용은 [마이그레이션 준비 가이드](#)를 참조하십시오. MRA는 [AWS 마이그레이션 전략](#)의 첫 번째 단계입니다.

마이그레이션 전략

워크로드를 AWS 클라우드로 마이그레이션하는 데 사용되는 접근 방식입니다. 자세한 내용은 이 용어집의 [7R](#) 항목과 [조직을 동원하여 대규모 마이그레이션 가속화](#)를 참조하세요.

ML

[기계 학습](#)을 참조하세요.

현대화

비용을 절감하고 효율성을 높이고 혁신을 활용하기 위해 구식(레거시 또는 모놀리식) 애플리케이션과 해당 인프라를 클라우드의 민첩하고 탄력적이고 가용성이 높은 시스템으로 전환하는 것입니다. 자세한 내용은 [AWS 클라우드에서 애플리케이션을 현대화하기 위한 전략](#)을 참조하세요.

현대화 준비 상태 평가

조직 애플리케이션의 현대화 준비 상태를 파악하고, 이점, 위험 및 종속성을 식별하고, 조직이 해당 애플리케이션의 향후 상태를 얼마나 잘 지원할 수 있는지를 확인하는 데 도움이 되는 평가입니다. 평가 결과는 대상 아키텍처의 청사진, 현대화 프로세스의 개발 단계와 마일스톤을 자세히 설명하는 로드맵 및 파악된 격차를 해소하기 위한 실행 계획입니다. 자세한 내용은 [AWS 클라우드에서 애플리케이션의 현대화 준비 상태 평가](#)를 참조하세요.

모놀리식 애플리케이션(모놀리식 유형)

긴밀하게 연결된 프로세스를 사용하여 단일 서비스로 실행되는 애플리케이션입니다. 모놀리식 애플리케이션에는 몇 가지 단점이 있습니다. 한 애플리케이션 기능에 대한 수요가 급증하면 전체 아키텍처 규모를 조정해야 합니다. 코드 베이스가 커지면 모놀리식 애플리케이션의 기능을 추가하거나 개선하는 것도 더 복잡해집니다. 이러한 문제를 해결하기 위해 마이크로서비스 아키텍처를 사용할 수 있습니다. 자세한 내용은 [마이크로서비스로 모놀리식 유형 분해](#)를 참조하십시오.

MPA

[Migration Portfolio Assessment](#)를 참조하세요.

MQTT

[메시지 큐 원격 분석 전송](#)을 참조하세요.

멀티클래스 분류

여러 클래스에 대한 예측(2개 이상의 결과 중 하나 예측)을 생성하는 데 도움이 되는 프로세스입니다. 예를 들어, ML 모델이 '이 제품은 책인가요, 자동차인가요, 휴대폰인가요?' 또는 '이 고객이 가장 관심을 갖는 제품 범주는 무엇인가요?'라고 물을 수 있습니다.

변경 가능한 인프라

프로덕션 워크로드에 대한 기존 인프라를 업데이트하고 수정하는 모델입니다. 일관성, 신뢰성 및 예측 가능성을 높이기 위해 AWS Well-Architected Framework는 [변경 불가능한 인프라](#)를 모범 사례로 사용할 것을 권장합니다.

O

OAC

[오리진 액세스 제어](#)를 참조하세요.

OAI

[오리진 액세스 ID](#)를 참조하세요.

OCM

[조직 변경 관리](#)를 참조하세요.

오프라인 마이그레이션

마이그레이션 프로세스 중 소스 워크로드가 중단되는 마이그레이션 방법입니다. 이 방법은 가동 중지 증가를 수반하며 일반적으로 작고 중요하지 않은 워크로드에 사용됩니다.

OI

[운영 통합](#)을 참조하세요.

OLA

[운영 수준 계약](#)을 참조하세요.

온라인 마이그레이션

소스 워크로드를 오프라인 상태로 전환하지 않고 대상 시스템에 복사하는 마이그레이션 방법입니다. 워크로드에 연결된 애플리케이션은 마이그레이션 중에도 계속 작동할 수 있습니다. 이 방법은 가동 중지 차단 또는 최소화를 수반하며 일반적으로 중요한 프로덕션 워크로드에 사용됩니다.

OPC-UA

[Open Process Communications - Unified Architecture\(OPC-UA\)](#)를 참조하세요.

Open Process Communications - Unified Architecture(OPC-UA)

산업 자동화를 위한 Machine-to-Machine(M2M) 통신 프로토콜입니다. OPC-UA는 데이터 암호화, 인증 및 권한 부여 체계에 관한 상호 운용성 표준을 제공합니다.

운영 수준 협약(OLA)

서비스 수준에 관한 계약(SLA)을 지원하기 위해 직무 IT 그룹이 서로에게 제공하기로 약속한 내용을 명확히 하는 계약입니다.

운영 준비 상태 검토(ORR)

인시던트 및 잠재적 장애의 범위를 이해, 평가 또는 예방하거나 줄이는 데 도움이 되는 질문 체크리스트 및 관련 모범 사례입니다. 자세한 내용은 AWS Well-Architected Framework의 [운영 준비 상태 검토\(ORR\)](#)를 참조하세요.

운영 기술(OT)

물리적 환경에서 작동하여 산업 운영, 장비 및 인프라를 제어하는 하드웨어 및 소프트웨어 시스템입니다. 제조 분야에서 OT 및 정보 기술(IT) 시스템의 통합은 [Industry 4.0](#) 트랜스포메이션의 주요 중점 사항입니다.

운영 통합(OI)

클라우드에서 운영을 현대화하는 프로세스로 준비 계획, 자동화 및 통합을 수반합니다. 자세한 내용은 [운영 통합 가이드](#)를 참조하십시오.

조직 트레일

조직 AWS 계정 내 모든에 대한 모든 이벤트를 로깅 AWS CloudTrail 하는에서 생성된 추적입니다 AWS Organizations. 이 트레일은 조직에 속한 각 AWS 계정 에 생성되고 각 계정의 활동을 추적합니다. 자세한 내용은 CloudTrail 설명서의 [Creating a trail for an organization](#)을 참조하십시오.

조직 변경 관리(OCM)

사람, 문화 및 리더십 관점에서 중대하고 파괴적인 비즈니스 혁신을 관리하기 위한 프레임워크입니다. OCM은 변화 채택을 가속화하고, 과도기적 문제를 해결하고, 문화 및 조직적 변화를 주도함으로써 조직이 새로운 시스템 및 전략을 준비하고 전환할 수 있도록 지원합니다. AWS 마이그레이션 전략에서는 클라우드 채택 프로젝트에 필요한 변경 속도 때문에이 프레임워크를 인력 가속화라고 합니다. 자세한 내용은 [사용 가이드](#)를 참조하십시오.

오리진 액세스 제어(OAC)

CloudFront에서 Amazon Simple Storage Service(S3) 콘텐츠를 보호하기 위해 액세스를 제한하는 고급 옵션입니다. OAC는 AWS KMS (SSE-KMS)를 사용한 모든 서버 측 암호화 AWS 리전와 S3 버킷에 대한 동적 PUT 및 DELETE 요청에서 모든 S3 버킷을 지원합니다.

오리진 액세스 ID(OAI)

CloudFront에서 Amazon S3 콘텐츠를 보호하기 위해 액세스를 제한하는 옵션입니다. OAI를 사용하면 CloudFront는 Amazon S3가 인증할 수 있는 보안 주체를 생성합니다. 인증된 보안 주체는 특정 CloudFront 배포를 통해서만 S3 버킷의 콘텐츠에 액세스할 수 있습니다. 더 세분화되고 향상된 액세스 제어를 제공하는 [OAC](#)도 참조하십시오.

ORR

[운영 준비 상태 검토](#)를 참조하세요.

OT

[운영 기술](#)을 참조하세요.

아웃바운드(송신) VPC

AWS 다중 계정 아키텍처에서 애플리케이션 내에서 시작된 네트워크 연결을 처리하는 VPC입니다. [AWS Security Reference Architecture](#)에서는 애플리케이션과 더 넓은 인터넷 간의 양방향 인터페이스를 보호하기 위해 인바운드, 아웃바운드 및 검사 VPC로 네트워크 계정을 설정할 것을 권장합니다.

P

권한 경계

사용자나 역할이 가질 수 있는 최대 권한을 설정하기 위해 IAM 보안 주체에 연결되는 IAM 관리 정책입니다. 자세한 내용은 IAM 설명서의 [권한 경계](#)를 참조하십시오.

개인 식별 정보(PII)

직접 보거나 다른 관련 데이터와 함께 짝을 지을 때 개인의 신원을 합리적으로 추론하는 데 사용할 수 있는 정보입니다. PII의 예로는 이름, 주소, 연락처 정보 등이 있습니다.

PII

[개인 식별 정보](#)를 참조하세요.

플레이북

클라우드에서 핵심 운영 기능을 제공하는 등 마이그레이션과 관련된 작업을 캡처하는 일련의 사전 정의된 단계입니다. 플레이북은 스크립트, 자동화된 런북 또는 현대화된 환경을 운영하는 데 필요한 프로세스나 단계 요약의 형태를 취할 수 있습니다.

PLC

[프로그래밍 가능 로직 컨트롤러](#)를 참조하세요.

PLM

[제품 수명 주기 관리](#)를 참조하세요.

정책

권한 정의([ID 기반 정책](#) 참조), 액세스 조건 지정([리소스 기반 정책](#) 참조), AWS Organizations 내 조직의 모든 계정에 대한 최대 권한 정의([서비스 제어 정책](#) 참조)와 같은 작업을 수행할 수 있는 객체입니다.

다국어 지속성

데이터 액세스 패턴 및 기타 요구 사항을 기반으로 독립적으로 마이크로서비스의 데이터 스토리지 기술 선택. 마이크로서비스가 동일한 데이터 스토리지 기술을 사용하는 경우 구현 문제가 발생하거나 성능이 저하될 수 있습니다. 요구 사항에 가장 적합한 데이터 저장소를 사용하면 마이크로서비스를 더 쉽게 구현하고 성능과 확장성을 높일 수 있습니다.

포트폴리오 평가

마이그레이션을 계획하기 위해 애플리케이션 포트폴리오를 검색 및 분석하고 우선순위를 정하는 프로세스입니다. 자세한 내용은 [마이그레이션 준비 상태 평가](#)를 참조하십시오.

조건자

보통 WHERE 절에 있는 true 또는 false를 반환하는 쿼리 조건입니다.

푸시다운 조건자

전송 전에 쿼리의 데이터를 필터링하는 데이터베이스 쿼리 최적화 기법입니다. 이렇게 하면 관계형 데이터베이스에서 검색하고 처리해야 하는 데이터의 양이 줄고 쿼리 성능이 향상됩니다.

예방적 제어

이벤트 발생을 방지하도록 설계된 보안 제어입니다. 이 제어는 네트워크에 대한 무단 액세스나 원치 않는 변경을 방지하는 데 도움이 되는 1차 방어선입니다. 자세한 내용은 Implementing security controls on AWS의 [Preventative controls](#)를 참조하십시오.

보안 주체

작업을 수행하고 리소스에 액세스할 수 있는 AWS 있는의 엔터티입니다. 이 엔터티는 일반적으로 , AWS 계정 IAM 역할 또는 사용자의 루트 사용자입니다. 자세한 내용은 IAM 설명서의 [역할 용어 및 개념](#)의 보안 주체를 참조하십시오.

개인 정보 보호 중심 설계

전체 개발 프로세스에서 개인 정보를 고려하는 시스템 엔지니어링에서의 접근 방식입니다.

프라이빗 호스팅 영역

Amazon Route 53에서 하나 이상의 VPC 내 도메인과 하위 도메인에 대한 DNS 쿼리에 응답하는 방법에 대한 정보가 담긴 컨테이너입니다. 자세한 내용은 Route 53 설명서의 [프라이빗 호스팅 영역 작업](#)을 참조하십시오.

선제적 제어

규정 미준수 리소스의 배포를 방지하도록 설계된 [보안 제어](#)입니다. 이러한 제어는 리소스를 프로비저닝하기 전에 리소스를 스캔합니다. 리소스가 제어를 준수하지 않으면 프로비저닝되지 않습니다. 자세한 내용은 AWS Control Tower 설명서의 [제어 참조 가이드](#)를 참조하고 보안 [제어 구현의 사전 예방적 제어](#)를 참조하세요. AWS

제품 수명 주기 관리(PLM)

설계, 개발 및 출시부터 성장 및 성숙도를 거쳐 거부 및 제거에 이르기까지 전체 수명 주기 동안 제품의 데이터 및 프로세스 관리를 나타냅니다.

프로덕션 환경

[환경](#)을 참조하세요.

프로그래밍 가능 로직 컨트롤러(PLC)

제조 분야에서 기계를 모니터링하고 제조 프로세스를 자동화하는 매우 안정적이고 적응력이 뛰어난 컴퓨터입니다.

프롬프트 체이닝

한 [LLM](#) 프롬프트의 출력을 다음 프롬프트의 입력으로 사용하여 더 나은 응답을 생성합니다. 이 기법은 복잡한 태스크를 하위 태스크로 나누거나 예비 응답을 반복적으로 세부 조정하거나 확장하는데 사용됩니다. 이를 통해 모델 응답의 정확성과 관련성을 개선하고 보다 세분화되고 개인화된 결과를 얻을 수 있습니다.

가명화

데이터세트의 개인 식별자를 자리 표시자 값으로 바꾸는 프로세스입니다. 가명화는 개인 정보를 보호하는 데 도움이 될 수 있습니다. 가명화된 데이터는 여전히 개인 데이터로 간주됩니다.

게시/구독(pub/sub)

여러 마이크로서비스에서 비동기 통신을 지원하여 확장성과 응답성을 개선하는 패턴입니다. 예를 들어 마이크로서비스 기반 [MES](#)에서 마이크로서비스는 다른 마이크로서비스가 구독할 수 있는 채널에 이벤트 메시지를 게시할 수 있습니다. 시스템은 게시 서비스를 변경하지 않고도 새 마이크로 서비스를 추가할 수 있습니다.

Q

쿼리 계획

SQL 관계형 데이터베이스 시스템의 데이터에 액세스하는 데 사용되는 명령어와 같은 일련의 단계입니다.

쿼리 계획 회귀

데이터베이스 서비스 최적화 프로그램이 데이터베이스 환경을 변경하기 전보다 덜 최적의 계획을 선택하는 경우입니다. 통계, 제한 사항, 환경 설정, 쿼리 파라미터 바인딩 및 데이터베이스 엔진 업데이트의 변경으로 인해 발생할 수 있습니다.

R

RACI 매트릭스

[Responsible, Accountable, Consulted, Informed\(RACI\)](#)를 참조하세요.

RAG

[검색 증강 생성](#)을 참조하세요.

랜섬웨어

결제 완료될 때까지 컴퓨터 시스템이나 데이터에 대한 액세스를 차단하도록 설계된 악성 소프트웨어입니다.

RASCI 매트릭스

[Responsible, Accountable, Consulted, Informed\(RACI\)](#)를 참조하세요.

RCAC

[행 및 열 액세스 제어](#)를 참조하세요.

읽기 전용 복제본

읽기 전용 용도로 사용되는 데이터베이스의 사본입니다. 쿼리를 읽기 전용 복제본으로 라우팅하여 기본 데이터베이스의 로드를 줄일 수 있습니다.

리아키텍팅

[7R](#)을 참조하세요.

Recovery Point Objective(RPO)

마지막 데이터 복구 시점 이후 허용되는 최대 시간입니다. 이에 따라 마지막 복구 시점과 서비스 중단 사이에 허용되는 데이터 손실로 간주되는 범위가 결정됩니다.

Recovery Time Objective(RTO)

서비스 중단과 서비스 복원 사이의 허용 가능한 지연 시간입니다.

리팩터링

[7R](#)을 참조하세요.

리전

지리적 영역의 AWS 리소스 모음입니다. 각 AWS 리전은 내결함성, 안정성 및 복원력을 제공하기 위해 서로 격리되고 독립적입니다. 자세한 내용은 [계정에서 사용할 수 있는 AWS 리전 지정](#)을 참조하세요.

회귀

숫자 값을 예측하는 ML 기법입니다. 예를 들어, '이 집은 얼마에 팔릴까?'라는 문제를 풀기 위해 ML 모델은 선형 회귀 모델을 사용하여 주택에 대해 알려진 사실(예: 면적)을 기반으로 주택의 매매 가격을 예측할 수 있습니다.

리호스팅

[7R](#)을 참조하세요.

릴리스

배포 프로세스에서 변경 사항을 프로덕션 환경으로 승격시키는 행위입니다.

재배치

[7R](#)을 참조하세요.

리플랫폼

[7R](#)을 참조하세요.

재구매

[7R](#)을 참조하세요.

복원력

중단에 저항하거나 중단을 복구할 수 있는 애플리케이션의 기능입니다. [고가용성](#) 및 [재해 복구](#)는 AWS 클라우드에서 복원력을 계획할 때 일반적인 고려 사항입니다. 자세한 내용은 [AWS 클라우드 복원력](#)을 참조하세요.

리소스 기반 정책

Amazon S3 버킷, 엔드포인트, 암호화 키 등의 리소스에 연결된 정책입니다. 이 유형의 정책은 액세스가 허용된 보안 주체, 지원되는 작업 및 충족해야 하는 기타 조건을 지정합니다.

RACI(Responsible, Accountable, Consulted, Informed) 매트릭스

마이그레이션 활동 및 클라우드 운영에 참여하는 모든 당사자의 역할과 책임을 정의하는 매트릭스입니다. 매트릭스 이름은 매트릭스에 정의된 책임 유형에서 파생됩니다. 실무 담당자 (R), 의사 결정권자 (A), 업무 수행 조언자 (C), 결과 통보 대상자 (I). 지원자는 (S) 선택사항입니다. 지원자를 포함하면 매트릭스를 RASCI 매트릭스라고 하고, 지원자를 제외하면 RACI 매트릭스라고 합니다.

대응 제어

보안 기준에서 벗어나거나 부정적인 이벤트를 해결하도록 설계된 보안 제어입니다. 자세한 내용은 AWS에서 보안 제어 구현의 [대응 제어](#)를 참조하세요.

retain

[7R](#)을 참조하세요.

사용 중지

[7R](#)을 참조하세요.

검색 증강 세대(RAG)

응답을 생성하기 전에 [LLM](#)이 훈련 데이터 소스 외부에 있는 신뢰할 수 있는 데이터 소스를 참조하는 [생성형 AI](#) 기술입니다. 예를 들어 RAG 모델은 조직의 지식 기반 또는 사용자 지정 데이터에 대

한 시맨틱 검색을 수행할 수 있습니다. 자세한 내용은 [검색 증강 생성\(RAG\)이란 무엇인가요?](#)를 참조하세요.

교체

공격자가 자격 증명에 액세스하는 것을 더욱 어렵게 만들기 위해 [보안 암호](#)를 주기적으로 업데이트하는 프로세스입니다.

행 및 열 액세스 제어(RCAC)

액세스 규칙이 정의된 기본적인 유연한 SQL 표현식을 사용합니다. RCAC는 행 권한과 열 마스크로 구성됩니다.

RPO

[목표 복구 시점\(RPO\)](#)을 참조하세요.

RTO

[목표 복구 시간\(RTO\)](#)을 참조하세요.

런북

특정 작업을 수행하는 데 필요한 일련의 수동 또는 자동 절차입니다. 일반적으로 오류율이 높은 반복 작업이나 절차를 간소화하기 위해 런북을 만듭니다.

S

SAML 2.0

많은 ID 제공업체(idP)에서 사용하는 개방형 표준입니다. 이 기능을 사용하면 연동 SSO(Single Sign-On)를 AWS Management Console 사용할 수 있으므로 사용자는 조직의 모든 사용자에게 대해 IAM에서 사용자를 생성하지 않고도 로그인하거나 AWS API 작업을 호출할 수 있습니다. SAML 2.0 기반 페더레이션에 대한 자세한 내용은 IAM 설명서의 [SAML 2.0 기반 페더레이션 정보](#)를 참조하십시오.

SCADA

[감독 제어 및 데이터 획득](#)을 참조하세요.

SCP

[서비스 제어 정책](#)을 참조하세요.

보안 암호

에는 암호 또는 사용자 자격 증명과 같이 암호화된 형식으로 저장하는 AWS Secrets Manager가 밀 또는 제한된 정보가 있습니다. 보안 암호 값과 메타데이터로 구성됩니다. 보안 암호 값은 바이너리, 단일 문자열 또는 여러 문자열일 수 있습니다. 자세한 내용은 AWS Secrets Manager 설명서의 [Secrets Manager 보안 암호란 무엇인가요?](#)를 참조하세요.

보안 중심 설계

전체 개발 프로세스에서 보안을 고려하는 시스템 엔지니어링에서의 접근 방식입니다.

보안 제어

위협 행위자가 보안 취약성을 악용하는 능력을 방지, 탐지 또는 감소시키는 기술적 또는 관리적 가드레일입니다. 보안 제어는 [예방](#), [감지](#), [대응](#), [선제적](#)과 같은 기본적인 네 가지 보안 제어 유형으로 구분됩니다.

보안 강화

공격 표면을 줄여 공격에 대한 저항력을 높이는 프로세스입니다. 더 이상 필요하지 않은 리소스 제거, 최소 권한 부여의 보안 모범 사례 구현, 구성 파일의 불필요한 기능 비활성화 등의 작업이 여기에 포함될 수 있습니다.

보안 정보 및 이벤트 관리(SIEM) 시스템

보안 정보 관리(SIM)와 보안 이벤트 관리(SEM) 시스템을 결합하는 도구 및 서비스입니다. SIEM 시스템은 서버, 네트워크, 디바이스 및 기타 소스에서 데이터를 수집, 모니터링 및 분석하여 위협과 보안 침해를 탐지하고 알림을 생성합니다.

보안 응답 자동화

보안 이벤트에 자동으로 응답하거나 이를 해결하도록 설계된 사전 정의되고 프로그래밍된 작업입니다. 이러한 자동화는 보안 모범 사례를 구현하는 데 도움이 되는 [탐지](#) 또는 [대응](#) AWS 보안 제어 역할을 합니다. 자동화된 응답 작업의 예로 VPC 보안 그룹 수정, Amazon EC2 인스턴스 패치 적용 또는 자격 증명 교체 등이 있습니다.

서버 측 암호화

데이터를 AWS 서비스 수신하는가 대상에서 데이터를 암호화합니다.

서비스 제어 정책(SCP)

AWS Organizations에 속한 조직의 모든 계정에 대한 권한을 중앙 집중식으로 제어하는 정책입니다. SCP는 관리자가 사용자 또는 역할에 위임할 수 있는 작업에 대해 제한을 설정하거나 가드레일을 정의합니다. SCP를 허용 목록 또는 거부 목록으로 사용하여 허용하거나 금지할 서비스 또는 작

업을 지정할 수 있습니다. 자세한 내용은 AWS Organizations 설명서의 [서비스 제어 정책을](#) 참조하세요.

서비스 엔드포인트

에 대한 진입점의 URL입니다 AWS 서비스. 엔드포인트를 사용하여 대상 서비스에 프로그래밍 방식으로 연결할 수 있습니다. 자세한 내용은 AWS 일반 참조의 [AWS 서비스 엔드포인트](#)를 참조하십시오.

서비스 수준에 관한 계약(SLA)

IT 팀이 고객에게 제공하기로 약속한 내용(예: 서비스 가동 시간 및 성능)을 명시한 계약입니다.

서비스 수준 지표(SLI)

오류 발생률, 가용성 또는 처리량과 같은 서비스의 성능 측면에 대한 측정값입니다.

서비스 수준 목표(SLO)

[서비스 수준 지표](#)로 측정되는 서비스의 상태를 나타내는 목표 지표입니다.

공동 책임 모델

클라우드 보안 및 규정 준수를 AWS 위해와 공유하는 책임을 설명하는 모델입니다. AWS 는 클라우드의 보안을 책임지고, 사용자는 클라우드의 보안을 책임집니다. 자세한 내용은 [공동 책임 모델](#)을 참조하십시오.

새도우 AI

조직 내 관리형 채널 외부에서 구축되거나 사용되는 승인되지 않은 [AI](#) 애플리케이션입니다.

SIEM

[보안 정보 및 이벤트 관리 시스템](#)을 참조하세요.

단일 장애점(SPOF)

애플리케이션을 중단시킬 수 있는 애플리케이션의 중요한 단일 구성 요소에서 발생하는 장애입니다.

SLA

[서비스 수준 계약](#)을 참조하세요.

SLI

[서비스 수준 지표](#)를 참조하세요.

SLO

[서비스 수준 목표](#)를 참조하세요.

분할 앤 시드 모델

현대화 프로젝트를 확장하고 가속화하기 위한 패턴입니다. 새로운 기능과 제품 릴리스가 정의되면 핵심 팀이 분할되어 새로운 제품 팀이 만들어집니다. 이를 통해 조직의 역량과 서비스 규모를 조정하고, 개발자 생산성을 개선하고, 신속한 혁신을 지원할 수 있습니다. 자세한 내용은 [AWS 클라우드에서 애플리케이션을 현대화하기 위한 단계별 접근 방식](#)을 참조하세요.

SPOF

[단일 장애점](#)을 참조하세요.

스타 스키마

하나의 큰 팩트 테이블을 사용하여 트랜잭션 또는 측정된 데이터를 저장하고 하나 이상의 더 작은 차원 테이블을 사용하여 데이터 속성을 저장하는 데이터베이스 조직 구조입니다. 이 구조는 [데이터 웨어하우스](#)에서 또는 비즈니스 인텔리전스 목적으로 사용하도록 설계되었습니다.

Strangler Fig 패턴

레거시 시스템을 폐기할 수 있을 때까지 시스템 기능을 점진적으로 다시 작성하고 교체하여 모놀리식 시스템을 현대화하기 위한 접근 방식. 이 패턴은 무화과 덩굴이 나무로 자라 결국 숙주를 압도하고 대체하는 것과 비슷합니다. [Martin Fowler](#)가 모놀리식 시스템을 다시 작성할 때 위험을 관리하는 방법으로 이 패턴을 도입했습니다. 이 패턴을 적용하는 방법의 예는 [컨테이너 및 Amazon API Gateway를 사용하여 기존의 Microsoft ASP.NET\(ASMX\) 웹 서비스를 점진적으로 현대화하는 방법](#)을 참조하십시오.

서브넷

VPC의 IP 주소 범위입니다. 서브넷은 단일 가용 영역에 상주해야 합니다.

감독 제어 및 데이터 획득(SCADA)

제조 분야에서 하드웨어와 소프트웨어를 사용하여 물리적 자산과 프로덕션 작업을 모니터링하는 시스템입니다.

대칭 암호화

동일한 키를 사용하여 데이터를 암호화하고 복호화하는 암호화 알고리즘입니다.

합성 테스트

사용자 상호 작용을 시뮬레이션하여 잠재적 문제를 감지하거나 성능을 모니터링하는 방식으로 진행되는 시스템 테스트입니다. [Amazon CloudWatch Synthetics](#)를 사용하여 이러한 테스트를 생성할 수 있습니다.

시스템 프롬프트

[LLM](#)에 컨텍스트, 명령 또는 지침을 제공하여 동작을 지시하는 기법입니다. 시스템 프롬프트는 컨텍스트를 설정하고 사용자와의 상호 작용을 위한 규칙을 설정하는 데 도움이 됩니다.

T

tags

AWS 리소스를 구성하기 위한 메타데이터 역할을 하는 키-값 페어입니다. 태그를 사용하면 리소스를 손쉽게 관리, 식별, 정리, 검색, 필터링할 수 있습니다. 자세한 내용은 [AWS 리소스에 태그 지정](#)을 참조하십시오.

대상 변수

지도 ML에서 예측하려는 값으로, 결과 변수라고도 합니다. 예를 들어, 제조 설정에서 대상 변수는 제품 결함일 수 있습니다.

작업 목록

런북을 통해 진행 상황을 추적하는 데 사용되는 도구입니다. 작업 목록에는 런북의 개요와 완료해야 할 일반 작업 목록이 포함되어 있습니다. 각 일반 작업에 대한 예상 소요 시간, 소유자 및 진행 상황이 작업 목록에 포함됩니다.

테스트 환경

[환경](#)을 참조하세요.

훈련

ML 모델이 학습할 수 있는 데이터를 제공하는 것입니다. 훈련 데이터에는 정답이 포함되어야 합니다. 학습 알고리즘은 훈련 데이터에서 대상(예측하려는 답)에 입력 데이터 속성을 매핑하는 패턴을 찾고, 이러한 패턴을 캡처하는 ML 모델을 출력합니다. 그런 다음 ML 모델을 사용하여 대상을 모르는 새 데이터에 대한 예측을 할 수 있습니다.

tool

[에이전트](#)가 외부 시스템에서 작업을 수행하기 위해 호출할 수 있는 함수 또는 API입니다.

Transit Gateway

VPC와 온프레미스 네트워크를 상호 연결하는 데 사용할 수 있는 네트워크 전송 허브입니다. 자세한 내용은 AWS Transit Gateway 설명서의 [전송 게이트웨이란 무엇입니까?](#)를 참조하세요.

트렁크 기반 워크플로

개발자가 기능 브랜치에서 로컬로 기능을 구축하고 테스트한 다음 해당 변경 사항을 기본 브랜치에 병합하는 접근 방식입니다. 이후 기본 브랜치는 개발, 프로덕션 이전 및 프로덕션 환경에 순차적으로 구축됩니다.

신뢰할 수 있는 액세스

사용자를 대신하여 AWS Organizations 및 해당 계정에서 조직에서 작업을 수행하도록 지정하는 서비스에 대한 권한 부여. 신뢰할 수 있는 서비스는 필요할 때 각 계정에 서비스 연결 역할을 생성하여 관리 작업을 수행합니다. 자세한 내용은 설명서의 [다른 AWS 서비스와 AWS Organizations 함께 사용](#)을 참조하세요 AWS Organizations .

튜닝

ML 모델의 정확도를 높이기 위해 훈련 프로세스의 측면을 여러 변경하는 것입니다. 예를 들어, 레이블링 세트를 생성하고 레이블을 추가한 다음 다양한 설정에서 이러한 단계를 여러 번 반복하여 모델을 최적화하는 방식으로 ML 모델을 훈련할 수 있습니다.

피자 두 판 팀

피자 두 판이면 충분한 소규모 DevOps 팀. 피자 두 판 팀 규모는 소프트웨어 개발에 있어 가능한 최상의 공동 작업 기회를 보장합니다.

U

불확실성

예측 ML 모델의 신뢰성을 저해할 수 있는 부정확하거나 불완전하거나 알려지지 않은 정보를 나타내는 개념입니다. 불확실성에는 두 가지 유형이 있습니다. 인식론적 불확실성은 제한적이고 불완전한 데이터에 의해 발생하는 반면, 우연한 불확실성은 데이터에 내재된 노이즈와 무작위성에 의해 발생합니다.

차별화되지 않은 작업

애플리케이션을 만들고 운영하는 데 필요하지만 최종 사용자에게 직접적인 가치를 제공하거나 경쟁 우위를 제공하지 못하는 작업을 헤비 리프팅이라고도 합니다. 차별화되지 않은 작업의 예로는 조달, 유지보수, 용량 계획 등이 있습니다.

상위 환경

[환경](#)을 참조하세요.

V

정리

스토리지를 회수하고 성능을 향상시키기 위해 증분 업데이트 후 정리 작업을 수반하는 데이터베이스 유지 관리 작업입니다.

버전 제어

리포지토리의 소스 코드 변경과 같은 변경 사항을 추적하는 프로세스 및 도구입니다.

VPC 피어링

프라이빗 IP 주소를 사용하여 트래픽을 라우팅할 수 있게 하는 두 VPC 간의 연결입니다. 자세한 내용은 Amazon VPC 설명서의 [VPC 피어링이란?](#)을 참조하십시오.

취약성

시스템 보안을 손상시키는 소프트웨어 또는 하드웨어 결함입니다.

W

웜 캐시

자주 액세스하는 최신 관련 데이터를 포함하는 버퍼 캐시입니다. 버퍼 캐시에서 데이터베이스 인스턴스를 읽을 수 있기 때문에 주 메모리나 디스크에서 읽는 것보다 빠릅니다.

웜 데이터

자주 액세스하지 않는 데이터입니다. 이런 종류의 데이터를 쿼리할 때는 일반적으로 적절히 느린 쿼리가 허용됩니다.

창 함수

현재 레코드와 어떤 식으로든 관련된 행 그룹에서 계산을 수행하는 SQL 함수입니다. 창 함수는 이동 평균을 계산하거나 현재 행의 상대적 위치를 기반으로 행 값에 액세스하는 등의 태스크를 처리하는 데 유용합니다.

워크로드

고객 대면 애플리케이션이나 백엔드 프로세스 같이 비즈니스 가치를 창출하는 리소스 및 코드 모음입니다.

워크스트림

마이그레이션 프로젝트에서 특정 작업 세트를 담당하는 직무 그룹입니다. 각 워크스트림은 독립적이지만 프로젝트의 다른 워크스트림을 지원합니다. 예를 들어, 포트폴리오 워크스트림은 애플리케이션 우선순위 지정, 웨이브 계획, 마이그레이션 메타데이터 수집을 담당합니다. 포트폴리오 워크스트림은 이러한 자산을 마이그레이션 워크스트림에 전달하고, 마이그레이션 워크스트림은 서버와 애플리케이션을 마이그레이션합니다.

WORM

[Write Once, Read Many\(WORM\)](#)를 참조하세요.

WQF

[AWS Workload Qualification Framework](#)를 참조하세요.

Write Once Read Many(WORM)

데이터를 한 번 쓰고 데이터가 삭제되거나 수정되지 않도록 하는 스토리지 모델입니다. 권한 있는 사용자는 필요한 만큼 여러 번 데이터를 읽을 수 있지만 데이터를 변경할 수는 없습니다. 이 데이터 스토리지 인프라는 [변경 불가능](#)한 항목으로 간주됩니다.

Z

제로데이 익스플로잇

[제로데이 취약성](#)을 악용하는 공격(일반적으로 맬웨어)입니다.

제로데이 취약성

프로덕션 시스템의 명백한 결함 또는 취약성입니다. 위협 행위자는 이러한 유형의 취약성을 사용하여 시스템을 공격할 수 있습니다. 개발자는 공격의 결과로 취약성을 인지하는 경우가 많습니다.

제로샷 프롬프팅

태스크를 수행하기 위해 [LLM](#)에 명령을 제공하지만 안내에 도움이 되는 예제(샷)는 제공하지 않습니다. LLM은 사전 훈련된 지식을 사용하여 태스크를 처리해야 합니다. 제로샷 프롬프팅의 효과는 태스크의 복잡성과 프롬프트의 품질에 따라 달라집니다. [퓨샷 프롬프팅](#)도 참조하세요.

좀비 애플리케이션

평균 CPU 및 메모리 사용량이 5% 미만인 애플리케이션입니다. 마이그레이션 프로젝트에서는 이러한 애플리케이션을 사용 중지하는 것이 일반적입니다.

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