AWS 클라우드 채택 프레임워크: 플랫폼 관점

AWS 권장 가이드

AWS 권장 가이드: AWS 클라우드 채택 프레임워크: 플랫폼 관점

Table of Contents

AWS Cloud Adoption Framework: 플랫폼 관점

Amazon Web Services(기여자)

2023년 10월(문서 기록)

디지털 트랜스포메이션은 임원이 고객 경험, 혁신 및 유연성을 개선할 수 있는 가장 큰 단일 원동력입니다. 기계 학습(ML), 인공 지능(AI), 빅 데이터 그리고 클라우드의 속도와 규모를 사용하여 변화하는 비즈니스 조건과 진화하는 고객 요구 사항을 충족합니다.

Amazon Web Services(AWS)는 세계에서 가장 포괄적이고 광범위하게 채택된 클라우드 플랫폼입니다. 이를 통해 비즈니스 위험을 줄이고, 환경, 사회 및 거버넌스(ESG) 성과를 개선하며, 수익을 늘리고, 운영 효율성을 개선하면서 조직을 혁신할 수 있습니다.

AWS Cloud Adoption Framework(AWS CAF)는 AWS 모범 사례를 사용하여 비즈니스 성과를 가속화하는 데 도움이 됩니다. AWS CAF를 사용하여 혁신 기회를 식별하고 우선순위를 지정하며 클라우드 준비 상태를 평가 및 개선하고 혁신 로드맵을 반복적으로 발전시킵니다.

AWS CAF는 비즈니스, 인력, 거버넌스, 플랫폼, 보안 및 운영이라는 6가지 관점에서 지침을 그룹화합니다. 각 관점은 별도의 가이드에서 다룹니다. 이 가이드에서는 플랫폼 관점을 다루며, 엔터프라이즈급의 확장 가능한 하이브리드 클라우드 환경을 통해 클라우드 워크로드 전달을 가속화하는 데 중점을 둡니다.

소개

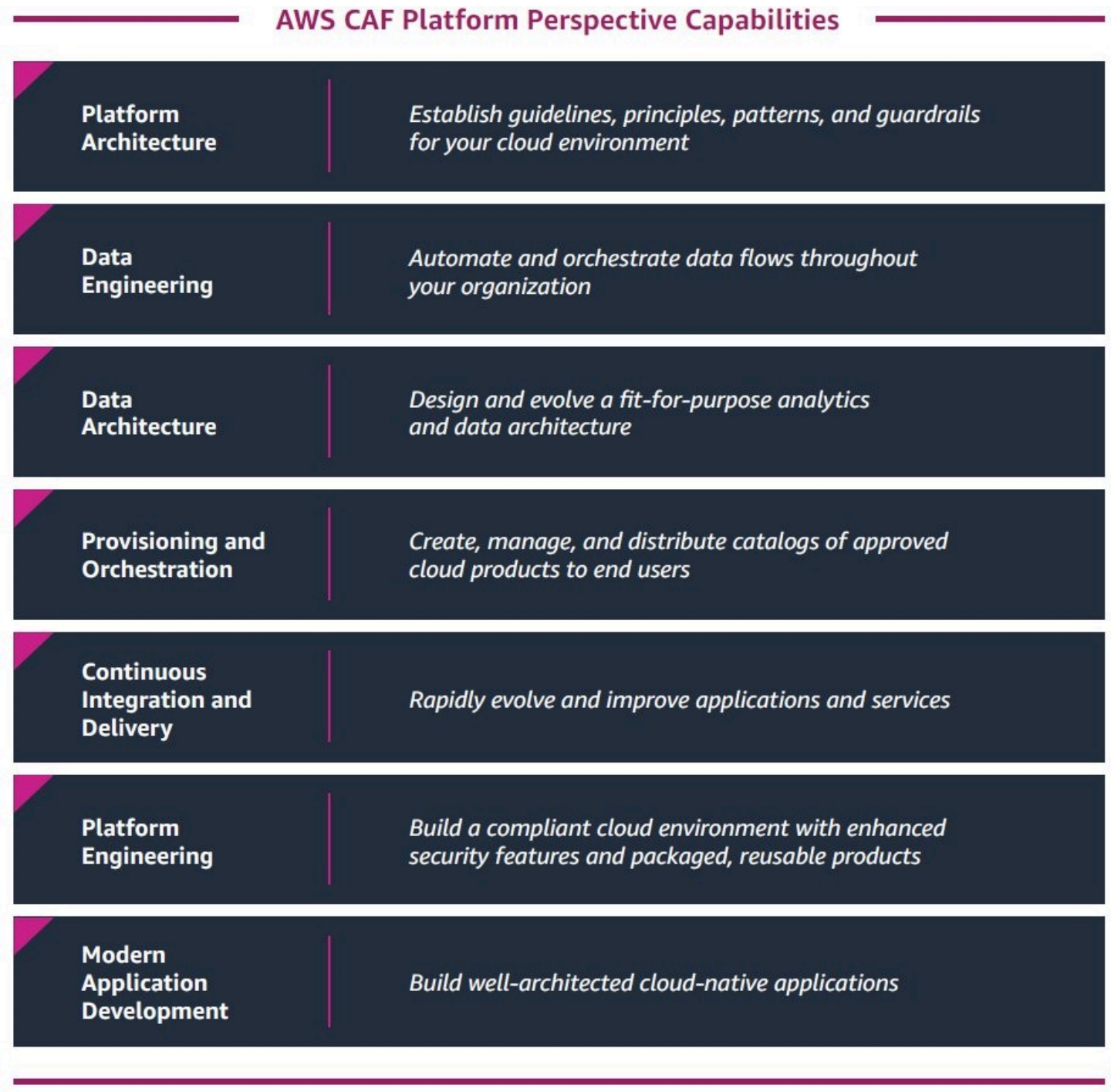
가장 빠르게 성장하는 스타트업, 대기업, 주요 정부 기관을 포함한 수백만 명의 고객이 AWS를 사용합니다. (AWS 웹 사이트의 Customer Success Stories를 참조하세요.) 레거시 워크로드를 마이그레이션 및 현대화하고, 데이터 기반 특성을 강화하며, 비즈니스 프로세스를 자동화 및 최적화하고, 운영 모델을 재창조할 수 있습니다. 비즈니스 위험을 줄이고, 환경, 사회 및 거버넌스(ESG) 성과를 개선하며, 수익을 높이고, 운영 효율성을 개선하여 비즈니스 성과를 개선할 수 있습니다.

클라우드를 효과적으로 사용하여 디지털 트랜스포메이션(조직 클라우드 준비)을 수행하는 조직의 역량은 일련의 기본 역량으로 강화됩니다. 역량이란, 프로세스를 사용하여 리소스(예: 인력, 기술 및 기타 유무형의 자산)를 배포하고 특정 성과를 달성하는 조직의 능력입니다. AWS CAF는 이러한 역량을 식별하고 전 세계 수천 개의 조직이 클라우드 준비 상태를 개선하고 클라우드 트랜스포메이션 여정을 가속화하는 데 성공적으로 사용해온 규범 가이드를 제공합니다.

AWS CAF는 다음과 같은 여섯 가지 측면에서 해당 기능을 그룹화합니다.

- 업무
- 사람
- 지배구조
- 플랫폼
- 보안.
- 운영

이 플랫폼 관점에서는 엔터프라이즈급의 확장 가능한 하이브리드 클라우드 환경을 통해 클라우드 워크로드 전달을 가속화하는 데 중점을 둡니다. 이 환경은 다음 다이어그램에 나온 7가지 역량으로 구성됩니다. 이러한 역량은 클라우드 트랜스포메이션 여정과 기능적으로 관련된 이해관계자가 관리합니다. 일반적인 이해관계자로, 최고 기술 책임자(CTO), 기술 리더, 아키텍트 및 엔지니어가 포함됩니다.



이 가이드의 다음 섹션에서 이러한 역량을 자세히 설명합니다. 각 섹션에서는 특정 역량을 시작하고 발전시키며 궁극적으로 우수한 수준으로 개발하는 방법에 대한 지침을 제공합니다.

- 플랫폼 아키텍처
- 플랫폼 엔지니어링
- 데이터 아키텍처

- 데이터 엔지니어링
- 프로비저닝 및 오케스트레이션
- 현대적 애플리케이션 개발
- 지속적 통합 및 지속적 전송(CI/CD)

플랫폼 관점은 AWS CAF의 핵심 부분입니다. 이는 비즈니스 민첩성과 가치를 제공하기 위해 다른 모든 관점에서 내린 의사 결정이 수렴되는 관계를 나타냅니다. 여기서 내린 결정은 기본적인 수준에서 비즈니스 목표에 도움이 되거나 방해가 될 수 있습니다. AWS CAF 플랫폼 관점은 조직의 트랜스포메이션을 뒷받침하는 확장 가능한 엔터프라이즈급 클라우드 환경을 조성하는 데 도움이 됩니다. 이러한 관점을 통해 AWS CAF는 클라우드 여정을 가능하게 하여 궁극적으로 상당한 비즈니스 트랜스포메이션과 성장으로 이어질 수 있는 강력한 플랫폼을 구축하는 데 도움이 됩니다.

플랫폼 관점을 살펴보면서 개발해야 하는 비즈니스 리더와의 부서 간 연결 관계와 이들이 팀과 조직에 가져오는 가치를 고려합니다. 요구 사항이 충족되도록 운영 모델 변경 및 팀 토폴로지에 추가로 중점을 둡니다. 또한 팀이 플랫폼을 빌드하고 애플리케이션 팀 전체에서 사용할 수 있도록 지원하는 데 필요한 기술을 개발합니다. 이러한 의사 결정을 내릴 때 조직의 사람, 비즈니스, 거버넌스, 보안 및 운영 목표를 염두에 둡니다. 그래야 플랫폼 채택과 노력의 성공을 보장할 수 있습니다.

AWS 및 AWS 파트너 네트워크는 이 여정에서 보안 태세를 구현하고 개선하는 데 도움이 될 수 있는 워크숍 및 교육과 같은 도구 및 서비스를 제공합니다. AWS Professional Services는 AWS CAF 맞춤 제품 모음을 통해 클라우드 트랜스포메이션과 관련된 특정 성과를 달성하는 데 도움을 줄 수 있는 글로벌 전문가 팀입니다.

플랫폼 아키텍처

클라우드 환경에 대한 지침, 원칙, 패턴 및 가드레일을 수립하고 유지 관리합니다.

잘 설계된 클라우드 환경은 구현을 가속화하고, 위험을 줄이며, 클라우드 채택을 촉진하는 데 도움이 됩니다. 플랫폼 아키텍처 역량은 조직 내에서 클라우드 채택을 주도하는 엔터프라이즈 표준에 대한 합의를 도출합니다. 인증, 보안, 네트워킹, 로깅 및 모니터링을 용이하게 하기 위해 모범 사례 블루프린트와 가드레일을 정의합니다. 또한 지연 시간, 데이터 처리 또는 데이터 레지던시 요구 사항으로 인해 온프레미스에 유지해야 할 수도 있는 워크로드를 고려하고 계획하며, 클라우드 버스팅, 클라우드로의 백업 및 재해 복구, 분산 데이터 처리, 엣지 컴퓨팅과 같은 하이브리드 클라우드 사용 사례를 평가합니다.

시작

다중 계정 전략 정의

좋은 다중 계정 전략에서는 규모와 운영 효율성 문제를 고려합니다. 즉, 운영 요구 사항에 가장 적합한 논리적 패턴으로 워크로드를 격리해야 합니다. 엔터프라이즈에서 중앙 집중식 및 분산형 서비스를 수용하려면 기본 계정 세트로 시작하는 것이 좋습니다. 보안, 재무 및 운영 기능을 중앙 집중화하여 분산 및 자율 팀과 계정을 효과적으로 관리하고 규제할 수 있습니다. 플랫폼과 워크로드를 분할하고 관리하는 방법을 이해하려면 조직 전체에서 조정해야 합니다. 이 구조를 이해하면 인증 및 권한 부여를 위한 보안 원칙을 마련하는 동시에 플랫폼에 허용되는 진화하는 사용 정책에 맞게 조정하는 데 도움이 됩니다.

예방적 제어 정의

기본 제어 세트(가드레일)가 포함된 안전한 다중 계정 환경을 계획합니다. 서비스 제어 정책(SCP)과 같은 메커니즘을 이해하고 사용하여 클라우드 플랫폼 내에서 소비할 수 있는 AWS 리전을 포함하여 조직 전체에서 서비스 사용을 관리합니다. 정책은 모든 계정에 사용할 수 있는 최대 권한을 제어하고 조직의 액세스 제어 지침을 준수하도록 보장하는 중앙 집중식 메커니즘을 제공합니다.

조직 단위 구조 정의

조직 단위(OU)는 규제 요구 사항 및 소프트웨어 개발 수명 주기(SDLC) 환경에 따라 계정을 관리하고 분류하는 실용적인 방법을 지원합니다. 조직은 OU를 사용하여 클라우드 인프라 전반에 걸쳐 적절한 정책 및 권한을 적용하는 프로세스를 간소화합니다. 워크로드 OU는 애플리케이션 인프라 리소스를 지원하는 계정을 위해 특별히 설계되었고, 올바른 정책이 적용되도록 보장합니다. OU 및 SCP를 사용하

면 조직의 클라우드 인프라에 대한 보안 및 규정 준수를 개선하는 동시에 애플리케이션 및 서비스의 원활한 운영을 보장할 수 있습니다. 그러면 궁극적으로 더 효율적이고 강력한 클라우드 채택 프로세스로 이어집니다.

하이브리드 네트워크 정의

네트워크 연결은 애플리케이션 및 워크로드를 지원하기 위해 안전하고 확장 가능하며 가용성이 높은 네트워크 생성을 지원하는 모든 클라우드 인프라의 중요한 측면입니다. 잘 설계된 네트워크는 일관되게 높은 성능을 제공하고 여러 환경에서 원활한 운영을 보장합니다.

네트워크 아키텍처를 설계하는 경우 지연 시간, 데이터 처리 또는 데이터 레지던시 요구 사항으로 인해 온프레미스에 유지하려는 워크로드가 있는지 고려합니다. 클라우드 버스팅, 클라우드로의 백업 및 재해 복구, 분산 데이터 처리, 엣지 컴퓨팅과 같은 하이브리드 클라우드 사용 사례를 평가하여 다음 측면의 주요 요구 사항을 식별할 수 있습니다.

- 인터넷과의 송수신 연결성. 이 측면에서는 애플리케이션 또는 워크로드와 인터넷 사이에서 안전하고 신뢰할 수 있는 연결을 제공해야 합니다. 이러한 연결성은 웹 기반 리소스에 대한 액세스를 용이하게 하고, 사용자와 애플리케이션 간 통신을 활성화하며, 필요할 때 서비스에 공개적으로 액세스할 수 있도록 하는 데 필수적입니다.
- 클라우드 환경 간 연결성. 이 영역은 클라우드 인프라 내 다양한 구성 요소와 서비스 사이에서 강력한 연결을 설정하는 데 중점을 둡니다. 이를 통해 여러 클라우드 서비스에서 데이터와 리소스를 쉽게 공유하고 액세스할 수 있으므로 효율적인 협업과 원활한 운영을 촉진할 수 있습니다. 여기서 주요 고려 사항은 가상 프라이빗 클라우드(VPC)의 사용입니다. 즉, VPC를 생성하고 추적하는 방법에 대한 표준을 생성하는 방법을 고려합니다. 이러한 표준을 프로그래밍 방식으로 생성하는 방법을 고려하고, IP Address Manager(IPAM) 솔루션을 사용하려고 합니다. 확장이 가능하도록 충분한 IP 공간을 할당하고 여러 가용 영역을 사용할 때 쉽게 문제를 해결할 수 있도록 서브넷 구조를 설계합니다. 네트워크 연결을 설계하고 구현할 때 VPC의 보안 모범 사례를 따라야 합니다.
- 온프레미스 네트워크와 클라우드 환경 간 연결성. 이 측면에서는 클라우드 기반 환경과 온프레미스 인프라의 통합을 다룹니다. 조직은 둘 사이에 안전하고 신뢰할 수 있는 연결을 생성하여 하이브리드 아키텍처의 이점을 활용할 수 있습니다. 예를 들어 성능, 확장성 및 비용 최적화 개선을 위해 온프레미스 리소스와 클라우드 서비스를 동시에 사용할 수 있습니다.

네트워크 연결성의 세 가지 주요 영역을 처리하면 애플리케이션과 워크로드를 효과적으로 지원하는 강력한 클라우드 인프라를 빌드하여 클라우드 채택의 이점을 활용할 수 있습니다. 네트워킹 요구 사항을 파악하고 다중 계정 전략에 따라 규모를 조정할 수 있는 간단한 설계를 생성합니다.

DNS 전략 정의

잘 계획된 DNS 전략은 클라우드 환경이 성장함에 따라 복잡성을 방지하는 데 도움이 됩니다. 온프레미스 DNS 기능을 유지 관리하는 경우 클라우드 기반 DNS 요구 사항을 이행하기 위해 클라우드 DNS와 함께 온프레미스 DNS 인프라를 사용하는 하이브리드 DNS 아키텍처를 설계하는 것이 좋습니다. 해석기 엔드포인트 및 전달 규칙을 사용하여 DNS 해석을 온프레미스 DNS 환경과 통합합니다. 프라이빗 호스팅 영역을 사용하여 클라우드 DNS가 하나 이상의 네트워크 내에서 도메인 및 하위 도메인에 대한 쿼리에 응답하는 방법에 대한 정보를 보유합니다.

태그 지정 표준 정의

리소스에 태그를 지정하는 것은 비용을 효과적으로 관리하고 리소스 소유권을 식별하기 위한 필수적인 관행입니다. 플랫폼 내 특정 서비스 사용을 포함하여 조직이 클라우드에서 소비를 추가로 허용하는 방법을 고려합니다. 어떤 팀이 어떤 리소스를 배포하고 있는지 추적하는 태그 지정 전략을 정의합니다. AWS CAF 운영 관점에서 입력을 받고 태그를 사용하여 배포된 인프라에 대한 태스크를 자동화합니다.

또한 관련 메타데이터로 리소스에 태그를 지정하면 AWS CAF 거버넌스 관점에서 클라우드 재무 관리(CFM) 기능에 명시된 조직 요구 사항에 따라 지출을 그룹화하고 추적할 수 있습니다. 재무 정책 위반 시 취해야 할 조치를 포함하여 회계 및 재무 사례를 지원하는 보고 메커니즘을 식별합니다.

관찰성 전략 정의

관찰성 전략의 수립은 클라우드 아키텍처를 최적화하고 보안하기 위한 중요한 단계입니다. 이 전략은 클라우드 서비스에서 생성된 지표와 로그를 전략적 의사 결정을 위한 실행 가능한 인사이트로 변환하는 작업을 중심으로 작동합니다. 핵심 성과 지표 모니터링의 우선순위를 정하고 잠재적 문제를 선제적으로 해결하기 위한 알림을 설정합니다. 도구 확산을 방지하고, 비용을 최적화하며, 조직에 가장 중요한 내용에 집중하려면 플랫폼과 애플리케이션 모두에 이 관찰성 전략을 통합합니다. 자세한 지침은 Developing an observability strategy(AWS re:Invent 2022)에 관한 프레젠테이션을 참조하세요.

고급

선제적 제어 및 감지 제어 정의

앞으로의 발전을 위해 조직은 환경 내에서 선제적 제어 및 감지 제어(가드레일)의 필요성을 식별해야 합니다. 조직 단위(OU)에 있는 계정에서 역할과 사용자가 보유한 가드레일 또는 제한을 정의하는 정책을 생성합니다. 플랫폼의 기본 감지 가드레일을 검토하고 적용할 가드레일을 선택합니다. 필요에 따라 추가적인 선제적 제어 및 감지 제어를 생성하고 OU별로 그룹화하여 다중 계정 전략에 맞게 조정합니

다. 감지 제어로 식별된 규정 미준수 리소스를 검사하는 데 필요한 조직 도구 및 메커니즘을 고려합니다.

서비스 온보딩을 위한 표준 정의

플랫폼의 허용 가능한 사용과 서비스 소비와 관련된 패턴 및 이를 관리하는 방법에 대한 표준을 생성합니다. 사용할 수 있는 초기 서비스를 고려합니다. 이러한 표준을 간략하게 설명하는 문서를 생성하고 플랫폼의 사용자와 운영자에게 게시합니다. 조직의 변화하는 목표와 클라우드 컴퓨팅의 진화하는 기능에 맞게 시간이 지남에 따라 이러한 표준을 조정해야 합니다.

패턴 및 원칙 정의

애플리케이션 소유자의 입력을 사용하여 조직 내에서 허용되는 아키텍처 패턴을 고려하고 표준화를 위한 블루프린트를 정의하기 시작합니다. 표준화를 통해 클라우드에서 규모를 조정할 때 거버넌스를 강화하고 관리 부담을 줄일 수 있습니다. 코드형 인프라(IaC)를 사용하는 패턴을 정의하고 변경 제어 프로세스 및 IT 서비스 관리(ITSM) 시스템에 통합된 서비스 카탈로그를 사용하여 단순화된 배포 모델을 계획합니다. 이러한 블루프린트를 사용하는 방법과 예외를 허용하는 상황을 정의합니다. 인증, 보안 모니터링 및 가드레일을 고려하여 이러한 예외와 거버넌스를 계획합니다.

Excel

문제 해결 패턴 정의

보안 및 규정 준수 프레임워크에 따라 문제를 해결할 수 있도록 감지 가드레일 조사 결과에 주석을 달고 우선순위를 지정하는 방법을 고려합니다. 자동화를 사용하여 예산 및 태그 지정 정책을 위반하는 리소스를 포함해 정책 범위를 벗어난 리소스의 프로비저닝을 감지할 계획을 세웁니다. 런북과 플레이북을 업데이트하는 동안 서비스 수준 목표를 설정하고 측정하는 데 필요한 역량을 식별합니다. 이러한 사례에 대한 정기적인 검토와 피드백 메커니즘을 설정하여 플랫폼 발전과 관련된 데이터를 캡처합니다. 적절히 런북과 플레이북을 생성하고 업데이트하는 메커니즘을 정의합니다.

정책 전달 및 세부 조정

모든 설명서에 대한 중앙 집중식 콘텐츠 관리 시스템을 생성하여 플랫폼의 사용자와 운영자에게 배포합니다. 정책 변경 사항에 대한 향후 고려를 위해 피드백을 캡처하기 위한 메커니즘을 생성합니다.

재무 관리 기능 이해

조직은 예산에 대한 투명하고 포괄적인 이해 기반을 유지할 때 역량을 발휘합니다. 이를 통해 정보에 입각한 의사 결정을 내리고, 리소스를 효율적으로 할당하며, 전략적 목표를 달성할 수 있습니다. 예산

을 명확하게 파악하면 정보에 입각한 의사 결정, 효과적인 리소스 할당, 비용 제어, 성능 측정, 책임 및 규정 준수 유지 관리를 용이하게 처리하여 조직이 뛰어난 성과를 거둘 수 있습니다. 결과적으로 더 효율적이고, 재정적으로 안정적이며, 성장하는 조직이 될 수 있습니다. 성공적인 태그 지정 전략을 갖추면 AWS Budgets에서 비용 필터를 사용하여 리소스 태그를 기준으로 비용을 필터링할 수 있습니다. 그러면 특정 프로젝트, 부서, 환경 또는 기타 기준에 맞게 조정된 예산을 생성하여 재무 관리 기능을 더욱 개선할 수 있습니다. 비용 할당 태그 및 AWS Cost Categories를 태그와 연결하여 비용을 보고할 때 재무 인사이트를 확보하고 투명성을 높일 수 있습니다.

플랫폼 엔지니어링

재사용 가능한 패키지형 클라우드 제품으로 안전하고 규정을 준수하는 다중 계정 클라우드 환경을 빌드합니다.

개발 팀을 뒷받침하여 혁신을 지원하려면 비즈니스 요구 사항에 맞게 신속하게 플랫폼이 적응해 나가야 합니다. (AWS CAF Business perspective를 참조하세요.) 제품 관리 요구 사항에 적응할 수 있을 만큼 유연하고, 보안 제약 조건을 준수할 만큼 견고하며, 운영 요구 사항을 충족할 수 있을 만큼 빠른 속도를 갖추려면 그렇게 해야 합니다. 이 프로세스를 수행하려면 향상된 보안 기능과 재사용 가능한 패키지형 클라우드 제품으로 규정을 준수하는 다중 계정 클라우드 환경을 빌드해야 합니다.

효과적인 클라우드 환경에서 팀은 새 계정을 쉽게 프로비저닝하는 동시에 해당 계정이 조직 정책을 준수하도록 보장할 수 있습니다. 엄선된 클라우드 제품 세트를 사용하면 모범 사례를 코드화하고, 거버넌스를 지원하며, 클라우드 배포의 속도와 일관성을 높일 수 있습니다. 모범 사례 블루프린트와 감지 및 예방 가드레일을 배포합니다. 클라우드 환경을 기존 환경과 통합하여 원하는 하이브리드 클라우드 사용 사례를 활성화합니다.

계정 프로비저닝 워크플로를 자동화하고 여러 계정을 사용하여 보안 및 거버넌스 목표를 지원합니다. 온프레미스 및 클라우드 환경과 서로 다른 클라우드 계정 간 연결을 설정합니다. 사용자가 기존 로그인 자격 증명을 사용하여 인증할 수 있도록 기존 ID 제공업체(idP)와 클라우드 환경 사이에서 페더레이션을 구현합니다. 로깅을 중앙 집중화하고, 교차 계정 보안 감사를 설정하며, 인바운드 및 아웃바운드 DNS 해석기를 생성하고, 계정 및 가드레일에 대한 대시보드 가시성을 확보합니다.

기업 표준 및 구성 관리에 맞춰 클라우드 서비스를 사용하고 있는지 평가하고 인증합니다. 엔터프라이즈 표준을 셀프 서비스 배포 가능 제품 및 소모성 서비스로 패키징하고 지속적으로 개선합니다. 코드형 인프라(IaC)를 활용하여 선언 방식으로 구성을 정의합니다. 개발자와 비즈니스 사용자에게 플랫폼을 전파하고 조직 전체에서 채택을 가속화하는 통합을 빌드할 수 있도록 지원 팀을 만듭니다.

다음 섹션에서 설명하는 태스크를 완료하려면 조직을 최신 플랫폼 엔지니어링으로 발전시키도록 역량과 팀을 구축해야 합니다. 기술 세부 정보는 Establishing your Cloud Foundation on AWS 백서를 참조하세요.

시작

랜딩 존 빌드 및 가드레일 배포

성숙한 플랫폼 엔지니어링으로의 여정을 시작할 때 먼저 플랫폼 아키텍처 기능에 정의된 대로 감지 및 예방 가드레일을 사용하여 랜딩 존을 배포해야 합니다. 가드레일을 사용하는 경우 애플리케이션 소유

자가 클라우드 리소스를 사용하므로 조직 표준을 위반하지 않습니다. 이 메커니즘에서는 계정 프로비저닝 워크플로를 자동화하여 보안 및 거버넌스 목표를 지원하는 여러 계정을 사용합니다.

인증 설정

AWS CAF 보안 관점에 명시된 표준에 따라 모든 환경, 시스템, 워크로드 및 프로세스에 ID 관리 및 액세스 제어를 구현합니다. 직원 ID의 경우 AWS Identity and Access Management (IAM) 사용자의 사용을 제한하고 대신 중앙 위치에서 ID를 관리할 수 있는 ID 제공업체에 의존합니다. 이렇게 하면 단일 위치에서 액세스를 생성, 관리 및 취소하므로 여러 애플리케이션과 서비스에 대한 액세스를 더 쉽게 관리할 수 있습니다. 기존 프로세스를 사용하여 AWS 환경을 포함하도록 액세스의 생성, 업데이트 및 제거를 관리합니다.

네트워크 배포

플랫폼 아키텍처 설계에 따라 중앙 집중식 네트워크 계정을 생성하여 환경과의 인바운드 및 아웃바운드 트래픽을 제어합니다. 온프레미스 네트워크와 AWS 환경, 인터넷 간 및 AWS 환경 간에 빠르게 프로비저닝되도록 네트워크를 설계하는 것이 좋습니다. 네트워크 관리를 중앙 집중화하면 예방 및 대응 제어를 사용하여 환경 전체에서 네트워크 및 연결을 격리하도록 네트워크 제어를 배포할 수 있습니다.

이벤트 및 로그 데이터의 수집, 집계 및 보호

Amazon CloudWatch 교차 계정 관찰성을 사용합니다. 연결된 계정에서 지표, 로그 및 추적을 검색, 시각화 및 분석하고 계정 경계를 제거합니다.

중앙 집중식 로그 제어 및 보안에 대한 특정 규정 준수 요구 사항이 조직에 있는 경우 전용 로그 아카이브 계정을 설정하는 방법을 고려합니다. 특별히 로그 데이터에 대해 암호화된 중앙 집중식 리포지토리를 제공합니다. 암호화 키를 정기적으로 교체하여 이 아카이브의 보안을 강화합니다.

필요에 따라 마스킹 기법을 사용하여 민감한 로그 데이터를 보호하기 위한 강력한 정책을 구현합니다. 규정 준수, 보안 및 감사 로그에 대해 로그 집계를 사용하고, 로그 구성에 대한 무단 변경을 방지하기 위해 엄격한 가드레일 및 ID 구문을 사용해야 합니다.

제어 설정

AWS CAF 보안 관점의 정의에 따라 비즈니스 요구 사항을 충족하는 기본 보안 기능을 배포합니다. 추가 예방 및 감지 제어를 배포하고 필요한 경우 모든 계정에 프로그래밍 방식으로 일관되게 프로비저닝합니다. 플랫폼 아키텍처 기능에 정의된 대로 감지 제어를 운영 도구에 통합합니다. 그러면 운영 메커니즘에서 규정 미준수 리소스를 검토할 수 있습니다.

클라우드 재무 관리 구현

AWS CAF 거버넌스 관점에 따라 비용 할당 태그와 조직의 태그 지정 전략을 클라우드 소비에 대한 재정적 책임과 일치시키는 AWS Cost Categories를 구현합니다. AWS Cost Categories를 사용하면에 게시된 AWS Cost Explorer 및 결제 데이터와 같은 도구를 사용하여 클라우드 요금을 내부 비용 센터로 청구하거나 표시할 수 있습니다AWS Cost and Usage Report.

고급

인프라 자동화 빌드

계속하기 전에 플랫폼 아키텍처에 맞춰 클라우드 서비스를 사용하고 있는지 평가하고 인증합니다. 그런 다음 엔터프라이즈 표준을 배포 가능한 제품 및 소모성 서비스로 패키징하고 지속적으로 개선하며 코드형 인프라(IaC)를 사용하여 선언 방식으로 구성을 정의합니다. 인프라 자동화는 역할 기반 액세스 제어(RBAC) 또는 속성 기반 액세스 제어(ABAC)를 사용하여 각 계정의 특정 서비스에 대한 액세스를 허용함으로써 소프트웨어 개발 주기를 모방합니다. API를 사용하여 새 계정을 신속하게 프로비저닝하고 서비스 및 인시던트 관리 기능에 맞게 조정하거나 셀프 서비스 기능을 개발하는 방법을 배포합니다. 규정 준수 및 네트워크 보안을 보장하기 위해 계정이 생성될 때 네트워크 통합 및 IP 할당을 자동화합니다. AWS에서 작동하도록 구성된 기본 커넥터를 사용하여 새 계정을 IT 서비스 관리(ITSM) 솔루션에 통합합니다. 플레이북과 런북을 적절하게 업데이트합니다.

중앙 집중식 관찰성 서비스 제공

효과적인 클라우드 관찰성을 달성하려면 플랫폼이 로컬 및 중앙 집중식 로그 데이터의 실시간 검색과 분석을 모두 지원해야 합니다. 운영 규모가 조정됨에 따라 로그, 지표 및 추적을 인덱싱, 시각화, 해석하는 플랫폼의 기능은 원시 데이터를 실행 가능한 인사이트로 전환하는 데 매우 중요합니다.

로그, 지표 및 추적을 상호 연관시켜 실행 가능한 결론을 추출하고 목표에 맞는 정보에 입각한 대응을 개발할 수 있습니다. 로그, 지표 또는 추적에서 식별된 보안 이벤트나 패턴에 대한 선제적 대응을 허용하는 규칙을 설정합니다. AWS 솔루션이 확장되면 모니터링 전략이 나란히 확장되어 관찰성 기능을 유지하고 개선해야 합니다.

시스템 관리 및 AMI 거버넌스 구현

Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 사용하는 조직에서 대규모로 인스턴스를 관리하려면 운영 도구가 필요합니다. 소프트웨어 자산 관리, 엔드포인트 감지 및 대응, 인벤토리 관리, 취약성 관리, 액세스 관리는 많은 조직의 기본적인 기능입니다. 이러한 기능은 인스턴스에 설치

된 소프트웨어 에이전트를 통해 제공되기도 합니다. 에이전트 및 기타 사용자 지정 구성을 Amazon Machine Image(AMI)로 패키징하고 이러한 AMI를 클라우드 플랫폼의 소비자가 사용할 수 있도록 제공합니다. 이러한 AMI의 사용을 규제하는 선제적 제어 및 감지 제어를 사용합니다. 특히 정기적으로 새 AMI를 소비하지 않는 변경 가능한 Amazon EC2 워크로드의 경우 AMI는 대규모로 장기 실행 EC2 인스턴스를 관리할 수 있는 도구가 포함되어야 합니다. 대규모로 AWS Systems Manager를 사용하여 에이전트 업그레이드를 자동화하고, 시스템 인벤토리를 수집하며, EC2 인스턴스에 원격으로 액세스하고, 운영 체제 취약성을 패치할 수 있습니다.

자격 증명 사용 관리

AWS CAF 보안 관점에 따라 역할 및 임시 자격 증명을 구현합니다. 도구를 사용하여 보안 암호를 저장하지 않고 사전 설치된 에이전트를 사용해 인스턴스 또는 온프레미스 시스템에 대한 원격 액세스를 관리합니다. 장기 자격 증명에 대한 의존을 줄이고 IaC 템플릿에서 하드 코딩된 자격 증명을 스캔합니다. 임시 자격 증명을 사용할 수 없는 경우 애플리케이션 토큰 및 데이터베이스 암호와 같은 프로그래밍 도구를 사용하여 자격 증명 교체 및 관리를 자동화합니다. IaC에서 최소 권한 원칙을 사용하여 사용자, 그룹 및 역할을 코드화하고 가드레일을 사용하여 자격 증명 계정의 수동 생성을 방지합니다.

보안 도구 설정

보안 모니터링 도구에서는 인프라, 애플리케이션 및 워크로드 전반의 세분화된 보안 모니터링을 지원하고 패턴 분석을 위한 집계된 보기를 제공해야 합니다. 다른 모든 보안 관리 도구와 마찬가지로 확장 탐지 및 대응(XDR) 도구를 확장하여 AWS CAF 보안 관점에 정의된 요구 사항에 AWS 따라에서 애플리케이션, 리소스 및 환경의 보안을 평가, 탐지, 대응 및 해결하는 기능을 제공해야 합니다.

Excel

자동화를 통해 ID 구문 소싱 및 분산

IaC 도구로 역할, 정책 및 템플릿과 같은 ID 구문을 코드화하고 버전을 관리합니다. 정책 검증 도구를 사용하여 보안 경고, 오류, 일반 경고, IAM 정책에 대한 제안된 변경 사항 및 기타 조사 결과를 확인합니다. 해당되는 경우 환경에 대한 임시 액세스를 자동화된 방식으로 제공하는 ID 구성을 배포 및 제거하고 콘솔을 사용하는 개인에 의한 배포는 금지합니다.

전체 환경에서 이상 패턴에 대한 감지 및 알림 추가

환경에 알려진 취약성이 있는지 선제적으로 평가하고 비정상적인 이벤트 및 활동 패턴에 대한 감지를 추가합니다. 조사 결과를 검토하고 플랫폼 아키텍처 팀에 추가적인 효율성과 혁신을 촉진하는 변경 사항을 추천합니다.

위협 분석 및 모델링

AWS CAF 보안 관점의 요구 사항에 따라 산업 및 보안 벤치마크에 대한 지속적인 모니터링과 측정을 구현합니다. 계측 접근 방식을 구현하는 경우 보안 관리 기능에 가장 적합한 이벤트 데이터 및 정보 유형을 결정합니다. 이 모니터링에는 서비스 사용량을 포함한 여러 공격 벡터가 포함됩니다. 보안 기반에는 여러 소스의 이벤트를 상호 연관시키는 기능을 포함하여 다중 계정 환경 전반에서 보안 로깅 및 분석을 위한 포괄적인 기능이 포함되어야 합니다. 특정 제어 및 가드레일을 사용하여 이 구성의 변경을 방지합니다.

지속적으로 권한 수집, 검토 및 구체화

ID 역할 및 권한에 대한 변경 사항을 기록하고 감지 가드레일이 예상되는 구성 상태와의 편차를 감지하면 알림을 구현합니다. 집계 및 패턴 식별 도구를 사용하여 중앙 집중식 이벤트 컬렉션을 검토하고 필요에 따라 권한을 세부 조정합니다.

플랫폼 지표 선택, 측정 및 지속적 개선

성공적인 플랫폼 작업을 지원하기 위해 포괄적인 지표를 설정하고 정기적으로 검토합니다. 조직의 목표 및 이해관계자 요구 사항에 부합하는지 확인합니다. 팀 지원 및 도구 채택 지표를 사용하여 플랫폼 성능 및 개선 지표를 모두 추적하고 패치, 백업, 규정 준수와 같은 운영 파라미터를 결합합니다.

효율적인 지표 관리를 위해 CloudWatch 교차 계정 관찰성을 사용합니다. 이 서비스는 데이터 집계 및 시각화를 간소화하여 정보에 입각한 의사 결정과 목표 개선을 지원합니다. 이러한 지표를 성공의 지표 및 변화의 동인으로 사용하여 지속적인 개선 환경을 조성합니다.

데이터 아키텍처

목적별 데이터 및 분석 아키텍처를 설계하고 발전시킵니다.

잘 설계된 데이터 및 분석 아키텍처는 실행 가능한 인사이트를 얻는 데 필수적입니다. 조직은 목적별 데이터 및 분석 아키텍처를 설계하고 발전시켜 복잡성, 비용 및 기술 부채를 줄이는 동시에 계속 증가하는 데이터 볼륨에서 소중한 인사이트를 얻을 수 있습니다. AWS CAF 원칙에 맞게 조정함으로써 비즈니스는 기존 플랫폼과 원활하게 통합되는 데이터 아키텍처를 생성할 수 있습니다. 이러한 조정을 통해 조직은 최신 데이터 처리 및 분석 기술이 제공하는 이점을 활용하여 수익을 창출할 수 있습니다.

데이터 및 분석 아키텍처는 데이터에서 가치를 도출하는 조직 역량의 블루프린트입니다. 그러면 조직이 새로운 비즈니스 인사이트를 얻는 데 도움이 되며 비즈니스 성장의 원동력이 됩니다. 비즈니스 요구 사항을 지원하기 위해 최신 데이터 아키텍처는 단기 및 장기 비즈니스 목표에 부합하고 조직의 문화적, 상황별 요구 사항에 고유해야 합니다. 오늘날 환경에서 데이터 및 분석 아키텍처의 성공적인 구현과 채택은 올바른 소비자에게 적시에 올바른 데이터를 지원하는 원칙을 기반으로 합니다.

이는 조직의 데이터 자산이 물리적 또는 논리적으로 모델링되는 방법, 데이터 보안 방법, 이러한 데이터 모델이 서로 상호 작용하여 비즈니스 문제를 해결하며 알려지지 않은 패턴을 도출하고 인사이트를 생성하는 방법을 계획하고 구성함으로써 달성됩니다.

시작

주요 기능 정의

현재 비즈니스 환경에서는 최신 데이터 분석 플랫폼이 데이터에서 가치를 도출하여 조직의 다양한 도메인을 지원하는 것이 중요합니다. 단일 데이터 아키텍처 접근 방식을 채택하는 대신 최신 데이터 아키텍처에는 특정 사용 사례에 맞게 최적화된 목적별 도구 세트와 패턴이 포함되어야 합니다. 아키텍처는 발전 가능해야 하며, 확장 가능한 데이터 레이크, 목적별 분석 서비스, 통합 데이터 액세스, 통합 거버넌스와 같은 기본 구성 요소를 포함해야 합니다.

데이터 존 구성

빠르고 쉬운 액세스를 위해 데이터를 구성하고 저장하는 방법은 데이터 아키텍처의 중요한 측면입니다. 데이터 레이크 내에 사용자 지정 데이터 존을 설정하면 됩니다. 데이터 존은 다음과 같이 분류됩니다.

- 이기종 소스에서 수집된 원시 데이터
- 각 도메인의 분석 요구 사항을 지원하기 위해 선별 및 변환된 데이터

- 보고 요구 사항에 대한 사용 사례 또는 제품 기반 데이터 마트
- 보안 및 규정 준수 제어와 함께 외부에 노출된 데이터

데이터의 민첩성 및 대중화를 위한 계획

분석 플랫폼의 효과는 데이터 프로비저닝 속도뿐만 아니라 소비를 위한 프로비저닝된 데이터의 대중화 속도에 따라 달라집니다. 데이터 프로비저닝 민첩성은 데이터 아키텍처가 사용 사례를 기반으로 실시간, 거의 실시간, 배치, 마이크로 배치 또는 하이브리드와 같은 다양한 방식을 통해 데이터를 조달하고 처리할 수 있는 역량으로 달성됩니다. 데이터 대중화는 데이터 관리자가 모니터링하는 데이터 공유 및 액세스 제어 워크플로를 정의하여 달성됩니다. 데이터 마켓플레이스의 구현은 데이터 대중화를 지원하는 기능 중 하나입니다.

보안 데이터 전달 정의

최신 데이터 아키텍처는 보안 측면에서 외부 세계로부터 보호된 요새와 같지만, 직무 정의에 따라 직원 또는 데이터 사용자에게 쉽게 액세스할 수 있으며 미국 건강 보험 양도 및 책임에 관한 법(HIPAA), 개인 식별 정보(PII), 일반 데이터 보호 규정(GDPR) 등과 같은 규정 준수 제한 사항을 준수합니다. 이는 역할 기반 액세스 제어(RBAC) 및 태그 기반 액세스 제어(TBAC) 방법을 이루어집니다. AWS에서 태그는 데이터에 대한 액세스를 제어하여 액세스 제어 관리를 간소화하는 데 사용됩니다. AWS CAF 보안 관점에 설명된 원칙에 따라 조정합니다.

비용 효율성을 위한 계획

기존 데이터 웨어하우스는 리소스 사용률이 높은 긴밀하게 결합된 컴퓨팅 및 스토리지를 제공합니다. 최신 아키텍처는 컴퓨팅과 스토리지를 분리하고 데이터 수명 주기를 기반으로 계층형 스토리지를 구현합니다. 예를 들어 AWS에서 Amazon Simple Storage Service(Amazon S3)를 사용하여 비용을 제어하고 컴퓨팅에서 데이터 스토리지를 분리할 수 있습니다. Amazon S3 스토리지 클래스는 다양한 액세스 패턴에 맞게 가장 저렴한 스토리지를 제공하도록 특별히 구축되었습니다. 또한 AWS 컴퓨팅 도구(예: Amazon Athena, AWS Glue, Amazon Redshift, Amazon SageMaker 런타임)는 서버리스 도구이므로 인프라를 관리할 필요가 없으며 사용한 만큼만 비용을 지불하면 됩니다.

고급

비즈니스 및 운영 기능을 지원하는 표준 분석부터 예측 및 인사이트를 지원하는 더 복잡한 기능에 이르기까지 데이터 사용량의 폭을 넓히고 더 빠른 의사 결정을 지원하도록 최신 데이터 아키텍처를 더욱 강화할 수 있습니다. 이를 위해 아키텍처는 다음 섹션에서 설명하는 기능을 지원합니다.

특성 엔지니어링 이해

특성 엔지니어링은 기계 학습을 사용하고 특성 저장소 또는 특성 마트 설정을 수행합니다. 데이터 과학 팀은 지도 학습 모델과 비지도 학습 모델 모두에 대한 새로운 기능(파생 속성)을 생성하고 특성 마트에 저장하여 변환을 간소화하고 데이터 정확도를 높입니다. 엔터프라이즈는 여러 분석 모델에서 기능을 재사용할 수 있으므로 출시 속도가 개선됩니다.

데이터세트 비정규화를 위한 계획

비정규화된 데이터세트 또는 데이터 마트를 구성하면 필요한 데이터를 단일 위치에서 쉽게 사용할 수 있고 분석 속도를 높여 비즈니스 사용자의 데이터세트를 크게 간소화할 수 있습니다. 신중하게 설계한 경우 하나의 레코드가 여러 사용 모델을 지원하고 전체 개발 수명 주기를 줄일 수 있습니다. 또한 비정규화된 데이터세트에 대한 효과적인 거버넌스는 두 가지 이유에서 중요합니다. 비정규화된 데이터를 구현하면 많은 중복 데이터세트가 생성되어 대규모로 관리하기 어려울 수 있습니다. 또한 이러한 데이터세트가 올바르게 모델링되지 않으면 용도를 변경하기가 점점 더 어려울 수 있습니다.

이식성 및 확장성 설계

대규모 조직은 단일 데이터 플랫폼에 모든 애플리케이션과 사용자를 보유하는 경우가 거의 없습니다. 조직의 애플리케이션과 데이터 저장소는 일반적으로 레거시 온프레미스와 클라우드 플랫폼에 분산되므로 분석 팀이 데이터를 혼합하고 병합하기가 어렵습니다. 이 경우 도메인, 지역, 비즈니스 사용 사례 등과 같은 특성을 기반으로 데이터를 컨테이너화하는 것이 좋습니다. 이러한 컨테이너화는 다양한 플랫폼과 애플리케이션 간 이동성을 높이고 보다 효과적인 사용을 지원합니다. 데이터를 컨테이너로 분할하고 API를 통해 노출하면 데이터 아키텍처를 더 쉽게 확장할 수 있습니다. 포괄적인 하이브리드 데이터 흐름을 지원하고 온프레미스 및 클라우드 기반 애플리케이션이 원활하게 작동하도록 지원합니다.

Excel

조직 내에서 최신 분석 아키텍처가 발전함에 따라 재사용 가능한 개념을 도입하여 이러한 변경 사항을 관리하는 것이 중요합니다. 이 개념은 비용을 관리하면서 내구성과 채택률을 높입니다. 다음 섹션에서 고려해야 할 몇 가지 개념을 설명합니다.

구성 가능한 프레임워크 설계

조직은 종종 고유한 비즈니스 요구 사항을 해결하기 위해 여러 복잡한 모델을 생성합니다. 이러한 모델은 여러 데이터 파이프라인과 엔지니어링된 특성을 생성해야 합니다. 시간이 지남에 따라 상당한 중복성이 발생하고 운영 비용이 증가합니다. 파라미터 기반의 구성 가능한 기본 모델 세트를 통합하는 프레

임워크를 생성하면 개발 시간과 운영 비용이 절감됩니다. 분석 엔진은 이러한 구성 가능한 모델을 구현하여 원하는 출력을 제공할 수 있습니다.

통합 분석 엔진 빌드를 위한 계획

비즈니스 문제는 고유하며 요구 사항을 해결하기 위해 사용자 지정 기술이 필요한 경우가 많으므로 조직에는 여러 분석 엔진이 있습니다. 여러 프로그래밍 패러다임을 지원할 수 있는 통합 AI 기반 분석 엔진 인터페이스를 설계하고 개발하면 사용이 간소화되고 비용이 절감됩니다.

DataOps 정의

대부분의 데이터 전문가는 올바른 데이터 찾기, 변환, 모델링 등과 같은 데이터 작업을 수행하는 데 상당한 시간을 소비합니다. 애자일 데이터 운영(DataOps)을 활용하면 데이터 엔지니어, 데이터 과학자, 데이터 소유자 및 분석가의 사일로를 허물고 데이터 아키텍처를 크게 개선할 수 있습니다. DataOps를 통해 팀 간의 통신을 개선하고, 주기 시간을 줄이며, 높은 데이터 품질을 보장할 수 있습니다. 데이터 및 분석 아키텍처는 변화하는 비즈니스 요구 사항과 기술 발전으로 인해 시간이 지남에 따라 수많은 변환을 거쳤습니다. 조직은 시간이 지남에 따라 발전하며 비즈니스를 지원하는 데이터 및 분석 아키텍처를 개발, 구현 및 유지하기 위해 노력해야 합니다.

데이터 엔지니어링

조직 전체에서 데이터 흐름을 자동화하고 오케스트레이션합니다.

메타데이터를 사용하여 원시 데이터를 처리하고 최적화된 출력을 생성하는 파이프라인을 자동화합니다. 여러 AWS CAF 플랫폼 아키텍처 및 플랫폼 엔지니어링 기능과 운영 관점에서 정의된 기존 아키텍처 가드레일 및 보안 제어를 활용합니다. 플랫폼 엔지니어링 지원 팀과 협력하여 파이프라인 배포를 간소화하는 일반적인 패턴에 대해 재사용 가능한 블루프린트를 개발합니다.

시작

데이터 레이크 배포

정형 및 비정형 데이터에 적합한 스토리지 솔루션을 사용하여 기본 데이터 스토리지 기능을 설정합니다. 이를 통해 다양한 소스에서 데이터를 수집 및 저장할 수 있으며 추가 처리 및 분석을 위해 데이터에 액세스할 수 있습니다. 데이터 스토리지는 데이터 엔지니어링 전략의 중요한 구성 요소입니다. 잘 설계된 데이터 스토리지 아키텍처를 통해 조직은 데이터를 효율적이고 비용 효율적으로 저장, 관리 및 액세스할 수 있습니다. AWS에서는 특정 비즈니스 요구 사항을 충족하는 다양한 데이터 스토리지 서비스를 제공합니다.

예를 들어 객체 스토리지에는 Amazon Simple Storage Service(Amazon S3), 관계형 데이터베이스에는 Amazon Relational Database Service(Amazon RDS), 데이터 웨어하우징에는 Amazon Redshift를 사용하여 기본적인 데이터 스토리지 기능을 설정할 수 있습니다. 이러한 서비스를 사용하면 데이터를 안전하고 비용 효율적으로 저장하고 추가 처리 및 분석을 위해 데이터에 쉽게 액세스할 수 있습니다. 성능을 개선하고 비용을 절감하기 위해 데이터 분할 및 압축과 같은 데이터 스토리지 모범 사례도 구현하는 것이 좋습니다.

데이터 수집 패턴 개발

데이터 흐름을 자동화하고 오케스트레이션하려면 데이터베이스, 파일 및 API를 비롯한 다양한 소스에서 데이터를 수집하는 데이터 수집 프로세스를 설정합니다. 데이터 수집 프로세스는 비즈니스 민첩성을 지원하고 거버넌스 제어를 고려해야 합니다.

오케스트레이터는 클라우드 기반 서비스를 실행할 수 있어야 하며 자동화된 예약 메커니즘을 제공해야 합니다. 폴링 및 오류 처리 기능과 함께 여러 태스크 사이에서 조건부 링크 및 종속성에 대한 옵션을 제공해야 합니다. 또한 파이프라인이 원활하게 실행되도록 알림 및 모니터링 시스템과 원활하게 통합되어야 합니다.

다음은 몇 가지 널리 사용되는 오케스트레이션 메커니즘입니다.

- 시간 기반 오케스트레이션은 재귀적 간격과 정의된 빈도로 워크플로를 시작합니다.
- 이벤트 기반 오케스트레이션은 파일 생성 또는 API 요청과 같은 이벤트 발생을 기반으로 워크플로를 시작합니다.
- 폴링은 태스크 또는 워크플로가 API 등을 통해 서비스를 직접 호출하고 정의된 응답을 기다린 후 다음 단계를 진행하는 메커니즘을 구현합니다.

최신 아키텍처 설계에서는 클라우드에서 인프라 관리를 간소화하고 개발자 및 인프라 팀의 부담을 줄이는 관리형 서비스의 이점을 강조합니다. 이 접근 방식은 데이터 엔지니어링에도 적용됩니다. 해당하는 경우 관리형 서비스를 사용하여 데이터 수집 파이프라인을 빌드해 데이터 엔지니어링 프로세스를 가속화하는 것이 좋습니다. 이러한 서비스 유형의 두 가지 예로 Amazon Managed Workflows for Apache Airflow(Amazon MWAA) 및 AWS Step Functions가 있습니다.

- Apache Airflow는 프로그래밍 방식으로 워크플로를 작성, 예약 및 모니터링하는 데 널리 사용되는 오케스트레이션 도구입니다. AWS에서는 개발자가 오케스트레이션 도구의 인프라를 관리하는 대신 빌드에 집중할 수 있도록 관리형 서비스로 Amazon Managed Workflows for Apache Airflow(Amazon MWAA)를 제공합니다. Amazon MWAA를 사용하면 Python 스크립트를 통해 워크플로를 쉽게 작성할 수 있습니다. 방향성 비순환 그래프(DAG)는 각 태스크의 관계 및 종속성을 표시하는 방식을 통해 태스크 컬렉션으로 워크플로를 나타냅니다. 원하는 만큼 DAG를 보유할 수 있으며, Apache Airflow는 각 태스크의 관계 및 종속성에 따라 이를 실행합니다.
- AWS Step Functions를 사용하면 개발자가 IT 및 비즈니스 프로세스를 자동화하기 위한 로코드 시각적 워크플로를 빌드할 수 있습니다. Step Functions로 빌드하는 워크플로를 상태 시스템이라고 하며, 워크플로의 각 단계를 상태라고 합니다. Step Functions를 사용하여 기본 제공 오류 처리, 파라미터 전달, 권장 보안 설정 및 상태 관리를 위한 워크플로를 생성할 수 있습니다. 그러면 작성 및 유지 관리해야 하는 코드 양이 줄어듭니다. 태스크는 온프레미스 또는 클라우드 환경에서 호스팅하는 애플리케이션 또는 다른 AWS 서비스와 함께 작업을 수행합니다.

데이터 처리 가속화

데이터 처리는 현대 조직에서 수집하는 방대한 양의 데이터를 이해하는 데 중요한 단계입니다. 데이터 처리를 시작하기 위해 AWS에서는 강력한 추출, 전환, 적재(ETL) 기능을 제공하는 AWS Glue와 같은 관리형 서비스를 제공합니다. 조직은 이러한 서비스를 사용하여 데이터를 정리, 정규화 및 집계하여 분석을 준비하는 등 원시 데이터의 처리 및 변환을 시작할 수 있습니다.

데이터 처리는 초기 데이터 변환을 수행하기 위해 집계 및 필터링과 같은 간단한 기법으로 시작됩니다. 데이터 처리 요구 사항이 진화함에 따라 다양한 소스에서 데이터를 추출하고, 특정 요구 사항에 맞게 데이터를 전환하며, 통합 분석을 위해 중앙 집중식 데이터 웨어하우스 또는 데이터베이스에 적재할 수

있는 고급 ETL 프로세스를 구현할 수 있습니다. 이 접근 방식을 사용하면 데이터가 정확하고 완전하며 데이터를 적시에 분석에 사용할 수 있습니다.

데이터 처리에 AWS 관리형 서비스를 사용하면 조직은 더 높은 수준의 자동화, 확장성 및 비용 효율성과 같은 이점을 얻을 수 있습니다. 이러한 서비스는 스키마 검색, 데이터 프로파일링 및 데이터 변환과 같은 많은 일상적인 데이터 처리 태스크를 자동화하고 더 전략적인 활동을 위해 귀중한 리소스를 확보합니다. 또한 이러한 서비스는 증가하는 데이터 볼륨을 지원하도록 자동으로 조정됩니다.

데이터 시각화 서비스 제공

데이터 시각화를 사용하여 데이터를 의미 있고 빠르게 해석하는 의사 결정권자가 데이터를 사용할 수 있는 방법을 찾습니다. 시각화를 통해 패턴을 해석하고 기술 역량에 관계없이 다양한 이해관계자의 참여를 높일 수 있습니다. 좋은 플랫폼을 사용하면 데이터 엔지니어링 팀이 거의 오버헤드 없이 빠르게 데이터 시각화를 제공하는 리소스를 프로비저닝할 수 있습니다. 엔지니어링 전문 지식 없이 데이터 저장소를 쉽게 쿼리할 수 있는 도구를 사용하여 셀프 서비스 기능을 제공할 수도 있습니다. 데이터 시각적 객체와 대화형 대시보드를 통해 서버리스 비즈니스 인텔리전스를 제공하고, 자연어를 사용하여 백엔드 데이터를 쿼리할 수 있는 기본 제공 도구를 사용하는 방법을 고려합니다.

고급

거의 실시간에 가까운 데이터 처리 구현

데이터 처리는 모든 데이터 엔지니어링 파이프라인의 필수 구성 요소로, 이를 통해 조직은 원시 데이터를 의미 있는 인사이트로 변환할 수 있습니다. 기존의 배치 처리 외에도 실시간 데이터 처리는 오늘날 빠르게 변화하는 비즈니스 환경에서 점점 더 중요해지고 있습니다. 실시간 데이터 처리를 통해 조직은 이벤트 발생 시 이에 대응할 수 있으며 의사 결정 및 운영 효율성을 개선할 수 있습니다.

데이터 품질 검증

데이터 품질은 데이터에서 파생된 인사이트와 의사 결정의 정확성 및 신뢰성에 직접적인 영향을 줍니다. 분석에 신뢰할 수 있는 고품질 데이터를 사용하려면 데이터 검증 및 정리 프로세스를 구현해야 합니다.

데이터 검증에는 사전 정의된 규칙 및 기준과 비교하여 데이터의 정확성, 완전성 및 일관성을 확인하는 작업이 포함됩니다. 이를 통해 데이터의 불일치 또는 오류를 식별하고 목적에 맞는지 확인할 수 있습니다. 데이터 정리에는 데이터의 부정확성, 불일치 또는 중복을 식별하고 수정하는 작업이 포함됩니다.

조직은 데이터 품질 프로세스 및 도구를 구현하여 데이터에서 파생된 인사이트의 정확성과 신뢰성을 개선함으로써 의사 결정 및 운영 효율성을 개선할 수 있습니다. 그러면 조직의 성과가 향상될 뿐만 아니라 생성된 데이터 및 분석에 대한 이해관계자의 신뢰를 구축하고 신뢰도를 높일 수 있습니다.

데이터 변환 서비스 증명

데이터 변환에서는 고급 분석 및 기계 학습 모델을 위해 데이터를 준비합니다. 여기에는 데이터 정규화, 보강 및 중복 제거와 같은 기술을 사용하여 데이터가 깔끔하고 일관되며 분석에 사용할 준비가 되었는지 확인하는 작업이 포함됩니다.

- 데이터 정규화에는 데이터를 표준 형식으로 구성하고, 중복을 제거하며, 여러 소스에서 데이터가 일관되게 유지되도록 하는 작업이 포함됩니다. 이를 통해 여러 소스의 데이터를 더 쉽게 분석하고 비교할 수 있으며 조직은 운영을 보다 포괄적으로 이해할 수 있습니다.
- 데이터 보강에는 인구 통계 데이터 또는 시장 추세와 같은 외부 소스의 추가 정보로 기존 데이터를 개선하는 작업이 포함됩니다. 이를 통해 내부 데이터 소스만으로는 명확하지 않을 수 있는 고객 행동 또는 업계 추세에 대한 소중한 인사이트를 얻을 수 있습니다.
- 중복 제거에는 중복 데이터 항목을 식별 및 제거하고 데이터가 정확하고 오류가 없는지 확인하는 작업이 포함됩니다. 특히 작은 비율의 중복이라도 분석 결과를 왜곡할 수 있는 대규모 데이터세트를 처리할 때 중요합니다.

조직은 고급 데이터 변환 기술을 사용하여 데이터가 고품질이고 정확하며 해당 데이터로 더 복잡한 분석을 수행할 준비가 되었는지 확인합니다. 이를 통해 의사 결정이 개선되고 운영 효율성이 향상되며 시장에서 경쟁 우위를 확보할 수 있습니다.

데이터 대중화 지원

모든 직원이 데이터에 액세스하고 데이터를 이해하고 사용할 수 있도록 지원함으로써 데이터 대중화 문화를 장려합니다. 데이터 대중화는 직원이 데이터에 기반한 결정을 내리고 조직의 데이터 기반 문화에 기여하는 데 도움이 됩니다. 즉, 사일로를 허물고 모든 직원이 데이터를 공유하고 사용하여 의사 결정을 내리는 문화를 조성해야 합니다.

전반적으로 데이터 대중화는 조직의 모든 사람이 데이터 가치를 인정하고, 데이터에 액세스하며, 데이터를 이해할 수 있는 문화를 구축하는 것입니다. 데이터 대중화를 지원함으로써 조직은 혁신을 주도하고, 의사 결정을 개선하며, 궁극적으로 비즈니스 성공으로 이어지는 데이터 기반 문화를 조성합니다.

Excel

UI 기반 오케스트레이션 제공

민첩하고 효과적인 접근 방식을 사용하는 조직을 구축하려면 여러 사업부의 개발 및 운영 리소스에서 사용하는 최신 오케스트레이션 플랫폼을 계획하는 것이 중요합니다. 목표는 하나의 팀, 기술 또는 지원 모델에 의존하지 않고 데이터 파이프라인과 워크플로를 개발, 배포 및 공유하는 것입니다. 이는 UI 기반 오케스트레이션과 같은 기능을 통해 달성할 수 있습니다. 끌어서 놓기(drag-and-drop 상호 작용과 같은 기능을 사용하면 기술적 전문 지식이 거의 없는 사용자가 DAG를 구성하고 시스템 데이터 흐름을 명시할 수 있습니다. 그런 다음 이러한 구성 요소는 데이터 파이프라인을 오케스트레이션하는 실행 코드를 생성할 수 있습니다.

DataOps는 데이터 관리의 복잡성을 극복하고 조직 전체에서 원활한 데이터 흐름을 보장합니다. 메타데이터 기반 접근 방식은 조직의 필수 규정에 따라 데이터 품질과 규정 준수를 보장합니다. 마이크로서비스, 컨테이너화, 서버리스 함수와 같은 도구 세트에 투자하면 확장성과 민첩성이 개선됩니다.

데이터 엔지니어링 팀에 의존하여 데이터에서 가치를 창출하고 일상적인 인프라 태스크를 자동화에 맡기면 조직은 자동화 및 오케스트레이션에서 우수한 성과를 달성할 수 있습니다. 데이터 흐름 관리 작업의 실시간에 가까운 모니터링 및 로깅은 즉각적인 문제 해결 조치를 지원하고 데이터 흐름 파이프라인의 성능 및 보안을 개선합니다. 이러한 원칙은 안전한 데이터 공유 모델을 보장하면서 확장성과 성능을 달성하고 향후 성공을 위해 조직을 준비하는 데 도움이 됩니다.

DataOps 통합

DataOps는 데이터 파이프라인 생성, 테스트 및 배포를 간소화하기 위해 개발 및 운영 프로세스의 통합을 강조하는 데이터 엔지니어링에 대한 최신 접근 방식입니다. DataOps 모범 사례를 구현하기 위해 조직은 코드형 인프라(IaC)와 지속적 통합 및 지속적 전송(CI/CD) 도구를 사용합니다. 이러한 도구는 자동화된 파이프라인 생성, 테스트 및 배포를 지원하며 효율성을 크게 개선하고 오류를 줄입니다. DataOps 팀은 플랫폼 엔지니어링 지원 팀과 협력하여 이러한 자동화를 빌드하므로 각 팀은 팀에서 가장 잘하는 업무에 집중할 수 있습니다.

DataOps 방법론을 구현하면 데이터 엔지니어, 데이터 과학자 및 비즈니스 사용자를 위한 협업 환경을 조성하고 데이터 파이프라인 및 분석 솔루션을 신속하게 개발, 배포 및 모니터링할 수 있습니다. 이 접근 방식은 여러 팀 간에 보다 원활한 통신과 협업을 지원하며 혁신을 가속화하고 결과를 개선합니다.

DataOps의 이점을 최대한 활용하려면 데이터 엔지니어링 프로세스를 간소화하는 것이 중요합니다. 코드 검토, 지속적 통합, 자동화된 테스트 등 플랫폼 엔지니어링 팀의 모범 사례를 사용하여 이를 달성합니다. 이러한 사례를 구현함으로써 조직은 데이터 파이프라인이 안정적이고 확장 가능하며 안전하고 비즈니스 및 기술 이해관계자의 요구 사항을 충족할 수 있도록 보장합니다.

프로비저닝 및 오케스트레이션

승인된 클라우드 제품의 카탈로그를 생성 및 관리하고 사용자에게 배포합니다.

조직이 성장함에 따라 일관되고 확장 가능하며 반복 가능한 방식으로 인프라를 프로비저닝하는 것이 더 어려워집니다. 간소화된 프로비저닝 및 오케스트레이션은 일관된 거버넌스를 달성하고 규정 준수 요구 사항을 충족하는 동시에 사용자가 승인된 클라우드 제품만 배포하도록 지원할 수 있습니다.

조직에서 사전 승인된 제품을 재사용하면 개발자가 조직의 보안 및 거버넌스 요구 사항을 충족하면서 애플리케이션을 더 빠르고 일관되게 빌드할 수 있습니다.

시작

허브 및 스포크 카탈로그 모델 배포

서비스 카탈로그에서 포트폴리오로 관리되는 소프트웨어 자산은 허브 및 스포크 패턴으로 하나 이상의 계정에 있는 사용자와 공유됩니다. 프라이빗 마켓플레이스와 프라이빗 제안을 사용하여 다양한 서드 파티 솔루션을 선별하고 코드형 인프라(IaC) 템플릿과 함께 배포할 수 있습니다.

빌더가 사전 승인된 제품을 소비할 수 있도록 하려면 이러한 제품을 검토, 승인하고 사용자에게 게시하는 프로세스를 정의합니다. 먼저 이러한 사전 승인된 제품이 포함된 중앙 관리형 리포지토리를 설계하고 구현합니다. 조직의 사용자가 각 제품을 소비해야 하는 경우 이 리포지토리의 라이선스 및 제품에 대한 액세스 권한을 부여하는 시스템을 설계합니다.

조직의 빌더가 승인을 위해 제품을 게시 메커니즘에 제출하도록 허용합니다. 그러면 승인 후 조직의 모든 사용자가 이러한 제품을 사용할 수 있습니다.

재사용을 위해 템플릿 선별

솔루션에 대한 IaC 템플릿을 코드화하고 허브 및 스포크 모델을 정의한 경우 각 스포크 계정에 프로비저닝/적용 및 소비 가능과 같은 두 가지 범주의 템플릿을 정의해야 합니다. 프로비저닝/적용 템플릿은 기본 기능으로 관리 계정에서 각 멤버 계정으로 직접 프로비저닝됩니다. 소비 가능 템플릿은 빌더가 셀프 서비스 방식으로 검색하고 프로비저닝하는 데 사용할 수 있습니다.

재사용을 위해 기본 파라미터 적용

빌더가 미리 선택할 수 있는 기본 파라미터가 포함된 IaC 템플릿을 구현합니다. 이를 통해 빌더는 각 파라미터의 세부 정보를 평가하지 않고도 거버넌스에 맞게 조정할 수 있으며 잘못된 선택을 방지합니다.

이 접근 방식은 설정에 필요한 항목만 노출합니다. 예를 들어 AWS Service Catalog는 특정 포트폴리오의 제품에 적용되는 규칙을 제어하는 제약 조건 기능을 사용하여 이 접근 방식을 구현합니다. 이 사용자 지정은 빌더 팀이 템플릿의 셀프 서비스 프로비저닝을 사용할 때 미리 구성됩니다.

승인 프로세스 설정

사용자는 승인되지 않은 제품 사용에 대한 비즈니스 근거가 있는 경우 해당 제품에 대한 액세스 요청을 제출할 수 있어야 합니다. 사용 중인 제품에 대한 업데이트를 사용할 수 있을 때 사용자에게 이를 알리는 알림 시스템을 빌드합니다. 그러면 최신 보안 업데이트를 준수할 수 있습니다.

빌더가 셀프 서비스 포털을 통해 검토할 새 제품을 제출할 워크플로를 설정합니다. 빌더는 포털을 사용하여 제품의 대상을 정의하고 제품에 액세스해야 하는 사용자 그룹을 식별할 수 있습니다. 제출할 때마다 정의된 프로세스를 사용하여 제품을 검토 및 승인하고 셀프 서비스 포털에 게시합니다.

고급

셀프 서비스 포털 생성

셀프 서비스 포털을 생성하여 승인된 클라우드 제품을 배포, 검색 및 소비합니다. 조직의 사용자는 이 포털을 사용하여 인프라를 빌드하고 애플리케이션을 환경에 배포하는 데 필요한 제품을 검색할 수 있습니다. 포털에서 제품에 액세스할 수 있는 사용자의 권한 경계를 설정하고 사용자가 라이선스가 부여된 제품을 소비할 수 있는 횟수에 대한 제한을 설정합니다. 계정은 AWS Control Tower에 대한 사용자 지정과 같은 솔루션을 사용하여 생성되므로 각 스포크 계정에서 직접 프로비저닝하거나 셀프 서비스 모델로 사용할 수 있는 기본 리소스 세트를 정의합니다.

프라이빗 마켓플레이스 활성화

프라이빗 마켓플레이스는 구매한 제품(소프트웨어, 데이터 및 전문 서비스)의 엄선된 카탈로그를 제공하며 허브 및 스포크 패턴(하나의 관리 계정과 여러 멤버 계정 포함)으로 구현되므로 스포크 계정은 승인된 소프트웨어만 구독할 수 있습니다. 이 제품 거버넌스는 소프트웨어 비용을 제어하고 법률 및 계약 검토를 간소화하는 데 도움이 됩니다. 관리 계정 수준에서 기본 허브 역할을 수행할 프라이빗 마켓플레이스를 생성합니다.

권한 관리

권한 부여된 사용자 및 워크로드만 벤더에서 정의한 한도 내에서 라이선스를 소비하도록 허용하는 제어를 활성화합니다. 그러면 비용이 많이 드는 감사와 예상치 못한 라이선스 조정의 위험을 줄일 수 있습니다.

Excel

조달 시스템과 통합

AWS Marketplace에 통합하여 기존 조달 프로세스를 보완합니다. 사용자가 기존 조달 및 승인 프로세스에 따라 소프트웨어를 얻을 수 있도록 조달 시스템(Coupa 또는 SAP Ariba)을 프라이빗 마켓플레이스로 확장하여 수행합니다. 적절한 IAM 관리형 권한을 생성하고, AWS Marketplace 를 사용하여 조달 솔루션을 구성하는 데 필요한 정보를 생성하고, 통합을 완료하도록 조달 솔루션을 구성합니다. 예를 들어 펀치아웃을 설정하고 인 AWS 보이스에 구매 주문을 연결한 다음 표준 프로비저닝 솔루션을 사용하도록 조달 프로세스를 조정할 수 있습니다.

빌더가 내부 API를 통해 사전 승인된 제품에 액세스할 수 있으므로 사용자는 제품을 애플리케이션에 통합하거나 팀이 제품을 소비할 수 있도록 개인화된 포털을 빌드할 수 있습니다. 새 제품을 생성하기 위한 제출 및 게시 프로세스를 통합하며, 사용자가 API를 통해 새 라이선스를 요청하고 제품에 액세스하도록 허용합니다.

ITSM 도구와 통합

해당하는 경우 IT 서비스 관리(ITSM) 도구에 연결하고 구성 관리 데이터베이스(CMDB)에 대한 업데이트를 자동화합니다. 조직이 사용하는 제품을 평가하는 프로세스와 메커니즘을 설정합니다. 사전 승인된 제품에 대해 규정 준수를 위해 업데이트해야 함을 사용자에게 알리는 메커니즘을 설정합니다. ITSM 도구를 사용하여 환경을 분석하고 중요한 업데이트가 필요할 때 조직 전체에서 제품에 보안 및 규정 준수 업데이트를 푸시합니다.

수명 주기 관리 및 버전 배포 시스템 구현

개발 수명 주기 동안 IaC 템플릿 버전과 템플릿에서 프로비저닝된 서비스 버전을 유지 관리합니다. 카탈로그에 구현한 허브 및 스포크 모델을 사용하여 스포크 수준에서 강제 업데이트가 필요한지(예: 셀프 서비스 프로비저닝에 동시 버전을 사용할 수 있는 경우)와 폐기를 위해 표시해야 하는 버전을 정의할 수 있습니다. 허브 및 스포크 카탈로그를 사용하면 필요에 따라 새 버전의 감사 및 배포를 관리하는 데도 도움이 됩니다.

현대적 애플리케이션 개발

잘 설계된 클라우드 네이티브 애플리케이션을 빌드합니다.

최신 애플리케이션 개발 사례는 조직이 잘 설계된 클라우드 네이티브 애플리케이션을 빌드하고 경쟁 우위를 유지하는 데 필수적입니다. 비즈니스는 컨테이너 및 서버리스 컴퓨팅과 같은 클라우드 네이티브 기술을 사용하여 변화하는 시장 수요에 맞게 확장 가능하고 민첩한 애플리케이션을 생성할 수 있습니다. 이러한 기술을 통해 조직은 리소스 사용률을 최적화하고 비용을 절감하며 애플리케이션의 성능을 개선할 수 있습니다.

최신 애플리케이션을 설계할 때 운영 및 개발을 위한 애자일 솔루션을 개발합니다. 최신 애플리케이션은 고객 수요 변화에 자동으로 대응하며 장애에 대해 뛰어난 복원력을 제공합니다. 엔지니어는 변경 사항을 신속하게 개발 및 배포하고 애플리케이션 성능을 모니터링할 수 있습니다. 최신 애플리케이션은 자체 복구가 가능하도록 설계되었으며, 필요할 때 트래픽과 비용을 전혀 들이지 않으면서 대규모 또는 소규모 트래픽으로 규모를 조정할 수 있습니다.

잘 설계된 클라우드 네이티브 애플리케이션을 빌드하려면 기본 기술과 관련 모범 사례를 심층적으로 이해해야 합니다. 조직은 마이크로서비스 아키텍처를 채택하고 애플리케이션을 모듈식으로 느슨하게 결합하도록 설계하여 독립적인 배포와 확장성을 허용해야 합니다. 이 접근 방식을 통해 조직은 애플리케이션을 빠르고 독립적으로 개발, 테스트 및 배포 가능한 더 작고 관리 가능한 구성 요소로 나눌 수 있습니다.

시작

최신 접근 방식 살펴보기

먼저 컨테이너, 서버리스 기술 및 마이크로서비스 개발을 지원하는 기타 접근 방식을 조사합니다. 이러한 방법은 리소스 효율성을 높이고 보안을 개선하며 인프라 비용을 최소화합니다. 기존의 차별화 및 엔터프라이즈 애플리케이션을 현대화하여 효율성을 개선하고 기존 투자의 가치를 극대화하려고 합니다. 가치 기반 의사 결정에 따라 리플랫포밍(자체 관리형 컨테이너, 데이터베이스 또는 메시지 브로커를 관리형 클라우드 서비스로 이전) 및 리팩터링(애플리케이션을 재개발하여 클라우드 네이티브 아키텍처 채택)을 고려합니다.

기존 클라우드 기반 애플리케이션을 업데이트하는 경우 성공적인 접근 방식으로, 스트랭글러 피그 패턴을 사용하여 아키텍처를 마이크로서비스로 점진적으로 분해하는 방법이 포함됩니다. 이 절차는 현대적인 애플리케이션 방법론을 채택하는 데 도움이 되므로 내재된 이점을 실현하고 대규모 조직에서 그 가치를 입증할 수 있습니다. 해당하는 경우 이벤트 기반 아키텍처를 활용하는 고유한 마이크로서비

스로 애플리케이션을 구성하는 방법을 고려합니다. 워크로드 성능 또는 신뢰성에 영향을 주지 않도록 아키텍처에서 변경할 수 없는 서비스 할당량과 물리적 리소스를 고려해야 합니다.

클라우드 네이티브 컴퓨팅 기능 채택

클라우드 네이티브 컴퓨팅 기능은 최신 애플리케이션 개발의 핵심입니다. 이 접근 방식을 사용하려면 조직이 컴퓨팅 유닛을 호스팅하는 방법을 고려하고 각 사용 사례 또는 서비스에 가장 적합한 옵션을 식별해야 합니다. 예를 들어 AWS Lambda는 애플리케이션 코드를 실행하기 위한 서버리스 메커니즘을 제공하고 이벤트 기반 아키텍처에서 중요한 역할을 합니다. Lambda 함수는 온디맨드 방식으로 시작되고 정의된 최대 동시성까지 병렬로 실행되므로, 다양한 태스크를 수행하도록 확장할 수 있습니다.

컨테이너화 사용

최신 소프트웨어 개발에서 애플리케이션 및 종속성 관리는 특히 다양한 환경에서 일관성을 유지해야 한다는 점을 고려할 때 점점 더 복잡한 태스크가 되었습니다. 이러한 문제를 해결하기 위해 Docker와 같은 컨테이너화 기술DL 애플리케이션 및 해당 종속성을 패키징하기 위한 효과적인 솔루션으로 부상했습니다. 컨테이너는 애플리케이션의 런타임 환경에 관계없이 일관되고 재현 가능한 배포를 보장하므로 로컬 환경의 개발이 클라우드 환경의 프로덕션 개발과 동일한 방식으로 작동합니다. 이 접근 방식은 환경 또는 해당 구성 내 불일치로 인해 발생할 수 있는 오류를 줄입니다.

최신 데이터베이스 사용

최신 데이터베이스를 사용하는 경우 애플리케이션 내 각 마이크로서비스는 요구 사항을 충족하는 올바른 목적별 데이터베이스를 사용할 수 있어서 민첩성과 성능이 향상되고 비용은 절감됩니다. 예를 들어 한 마이크로서비스는 세션 데이터를 저장할 때 높은 처리량을 달성하기 위해 NoSQL 데이터베이스를 사용할 수 있고, 다른 마이크로서비스는 관계형 데이터베이스를 사용하여 복잡한 테이블 조인을 수행할 수 있으며, 또 다른 마이크로서비스는 양자 원장 데이터베이스를 사용하여 블록체인의 변경 사항을 추적할 수 있습니다.

최신 데이터베이스는 확장성과 유연성을 제공합니다. 또한 기존 데이터베이스보다 더 나은 보안, 규정 준수 및 신뢰성을 제공하는 데 도움이 됩니다. 이를 통해 조직은 데이터를 더 효율적으로 저장 및 관리하고 애플리케이션은 적시에 올바른 데이터에 액세스할 수 있으므로 성능과 사용자 경험이 개선됩니다.

최신 데이터베이스로 마이그레이션하는 것은 최신 애플리케이션 개발의 중요한 부분입니다. 조직은 올바른 데이터 스토리지 솔루션을 사용하여 데이터 관리 기능을 최적화하고 보다 효율적이고 신뢰할 수 있는 애플리케이션을 제공할 수 있습니다. 각 마이크로서비스를 독립적으로 구성하고 각 마이크로서비스에 대한 올바른 기술을 선택하면 조직은 비용을 최소화하면서 효율성과 확장성을 극대화할 수 있도록 데이터 기능을 추가로 최적화할 수 있습니다.

고급

최신 아키텍처 최적화

추가 최적화를 달성하려면 서버리스 기술의 구현을 세부 조정하고 Amazon API Gateway 및 AWS Lambda와 같은 AWS 서비스를 사용하여 독립적으로 규모를 조정하고 배포할 수 있는 아키텍처를 개발합니다. Amazon Route 53 및 AWS Cloud Map을 사용하여 서비스 검색을 구현해 구성 요소 간에 원활한 통신을 보장합니다.

API 버전 관리, 캐싱 및 사용량 제한 방식을 채택하여 다양한 애플리케이션 버전에서 호환성과 성능을 유지 관리합니다. AWS Identity and Access Management(IAM) 및 리소스 정책을 사용하여 보안을 강화합니다. 그러면 인프라를 보호하고 권한 있는 엔터티에만 액세스 권한을 부여할 수 있습니다.

가능한 경우 서버리스 서비스를 사용하여 기본 인프라를 관리할 필요 없이 컨테이너를 실행합니다. 이를 통해 핵심 애플리케이션 개발에 집중할 수 있으며 리소스 관리 및 성능을 개선할 수 있습니다. 또한 확장성, 유연성 및 비용 효율성의 이점을 최대한 활용할 수 있습니다.

서버리스 아키텍처의 복잡성을 심층 분석하고 이러한 고급 사례를 통합하면 조직은 개선 및 미세 조정 기회를 발견하고 궁극적으로 클라우드 네이티브 애플리케이션의 잠재력을 극대화할 수 있습니다. 이러한 노력은 전반적인 사용자 경험을 더욱 개선시키는 보다 정교한 애플리케이션 패턴을 채택하는 데 도움이 됩니다. 또한 조직이 소프트웨어 개발 프로세스에서 더 민첩하고 효율적으로 처리할 수 있도록 지원합니다.

서비스 메시 기술 사용

조직이 애플리케이션을 빌드하고 배포하기 위한 마이크로서비스 아키텍처를 더 많이 채택함에 따라 이러한 서비스 간의 복잡성, 보안 및 통신을 관리하는 것이 중요합니다. Istio, Linkerd 또는 Consul과 같은 서비스 메시 기술은 마이크로서비스의 보안, 관찰성 및 신뢰성을 개선하는 데 중요한 역할을 합니다.

가시성 및 추적성 보장

최신 사례에서는 개발 프로세스에서 가시성과 추적성을 높이고 업계 표준 및 모범 사례를 더 쉽게 준수할 수 있습니다. 가시성과 모니터링은 최신 애플리케이션 개발에 필수적입니다. 애플리케이션 성능에 대한 중요한 인사이트를 제공하기 위해 모니터링 및 로깅 솔루션을 구현하면 조직은 개선이 필요한 영역을 식별하고 애플리케이션을 최적화할 수 있습니다. 플랫폼 엔지니어링 팀과 협력하여 애플리케이션 오류, 성능 및 규정 준수에 대한 포괄적인 가시성과 모니터링을 제공하기 위해 여러 도구를 사용할 수 있어야 하며 이를 바탕으로 문제를 신속하게 감지, 진단 및 해결할 수 있습니다.

Excel

마이크로서비스 수용

많은 조직에서 최신 애플리케이션 개발은 비즈니스 성공과 동의어입니다. 마이크로서비스는 이러한 트랜스포메이션의 핵심이며 조직은 이러한 강력한 아키텍처 패턴을 수용하여 이점을 얻을 수 있습니다.

마이크로서비스는 확장성과 복원력이 뛰어나고 민첩한 애플리케이션 아키텍처를 제공합니다. 애플리케이션을 독립적으로 배포 가능한 소규모 서비스로 분류하여 조직은 애플리케이션의 다른 부분에 영향을 주지 않고 특정 구성 요소를 빠르게 반복하도록 선택할 수 있습니다. 회로 차단기 및 벌크헤드와 같은 고급 복원력 패턴은 이러한 애플리케이션의 고가용성을 보장하는 데 중요한 역할을 합니다.

회로 차단기는 비정상 서비스에서 통신을 일시적으로 중지하거나 전환하여 문제를 복구할 수 있도록 계단식 장애를 차단하는 안전 메커니즘 역할을 합니다. 벌크헤드는 리소스를 격리하고 잠재적 장애의 영향 범위를 제한합니다. 이러한 패턴을 함께 사용하면 예상치 못한 중단을 견디고 최적의 성능을 유지하는 강력한 아키텍처가 생성됩니다.

마이크로서비스 구현의 또 다른 중요한 측면은 도메인 기반 설계(DDD) 원칙을 채택하는 것입니다. DDD는 비즈니스 도메인에 대한 공유된 이해 기반을 구축하고 이를 체계적인 소프트웨어 설계로 변환하는 데 중점을 둡니다. 이 접근 방식은 보다 응집력 있고 유지 관리 가능한 마이크로서비스로 이어지며, 조직의 요구 사항에 따라 애플리케이션의 단계별 발전을 보장합니다.

마이크로서비스 기반 애플리케이션에서는 서비스 간 통신을 최적화하는 작업도 중요합니다. 조직은 gRPC 또는 GraphQL과 같은 고급 프로토콜을 구현하여 서비스 간 통신 효율성을 크게 개선할 수 있습니다. 이러한 프로토콜은 유형의 안전, 짧은 지연 시간 및 유연성과 같은 기능을 제공하여 애플리케이션의 전반적인 성능과 유지 관리를 개선하는 데 도움이 됩니다.

마이크로서비스를 채택하는 조직은 혁신, 민첩성 및 협업을 장려하는 환경을 제공합니다. 개발 팀은 일반적으로 비즈니스 역량을 중심으로 구성되며 지속적 통합 및 지속적 전송(CI/CD) 사례에 집중합니다. 개발 팀은 공동 책임의 문화를 수용합니다. 또한 의사 결정을 내리고, 실험하며, 반복할 수 있는 이들의 능력을 뒷받침해줍니다.

지속적 통합 및 지속적 전송(CI/CD)

기존 소프트웨어 개발 및 인프라 관리 프로세스를 사용하는 조직보다 더 빠르게 애플리케이션과 서비스를 발전시키고 개선합니다.

지속적 통합 및 지속적 전송(CI/CD)을 통해 DevOps 사례를 채택하면 애플리케이션을 빌드, 테스트 및 배포하기 위한 간소화 및 자동화된 효율적인 프로세스가 촉진됩니다. CI/CD를 사용하면 소프트웨어를 신속하게 제공할 수 있으며, 배포 오류의 위험을 줄이고, 애플리케이션이 항상 최신 기능 및 버그 수정 사항으로 최신 상태를 유지할 수 있습니다. 주요 목표는 기존 소프트웨어 개발 및 인프라 관리 프로세스의 사용에서 발전하여 애플리케이션과 서비스를 더 빠른 속도로 발전시키고 개선하는 것입니다.

시작

소프트웨어 구성 요소 관리 채택

소프트웨어 구성 요소 관리는 라이브러리, 프레임워크, 소스 코드 리포지토리, 모듈, 아티팩트 및 서드파티 종속성을 포함하여 소프트웨어를 빌드하는 데 사용되는 모든 개별 구성 요소를 관리하는 사례입니다. Git 또는 Apache Subversion과 같은 버전 제어 시스템을 사용하여 소스 코드를 관리하고, 협업을 활성화하며, 코드 변경 기록을 유지하는 것이 좋습니다. 리포지토리의 변경 사항 및 이벤트를 모니터링하여 프로세스를 자동화하고, 파이프라인을 생성하며, 코드를 관리하고, 필요에 따라 워크플로를 추가 서비스와 통합할 수 있습니다.

CI/CD 파이프라인 생성

CI/CD 파이프라인은 버전 제어 시스템에 커밋된 변경 사항에 의해 시작되는 일련의 자동화된 명령입니다. 여기에는 일반적으로 애플리케이션 빌드, 자동화된 테스트 실행, 특정 환경에 코드 배포에 대한 명령이 포함됩니다. AWS CodePipeline, Jenkins, GitLab 또는 CircleCI와 같은 도구를 사용하여 자동화된 CI/CD 파이프라인을 설정할 수 있습니다. 파이프라인 생성을 지원하는 버전 관리 시스템에서 직접 설정할 수도 있습니다.

지속적 통합을 위해 실행 가능한 최소 파이프라인으로 시작한 다음 더 많은 작업과 단계가 포함된 지속적 전송 파이프라인으로 전환합니다. 지속적 전송 구성을 코드로 처리합니다. 각 브랜치 및 팀에 대해 여러 개의 고유한 파이프라인을 사용할 수 있으므로, 설정해야 하는 구성 변수와 파이프라인을 사용할 팀을 가장 잘 지원하는 방법을 생각해 보세요.

코드를 배포하려는 날짜 및 시간인 배포 기간을 고려합니다. 시스템의 수요가 적은 시간을 고려합니다. 롤백해야 하는 경우 이때가 고객에게 미치는 영향이 가장 적습니다. 다른 모범 사례로, 금요일에는 배

포를 피하고 성수기 또는 공휴일 이전에 코드 동결을 구현하는 것이 포함됩니다. 커밋 작성자를 사용할 수 없는 경우(예: 휴가 중) 코드 배포 규칙을 정의하는 것이 좋습니다. 배포에 실패하고 외부 도움이 필요할 수 있습니다. 인플레이스, 롤링, 변경 불가능, 블루/그린 배포와 같은 여러 배포 방법을 평가합니다. 가용성과 보안을 높이는 동시에 복잡성과 관리를 최소화하려면 지속적 전송 워크플로에 완전관리형 서비스를 사용하는 방법을 고려합니다.

자동화된 테스트 배포

최신 사례에서는 리포지토리에 커밋되고 파이프라인을 시작하기 전에 문제를 감지하고 해결하도록 시프트 레프트(테스트를 개발자 및 IDE에 더 가깝게 그리고 수명 주기 초기에 이동하는 방식)가 권장됩니다. 개발자가 코딩하는 동안 오류가 감지되므로 이 사례에는 개발자와의 빠른 피드백 루프가 포함됩니다. 시프트 레프트 방식에서는 저렴한 비용이 특징입니다. 테스트할 때 파이프라인을 실행할 필요가 없기 때문입니다. 이때 비동기 피드백이 제공되고 운영 비용이 증가할 수 있습니다.

자동화된 테스트는 개발 프로세스 초기에 오류를 포착하며 유닛 테스트, 통합 테스트 및 기능 테스트를 포함합니다. 개발자에게 도구를 사용하여 유닛 테스트를 최대한 빨리 생성하고 코드를 중앙 리포지토리로 푸시하기 전에 실행하도록 장려하는 것이 좋습니다. 또한 자동화된 프로세스에 정적 코드 분석, 성능 벤치마킹 및 보안 애플리케이션 테스트가 포함되어 있는지 확인합니다.

설명서 생성

CI/CD 파이프라인을 구현하여 개발 워크플로를 간소화하는 것 외에도 파이프라인의 지속적인 효율성, 유지 관리 가능성 및 확장성을 보장하기 위해 명확하고 포괄적인 설명서를 유지 관리해야 합니다. 설명서는 이를 통해 개발 팀이 파이프라인의 설계, 구성 요소 및 프로세스를 명확하게 이해할 수 있기 때문에 CI/CD 파이프라인의 중요한 측면입니다. 설명서를 생성할 때 파이프라인 개요로 시작하고, 아키텍처 및 설계 장단점을 설명하며, 사용 중인 도구 및 기술을 설명하고, 초기 구성 및 설정을 지정하며, 보안 조치 및 액세스 제어를 간략하게 설명하고, 문제 해결 및 유지 관리 정보를 포함합니다.

코드형 인프라(IaC) 사용

Terraform, Ansible 또는 AWS CloudFormation와 같은 도구를 사용하여 인프라를 관리하고 일관되고 재현 가능한 환경을 보장합니다. 인프라를 코드로 처리하고, 인프라의 변경 사항을 추적하며, 콘솔에서 직접 변경하지 않도록 합니다. 데이터베이스 프로비저닝을 포함한 모든 인프라를 코드로 정의하고 파이프라인을 사용하여 이러한 변경 사항을 배포합니다. 제거된 프로덕션 데이터의 작은 하위 세트가 있는 파이프라인에서 데이터베이스 통합을 코드로 실행하는 방법을 고려합니다. 가능하면 변경을 수행하고 코드에서 해당 변경 사항을 추적합니다.

소프트웨어 코드와 마찬가지로 인프라 코드에 대한 다음 모범 사례를 따릅니다.

- 버전 제어를 사용합니다.
- 버그 추적 및 티켓팅 시스템을 사용합니다.
- 동료가 변경 사항을 적용하기 전에 검토하도록 합니다.
- 인프라 코드 패턴 및 설계를 설정합니다.
- 인프라 변경 사항을 테스트합니다.

표준 지표 유지 및 추적

높은 수준의 성능을 유지하려면 다음을 포함하여 파이프라인의 상태 및 비즈니스 영향을 이해하기 위해 주요 지표를 개발하고 이를 기준으로 추적합니다.

- 빌드 빈도. 빌드 수는 팀의 생산성과 변경 사항의 복잡성에 대한 인사이트를 제공합니다.
- 배포 빈도. 정기적인 배포는 정상적이고 민첩한 개발 프로세스를 나타냅니다.
- 변경 사항에 대한 리드 타임. 변경 사항이 프로덕션에 도달하는 평균 시간을 측정하면 배포 프로세스의 병목 현상을 식별하는 데 도움이 될 수 있습니다.
- 파이프라인까지 평균 시간. 초기 파이프라인 단계에서 각 후속 단계까지의 평균 시간은 워크플로를 최적화하는 데 도움이 될 수 있습니다.
- 프로덕션 변경 볼륨. 프로덕션에 도달한 변경 횟수를 추적하면 프로덕션 환경의 안정성에 대한 인사이트를 얻을 수 있습니다.
- 빌드 시간. 평균 빌드 시간은 코드베이스 또는 인프라의 잠재적 문제를 나타낼 수 있습니다.

고급

구성 관리 사용

구성 관리 도구는 소프트웨어 및 인프라의 배포, 구성, 관리를 자동화하는 데 중요한 역할을 합니다. 이러한 도구는 다양한 환경에서 인프라, 소프트웨어 및 구성의 원하는 상태를 유지하고 변경 사항을 처리하는 체계적인 접근 방식을 제공합니다. 이러한 도구를 통해 개발자는 선언적 또는 필수 언어를 사용하여 원하는 시스템 상태를 정의할 수 있습니다. 그런 다음 구성 관리 도구는 이러한 구성을 대상 시스템에 적용하는 프로세스를 자동화하여 일관성과 반복성을 보장합니다.

구성 관리 도구를 사용하여 소프트웨어 및 인프라의 배포, 구성, 관리를 자동화합니다. AWS Systems Manager State Manager는 안전하고 확장 가능한 구성 관리 서비스로, 관리형 노드 및 다른 AWS 리소스를 사용자가 정의한 상태로 유지하는 프로세스를 자동화합니다.

모니터링 및 로깅 통합

모니터링 및 로깅 솔루션을 CD 파이프라인에 통합하면 개발 팀과 전체 소프트웨어 개발 프로세스에 많은 이점이 있습니다. 이러한 솔루션은 애플리케이션 성능에 대한 실시간 인사이트를 제공하고, 문제를 더 빠르게 식별 및 해결하며, 지속적 개선을 촉진하여 애플리케이션이 수명 주기 전반에 걸쳐 신뢰성, 성능 및 확장성을 유지할 수 있도록 지원합니다. 모니터링 및 로깅 솔루션에 대한 투자는 강력하고 효율적인 CD 파이프라인을 유지 관리하는 핵심이며, 궁극적으로 고품질 소프트웨어를 성공적으로 전달하는 데 기여합니다.

병합을 위한 케이던스 생성

코드 변경 사항을 매일 한 번 이상 또는 각 태스크 이후 하루에 여러 번 기본 라인(트렁크 또는 기본) 브랜치에 커밋하거나 병합합니다. 이 케이던스로 인해 일일 파이프라인 간접 호출이 여러 번 발생합니다. 풀 기반 분기 워크플로 모델이 이 접근 방식에 부합합니다. 기능 플래그, 다크 런칭 및 유사한 기술을 사용하여 고객이 사용하는 기능을 사용자 지정합니다.

배포 후 동작 캡처

배포 후 자동화된 가상 테스트를 사용하여 프로덕션 동작을 캡처하고 결과를 지속적 전송 파이프라인과 동기화하여 수정 조치가 즉시 수행되도록 합니다. 개발자의 최고 우선순위는 파이프라인에서 발견된 오류를 최대한 빨리 수정하고, 소스 코드 리포지토리에 코드 변경을 커밋하며, 파이프라인에서 오류 해결을 확인하는 것입니다.

배포 후 모범 사례에는 가장 중요한 핵심 성과 지표(KPI)를 관찰하고 프로덕션 환경에 오류가 없는지 확인하는 작업이 포함됩니다. 오류 처리 및 배포 후 KPI 평가를 자동화하여 릴리스의 영향을 정량화합니다. 개발자가 개선에 사용할 수 있는 속도, 보안 및 안정성 지표를 자동으로 생성합니다. 자세한 내용은 AWS의 DevOps Monitoring Dashboard를 참조하세요.

Excel

최적의 성능을 위해 최첨단 관행 및 기술을 채택합니다. CI/CD 프로세스를 지속적으로 개선하면 소프트웨어 품질을 개선하고, 출시 시간을 단축하며, 민첩성을 높일 수 있습니다. 새로운 기술과 도구가 지속적으로 출현하므로 조직은 최신 정보를 유지하고 경쟁 우위를 유지하기 위해 적응해야 합니다.

적응성을 유지하려면 다음을 고려하세요.

- 애플리케이션, 구성, 인프라, 데이터, AWS 계정 및 조직, 배포 파이프라인, 네트워킹, 보안 및 규정 준수 제어를 포함하여 모든 기능을 코드로 정의합니다.

- 컴퓨팅 이미지, 공유 서비스 및 애플리케이션에 해당하는 배포 파이프라인을 생성합니다.
- 코드에서 설명한 대로 기존 인프라 상태를 원하는 상태와 비교하여 풀 기반 요청이 변경 사항을 배포하는 워크플로를 시작하는 GitOps 모델을 고려합니다.
- CD 파이프라인을 사용하여 기계 학습(ML), 데이터, 사물 인터넷(IoT) 및 기타 워크로드를 배포하는 방법을 고려합니다.
- 모든 빌드 아티팩트에 디지털 서명을 하고 안전한 리포지토리에 저장합니다.
- 고객에게 배포된 모든 버전 관리 및 디지털 서명 아티팩트의 레코드를 생성하는 소프트웨어 BOM을 자동으로 생성하여 소프트웨어 출처를 추적합니다.
- 소프트웨어 전달 프로세스에서 모든 수동 활동을 제거한 후에는 수동 검토 보드를 제거합니다.

전체 소프트웨어 전달 프로세스를 자동화한 애플리케이션 및 서비스의 경우 팀이 파이프라인의 모든 검사를 통과한 변경 사항을 프로덕션의 고객에게 전달하는 지속적 배포를 고려합니다. 시각화의 경우 AWS 웹 사이트의 What is Continuous Delivery?에서 첫 번째 다이어그램을 참조하세요.

AI/ML 기술 통합

인공 지능(AI) 및 기계 학습(ML) 기술을 CI/CD 파이프라인에 통합하면 다음을 비롯해 몇 가지 이점이 있습니다.

- 자동화된 테스트 생성
- 지능형 테스트 우선순위 지정
- 문제 감지를 위한 예측 분석
- 이상 감지 및 근본 원인 분석
- 코드 검토 및 품질 보증
- 배포 최적화

자세한 내용은 AWS 웹 사이트의 개발자 작업에 인텔리전스 추가를 참조하세요.

카오스 엔지니어링 사례 채택

카오스 엔지니어링에는 예기치 않은 이벤트를 견디고 복구하는 역량을 테스트하기 위해 의도적으로 시스템에 장애를 주입하는 작업이 포함됩니다. 조직은 약점을 식별하고 선제적으로 해결함으로써 전반적인 시스템 신뢰성을 개선하고 잠재적 문제의 영향을 최소화할 수 있습니다.

카오스 엔지니어링 사례를 채택하여 Gremlin, Chaos Monkey 또는 Litmus와 같은 도구를 통해 시스템의 복원력을 테스트합니다. 제어된 실험을 정기적으로 실행하여 취약성을 식별하고, 내결함성을 검증

하며, 애플리케이션이 예상치 못한 장애를 원활하게 처리하도록 합니다. 이 선제적 접근 방식은 시스템 신뢰성을 개선하고 보다 강력한 CI/CD 파이프라인에 기여하는 데 도움이 됩니다.

성능 최적화

프로파일링 도구, 실시간 모니터링 및 피드백 루프를 사용하여 애플리케이션 성능을 지속적으로 최적화합니다. 애플리케이션이 늘어난 트래픽과 수요를 처리할 수 있도록 다음과 같은 기술을 적용합니다.

- 비용 최적화
- 프로파일링
- 실시간 모니터링
- 피드백 루프
- 캐싱
- 로드 밸런싱
- 확장성 및 성능 테스트

고급 관찰성 구현

클라우드 인프라의 관찰성을 높이는 작업은 지표, 로그 및 추적을 수집, 집계, 분석하는 기본적인 수준을 넘어섭니다. Amazon CloudWatch 및 AWS X-Ray와 같은 도구를 사용하여 관찰성이 향상되면 지속적 전송과 혁신을 촉진하는 전략적 사례로 발전합니다.

강력한 CI/CD 파이프라인에서 고급 관찰성을 활용하면 애플리케이션 및 인프라뿐만 아니라 파이프라인 자체를 포함한 전체 시스템의 성능 및 상태에 대한 인사이트를 발견할 수 있습니다. 이러한 인사이트로 다음과 같은 이점을 얻을 수 있습니다.

- 잠재적 문제를 신속하게 식별, 파악 및 해결하여 애플리케이션 안정성을 개선하고 가동 중지 시간을 줄입니다.
- CI/CD 프로세스를 간소화하여 더 빠르고 신뢰할 수 있는 전달 지원
- 코드 변경 및 배포의 영향에 대한 심층적인 인사이트를 확보하여 정보에 입각해 의사 결정을 촉진합니다.
- 운영 효율성 향상 및 비용 효율성을 개선하도록 리소스 사용률을 최적화합니다.

관찰성을 높이려면 다음을 수행합니다.

- 애플리케이션 및 인프라의 모든 계층에 관찰성을 포함시켜 시스템의 성능, 동작 및 상태에 대한 포괄적인 보기를 생성합니다.
- Amazon CloudWatch와 같은 도구를 사용하여 데이터 수집, 스토리지 및 분석을 중앙 집중화함으로서 쉽게 액세스하고 해석할 수 있도록 관찰성 데이터를 통합합니다.
- 분산 추적에 AWS X-Ray를 사용하여 애플리케이션과 기본 서비스의 성능을 파악합니다.
- 지속적 개선을 위해 피드백 루프를 설정하고 관찰성 데이터를 사용하여 시스템을 반복적으로 개선합니다.

고급 관찰성을 채택하는 것은 단순히 시스템을 유지 관리하는 것이 아니라 운영 우수성을 달성하고 조직에서 지속적 혁신을 추진하기 위한 전략적 움직임입니다.

GitOps 사례 구현

Git 리포지토리를 단일 정보 소스로 사용하여 인프라 및 애플리케이션 구성을 관리하는 GitOps 사례를 구현합니다. 이 접근 방식은 변경 관리를 간소화하고 추적성을 개선하며 환경 간 일관성을 보장합니다.

결론

이 가이드는 성공적인 클라우드 채택을 위한 기반을 성공적으로 구현하고 관리하기 위한 플레이북 역할을 합니다. 여기에서는 다음 방법을 설명합니다.

- 플랫폼 아키텍처의 기술적 과제와 복잡성을 직접 해결하여 클라우드 환경과 클라우드 환경에 상주하는 데이터에 대한 강력한 지침과 원칙을 설정합니다.
- 강력한 프로비저닝 및 오케스트레이션을 통해 플랫폼 엔지니어링을 빌드합니다.
- 확장 가능하고 반복 가능한 방식으로 승인된 클라우드 제품을 관리하고 사용자에게 배포하는 규정을 준수하는 다중 계정 클라우드 환경을 지원합니다.
- 데이터 엔지니어링에서 데이터 기반 의사 결정을 주도하는 데 필요한 도구를 사용하여 데이터 아키텍처 의사 결정을 지원합니다.
- 이러한 기능을 최신 애플리케이션 개발 전략 및 CI/CD 프로세스와 결합하여 조직 내에서 민첩성, 효율성 및 혁신을 촉진합니다.
- 부서 간 관계를 구축하고 자체 의사 결정에서 다른 AWS CAF 관점의 입력을 수렴하여 플랫폼과 그 이면에서 팀의 성공을 보장합니다.

참조 자료

AWS Cloud Adoption Framework(AWS CAF) 리소스:

- eBook
- 오디오북
- 인포그래픽
- AWS CAF for Artificial Intelligence, Machine Learning, and Generative AI
- 비즈니스 관점
- 인력 관점
- 거버넌스 관점
- 운영 관점
- 보안 관점

추가 리소스:

- AWS 아키텍처 센터
- AWS 사례 연구
- AWS 일반 참조
- AWS 용어집
- AWS 지식 센터
- AWS 권장 가이드
- AWS 파트너 솔루션(이전의 Quick Starts)
- AWS 보안 설명서
- AWS Solutions Library
- AWS 교육 및 자격증
- AWS Well-Architected
- AWS 백서 및 가이드
- 시작하기AWS
- Overview of Amazon Web Services

기여자

이 가이드의 기여자는 다음과 같습니다.

- Tony Santiago, Senior Partner Solutions Architect, AWS
- Matias Undurraga, Enterprise Technologist, AWS
- Alex Torres, AWS Senior Solutions Architect
- Michael Rhyndress, Senior DevSecOps Consultant, AWS
- Alex Livingstone, Principal Solutions Architect and CloudOps Specialist, AWS
- Bruce Cooper, Principal SDE, AWS
- Ravinder Thota, Senior Advisory Consultant, AWS
- Sausan Yazji, Senior Practice Manager, AWS
- Paul Duvall, Director, DevSecOps, AWS
- Jeremy Tennant, Principal Cloud Delivery Manager, AWS
- Sneh Shah, Principal Infrastructure Lead, AWS
- Sasa Baskarada, Worldwide Lead, AWS Cloud Adoption Framework, AWS

문서 기록

아래 표에 이 가이드의 주요 변경 사항이 설명되어 있습니다. 향후 업데이트에 대한 알림을 받으려면 RSS 피드를 구독하십시오.

변경 사항	설명	날짜
최초 게시	—	2023년 10월 25일

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.
- 리플랫포밍(리프트 앤드 리셰이프) - 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 숍) - 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.
- 재배치(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 나중으로 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

ABAC

속성 기반 액세스 제어를 참조하세요.

추상화된 서비스

관리형 서비스를 참조하세요.

ACID

원자성, 일관성, 격리성, 내구성을 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 마이그레이션보다 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.

집계 함수

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

AI

인공 지능을 참조하세요.

AIOps

인공 지능 운영을 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 프로세스의 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 가이드를 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 AWS를 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 전환 AWS 하기 위한 효율적이고 효과적인 계획을 개발하는 데 도움이 되는의 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 웹사이트와 AWS CAF 백서를 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 봇입니다.

BCP

비즈니스 연속성 계획을 참조하세요.

동작 그래프

리소스 동작과 시간 경과에 따른 상호 작용에 대한 통합된 대화형 뷰입니다. Amazon Detective에서 동작 그래프를 사용하여 실패한 로그온 시도, 의심스러운 API 직접 호출 및 유사한 작업을 검사할 수 있습니다. 자세한 내용은 Detective 설명서의 Data in a behavior graph를 참조하십시오.

빅 엔디안 시스템

가장 중요한 바이트를 먼저 저장하는 시스템입니다. 엔디안도 참조하세요.

바이너리 분류

바이너리 결과(가능한 두 클래스 중 하나)를 예측하는 프로세스입니다. 예를 들어, ML 모델이 “이 이메일이 스팸인가요, 스팸이 아닌가요?”, '이 제품은 책인가요, 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

맬웨어에 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 없는에 액세스할 수 AWS 계정 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 Framework를 참조하세요.

카나리 배포

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

CCoE

클라우드 혁신 센터를 참조하세요.

CDC

데이터 캡처 변경을 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 전송을 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.

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 게시물을 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 구축을 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 가이드를 참조하세요.

CMDB

구성 관리 데이터베이스를 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 수정 작업 모음입니다. YAML 템플릿을 사용하여 적합성 팩을 AWS 계정 및 리전 또는 조직 전체에 단일 엔터티로 배포할 수 있습니다. 자세한 내용은 AWS Config 설명서의 적합성 팩을 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 이점을 참조하십시오. CD는 지속적 배포를 의미하기도 합니다. 자세한 내용은 지속적 전달(Continuous Delivery)과 지속적인 개발을 참조하십시오.

CV

컴퓨터 비전을 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 분류를 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 언어를 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 제어를 AWS추가하여 리소스를 보호할 수 있습니다. 예를 들어, 심층 방어 접근 방식은 다단계 인증, 네트워크 세분화 및 암호화를 결합할 수 있습니다.

위임된 관리자

에서 AWS Organizations호환되는 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 서비스를 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

환경을 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 제어를 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 스키마에서 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)

재해로 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 Cloud를 참조하세요.

DML

데이터베이스 조작 언어를 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 방법을 참조하십시오.

DR

재해 복구를 참조하세요.

드리프트 감지

기준이 되는 구성과의 편차 추적을 말합니다. 예를 들어 AWS CloudFormation 를 사용하여 시스템 리소스의 드리프트를 감지하거나를 사용하여 AWS Control Tower 거버넌스 요구 사항 준수에 영향을 미칠 수 있는 랜딩 존의 변경 사항을 감지할 수 있습니다.

DVSM

개발 가치 흐름 매핑을 참조하세요.

E

EDA

탐색 데이터 분석을 참조하세요.

EDI

전자 데이터 교환을 참조하세요.

엣지 컴퓨팅

IoT 네트워크의 엣지에서 스마트 디바이스의 컴퓨팅 성능을 개선하는 기술 엣지 컴퓨팅은 클라우드 컴퓨팅에 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 엔드포인트를 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 생성을 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 암호화를 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 가이드를 참조하십시오.

ERP

엔터프라이즈 리소스 계획을 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 스키마의 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 경계를 참조하세요.

기능 브랜치

브랜치를 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 그레디언트와 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 활용

할 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 LLM에 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 프롬프팅도 참조하세요.

FGAC

세분화된 액세스 제어를 참조하세요.

세분화된 액세스 제어(FGAC)

여러 조건을 사용하여 액세스 요청을 허용하거나 거부합니다.

플래시컷 마이그레이션

단계적 접근 방식을 사용하는 대신 변경 데이터 캡처를 통해 지속적 데이터 복제를 사용하여 최단 시간에 데이터를 마이그레이션하는 데이터베이스 마이그레이션 방법입니다. 목표는 가동 중지 시간을 최소화하는 것입니다.

FM

파운데이션 모델을 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.

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 제한을 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 제한을 참조하십시오.

Gitflow 워크플로

하위 환경과 상위 환경이 소스 코드 리포지토리의 서로 다른 브랜치를 사용하는 방식입니다. Gitflow 워크플로는 레거시로 간주되며 트렁크 기반 워크플로는 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 브라운필드라고도 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.

H

HA

고가용성을 참조하세요.

이기종 데이터베이스 마이그레이션

다른 데이터베이스 엔진을 사용하는 대상 데이터베이스로 소스 데이터베이스 마이그레이션(예: Oracle에서 Amazon Aurora로) 이기종 마이그레이션은 일반적으로 리아키텍트 작업의 일부이며 스

키마를 변환하는 것은 복잡한 작업일 수 있습니다.AWS 는 스키마 변환에 도움이 되는 AWS SCT를 제공합니다.

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.

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 리플랫포밍 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 인프라를 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 인터넷을 참조하세요.

변경 불가능한 인프라

기존 인프라를 업데이트, 패치 또는 수정하는 대신 프로덕션 워크로드에 대한 새 인프라를 배포하는 모델입니다. 변경 불가능한 인프라는 변경 가능한 인프라보다 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 Architecture에서는 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 Schwab에서 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 구축을 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 Architecture에서는 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 인터넷을 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 가이드를 참조하십시오.

ITIL

IT 정보 라이브러리를 참조하세요.

ITSM

IT 서비스 관리를 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 설정을 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 제어를 참조하세요.

최소 권한

작업을 수행하는 데 필요한 최소 권한을 부여하는 보안 모범 사례입니다. 자세한 내용은 IAM 설명서의 최소 권한 적용을 참조하십시오.

리프트 앤드 시프트

7R을 참조하세요.

리틀 엔디안 시스템

가장 덜 중요한 바이트를 먼저 저장하는 시스템입니다. 엔디안도 참조하세요.

LLM

대규모 언어 모델을 참조하세요.

하위 환경

환경을 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 학습을 참조하십시오.

기본 브랜치

브랜치를 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 서비스 는 인프라 계층, 운영 체제 및 플랫폼을 AWS 운영하고, 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 Program을 참조하세요.

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 구축을 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 시스템을 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 마이크로서비스를 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 전략의 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 가이드와 마이그레이션 팩토리에 대한 설명을 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 가이드를 참조하십시오. MRA는 AWS 마이그레이션 전략의 첫 번째 단계입니다.

마이그레이션 전략

워크로드를 AWS 클라우드로 마이그레이션하는 데 사용되는 접근 방식입니다. 자세한 내용은 이 용어집의 7R 항목과 조직을 동원하여 대규모 마이그레이션 가속화를 참조하세요.

ML

기계 학습을 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 전략을 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 평가를 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 분해를 참조하십시오.

MPA

Migration Portfolio Assessment를 참조하세요.

MQTT

메시지 큐 원격 분석 전송을 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 Framework에서는 변경 불가능한 인프라를 모범 사례로 사용할 것을 권장합니다.

O

OAC

오리진 액세스 제어를 참조하세요.

OAI

오리진 액세스 ID를 참조하세요.

OCM

조직 변경 관리를 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 통합을 참조하세요.

OLA

운영 수준 계약을 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 가이드를 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 organization을 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 가이드를 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 특

정 CloudFront 배포를 통해서만 S3 버킷의 콘텐츠에 액세스할 수 있습니다. 더 세분화되고 향상된 액세스 제어를 제공하는 OAC도 참조하십시오.

ORR

운영 준비 상태 검토를 참조하세요.

OT

운영 기술을 참조하세요.

아웃바운드(송신) VPC

AWS 다중 계정 아키텍처에서 애플리케이션 내에서 시작된 네트워크 연결을 처리하는 VPC입니다. AWS Security Reference Architecture에서는 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 경계를 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 정보를 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 컨트롤러를 참조하세요.

PLM

제품 수명 주기 관리를 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 평가를 참조하십시오.

조건자

보통 `WHERE` 절에 있는 `true` 또는 `false`를 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 controls를 참조하십시오.

보안 주체

작업을 수행하고 리소스에 액세스할 수 AWS 있는의 엔터티입니다. 이 엔터티는 일반적으로 , AWS 계정 IAM 역할 또는 사용자의 루트 사용자입니다. 자세한 내용은 IAM 설명서의 역할 용어 및 개념의 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 작업을 참조하십시오.

선제적 제어

규정 미준수 리소스의 배포를 방지하도록 설계된 보안 제어입니다. 이러한 제어는 리소스를 프로비저닝하기 전에 리소스를 스캔합니다. 리소스가 제어를 준수하지 않으면 프로비저닝되지 않습니다. 자세한 내용은 AWS Control Tower 설명서의 제어 참조 가이드를 참조하고 보안 제어 구현의 사전 예방적 제어를 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

환경을 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 확장하는 데 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 MES에서 마이크로서비스는 다른 마이크로서비스가 구독할 수 있는 채널에 이벤트 메시지를 게시할 수 있습니다. 시스템은 게시 서비스를 변경하지 않고도 새 마이크로서비스를 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 생성을 참조하세요.

랜섬웨어

결제가 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 제어를 참조하세요.

읽기 전용 복제본

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

리아키텍팅

7R을 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

7R을 참조하세요.

리전

지리적 영역의 AWS 리소스 모음입니다. 각 AWS 리전 는 내결함성, 안정성 및 복원력을 제공하기 위해 서로 격리되고 독립적입니다. 자세한 내용은 계정에서 사용할 수 있는 AWS 리전 지정을 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

7R을 참조하세요.

릴리스

배포 프로세스에서 변경 사항을 프로덕션 환경으로 승격시키는 행위입니다.

재배치

7R을 참조하세요.

리플랫포밍

7R을 참조하세요.

재구매

7R을 참조하세요.

복원력

중단에 저항하거나 중단을 복구할 수 있는 애플리케이션의 기능입니다. 고가용성 및 재해 복구는 AWS 클라우드에서 복원력을 계획할 때 일반적인 고려 사항입니다. 자세한 내용은 AWS 클라우드 복원력을 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 제어를 참조하세요.

retain

7R을 참조하세요.

사용 중지

7R을 참조하세요.

검색 증강 세대(RAG)

응답을 생성하기 전에 LLM이 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 대한 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 암호를 주기적으로 업데이트하는 프로세스입니다.

행 및 열 액세스 제어(RCAC)

액세스 규칙이 정의된 기본적이고 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 사용자에 대해 IAM에서 사용자를 생성하지 않고도에 로그인하거나 AWS API 작업을 호출할 수 있습니다. SAML 2.0 기반 페더레이션에 대한 자세한 내용은 IAM 설명서의 SAML 2.0 기반 페더레이션 정보를 참조하십시오.

SCADA

감독 제어 및 데이터 획득을 참조하세요.

SCP

서비스 제어 정책을 참조하세요.

보안 암호

에는 암호 또는 사용자 자격 증명과 같이 암호화된 형식으로 저장하는 AWS Secrets Manager기밀 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, 선제적과 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

대상에서 데이터를 수신하는 AWS 서비스 에 의한 데이터 암호화.

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 작업을 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 엔드포인트를 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 지표로 측정되는 서비스의 상태를 나타내는 목표 지표입니다.

공동 책임 모델

클라우드 보안 및 규정 준수를 AWS 위해와 공유하는 책임을 설명하는 모델입니다. AWS 는 클라우드의 보안을 담당하는 반면, 사용자는 클라우드의 보안을 담당합니다. 자세한 내용은 공동 책임 모델을 참조하십시오.

SIEM

보안 정보 및 이벤트 관리 시스템을 참조하세요.

단일 장애점(SPOF)

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

SLA

서비스 수준 계약을 참조하세요.

SLI

서비스 수준 지표를 참조하세요.

SLO

서비스 수준 목표를 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 방식을 참조하세요.

SPOF

단일 장애점을 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 웨어하우스에서 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 Fowler가 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 방법을 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 Synthetics를 사용하여 이러한 테스트를 생성할 수 있습니다.

시스템 프롬프트

LLM에 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 지정을 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

환경을 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.

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 사용을 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. 자세한 내용은 Quantifying uncertainty in deep learning systems 가이드를 참조하십시오.

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

환경을 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 Framework를 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 불가능한 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.