운영 우수성 원칙

AWS Well-Architected 프레임워크

2020 년 7 월

This paper has been archived.

The latest version is now available at:

https://docs.aws.amazon.com/ko_kr/wellarchitected/latest/operational-excellence-pillar/welcome.html

고지 사항

고객에게는 본 문서에 포함된 정보를 독자적으로 평가할 책임이 있습니다. 본 문서는 (a) 정보 제공만을 위한 것이며, (b)사전 고지 없이 변경될 수 있는 현재의 AWS 제품 제공 서비스 및 사례를 보여 주며, (c) AWS 및 자회사, 공급업체 또는 라이선스 제공자로부터 어떠한 약정 또는 보증도 하지 않습니다. AWS 제품 또는 서비스는 명시적이든 묵시적이든 어떠한 종류의 보증, 진술 또는 조건 없이 "있는 그대로" 제공됩니다. 고객에 대한 AWS 의 책임과 법적 책임은 AWS 계약서에 준하며 본 문서는 AWS 와 고객 간의 계약에 포함되지 않고 계약을 변경하지도 않습니다.

목차

소개 1

운영 우수성 1

설계 원칙 1

정의 2

조직 2

조직 우선순위 3

운영 모델 6

조직 문화 14

준비 18

원격 측정 설계 18

운영을 고려한 설계 21

배포 위험 완화 24

운영 준비 27

운영 30

워크로드 상태 파악 q 30

운영 상태 파악 33

이벤트에 응답 36

개선 39

학습, 공유 및 개선 39

결론 42

기고자 42

추가 자료 43

문서 개정 43

개요

AWS Well-Architected 프레임워크의 운영 우수성 원칙에 대해 중점적으로 설명하는 이 백서에서는 AWS 워크로드 설계, 제공 및 유지 관리에 모범 사례를 적용할 때 참조할 수 있는 지침을 제공합니다.

소개

AWS Well-Architected 프레임워크는 AWS 에서 시스템을 구축할 때 내리는 의사 결정의 이점과 위험성을 이해하는 데 도움이 됩니다. 이 프레임워크를 사용하면 클라우드에서 안정적이고 안전하며 효율적이고 경제적인 워크로드를 운영하고 설계할 수 있는 운영 및 아키텍처의 모범 사례를 알아볼 수 있습니다. 그리고 모범 사례와 비교하여 운영 상태와 아키텍처를 일관적으로 측정하고 개선할 영역을 식별할 수 있습니다. 운영을 염두에 두고 Well-Architected 워크로드를 제대로 설계하면 비즈니스 성공 가능성을 높일 수 있습니다.

이 프레임워크는 다음 5 가지 원칙을 기초로 합니다.

- 운영 우수성
- 보안
- 안정성
- 성능 효율성
- 비용 최적화

이 백서에서는 운영 우수성 원칙을 소개하고 Well-Architected 솔루션을 토대로 해당 원칙을 적용하는 방법에 대해 중점적으로 설명합니다. 운영이 지원 대상 사업부 및 개발 팀과 분리된 고유한 기능으로 인식되는 환경에서는 운영 우수성을 달성하기가 어렵습니다. 이 백서의 사례를 도입하면 상태에 대한 인사이트를 제공하는 아키텍처를 구축할 수 있습니다. 또한 효과적이고 효율적인 운영 및 이벤트 대응을 지원하며, 비즈니스 목표를 지속적으로 개선하고 지원할 수 있습니다.

이 문서는 CTO(최고 기술 책임자), 아키텍트, 개발자, 운영 팀 팀원 등 기술 업무 담당자를 위해 작성되었습니다. 이 백서의 내용을 읽고 나면 운영 우수성을 위한 클라우드 아키텍처를 설계할 때 사용할 AWS 모범 사례와 전략을 파악할 수 있습니다. 이 백서에는 구현 세부 정보나 아키텍처 패턴은 제공되지 않습니다. 하지만 이 정보를 확인할 수 있는 적절한 리소스에 대한 참조는 포함되어 있습니다.

운영 우수성

운영 우수성 원칙에는 조직이 비즈니스 목표를 달성하는 방법과 효과적으로 워크로드를 실행하고, 운영에 대한 인사이트를 얻으며, 지원 대상 프로세스와 절차를 지속적으로 개선시켜 비즈니스 가치를 창출할 수 있는 역량이 포함됩니다.

설계 원칙

클라우드의 운영 우수성에는 5 가지 설계 원칙이 있습니다.

- **코드를 통한 운영**: 애플리케이션 코드에 사용되었던 동일한 엔지니어링 원칙을 클라우드의 전체 환경에 적용할 수 있습니다. 그러면 전체 워크로드(애플리케이션, 인프라 등)를 코드로 정의하고 코드로 업데이트할 수 있습니다. 운영 절차를 스크립트로 작성하고 이벤트에 응답하여 스크립트를 트리거하면 실행을 자동화할 수 있습니다. 운영을 코드로 수행하여 인적 오류를 방지하고 이벤트에 대한 지속적인 응답을 활성화합니다.
- **되돌리기 용이한 작은 단위의 변경 내용을 자주 적용**: 주기적으로 구성 요소 업데이트가 가능하도록 워크로드를 설계하여 워크로드에 유익한 변경 사항이 지속적으로 적용되게 합니다. 운영 환경에서 발생한 문제를 식별하고 해결하는 데 어려움이 있는 경우, 가능한 한 고객에게 영향을 주지 않고 되돌릴 수 있을 정도의 작은 단위로 변경 내용을 적용합니다.
- **수시로 운영 절차 수정**: 사용 중인 운영 절차에 개선할 여지가 있는지 확인합니다. 워크로드가 개선되면 절차도 적절하게 개선합니다. 정기적인 게임 데이를 준비하여 모든 절차가 효과가 있는지, 이러한 절차에 팀원들이 익숙한지 검토하고 검증합니다.
- **실패 예측**: "사전 분석(pre-mortem)" 연습을 수행하여 잠재적인 실패의 원인을 찾아내 이를 없애거나 완화할 수 있도록 합니다. 실패 시나리오를 테스트하고 그에 따른 영향을 이해했는지 여부를 검증합니다. 응답 절차를 테스트하여 효과가 있는지, 팀이 이 실행 단계에 익숙한지 확인합니다. 정기 게임 데이를 준비하여 시뮬레이션된 이벤트에 대한 팀의 응답 및 워크로드를 테스트합니다.
- **모든 운영 실패로부터 학습**: 모든 운영 이벤트 및 실패로부터 파악한 내용을 바탕으로 개선합니다. 팀 전반 및 전체 조직에 걸쳐 파악한 내용을 공유합니다.

정의

클라우드의 운영 우수성 원칙에는 다음의 네 가지 영역이 포함됩니다.

- 조직
- 준비
- 운영
- 개선

조직의 경영진이 비즈니스 목표를 정합니다. 조직은 요구 사항과 우선순위를 파악하고, 이를 통해 비즈니스 성과를 실현할 수 있도록 업무를 구성하고 수행해야 합니다. 또한 워크로드에서 이를 지원하는 데 필요한 정보를 생성해야 합니다. 워크로드의 통합, 배포 및 제공하는 서비스를 구현하고 반복적인 프로세스로 자동화하면 프로덕션 환경에 유익한 변경 사항을 지속적으로 더 많이 적용할 수 있습니다.

워크로드 운영에 위험이 내재되었을 수 있습니다. 이러한 위험을 파악하고 정보에 근거하여 프로덕션 환경에 적용할지 여부를 결정해야 합니다. 그리고 팀에서 워크로드를 지원할 수 있어야 합니다. 원하는 비즈니스 성과에서 도출된 비즈니스 및 운영 지표를 통해 워크로드 상태, 운영 활동, 인시던트에 대한 대응 능력을 파악할 수 있습니다. 우선순위는 비즈니스 요구 사항과 비즈니스 환경 변화에 따라 달라집니다. 이를 피드백 루프로 활용하여 조직과 워크로드 운영을 지속적으로 개선합니다.

조직

조직의 우선순위, 조직 구조, 그리고 팀원들이 비즈니스 성과를 뒷받침할 수 있도록 조직에서 팀원을 지원하는 방식을 파악해야 합니다.

운영 우수성을 실현하려면 다음을 파악해야 합니다.

- 조직 우선순위
- 운영 모델
- 조직 문화

조직 우선순위

업무를 적절하게 수행하는 기준이 되는 우선순위를 설정하려면 팀이 전체 워크로드, 워크로드 내 각 팀원의 역할, 그리고 공동의 업무 목표를 파악해야 합니다. 우선순위를 잘 정하면 운영 개선 작업의 이점을 극대화할 수 있습니다. 주기적으로 우선순위를 검토하여 요구 사항의 변화에 따라 우선순위를 업데이트합니다.

외부 고객 요구 사항 평가: 사업 팀, 개발 팀, 운영 팀 등의 주요 이해관계자와 함께 외부 고객 요구 사항 충족을 위해 주력할 영역을 결정합니다.

내부 고객 요구 사항 평가: 사업 팀, 개발 팀, 운영 팀 등의 주요 이해관계자와 함께 내부 고객 요구 사항 충족을 위해 주력할 영역을 결정합니다.

고객 요구 사항을 평가하면 비즈니스 성과를 달성하기 위해 어떤 지원이 필요한지 철저하게 파악할 수 있습니다.

설정한 우선순위를 활용해 가장 영향력이 큰 개선 작업 부분부터 중점적으로 수행합니다. 팀 기술 개발, 워크로드 성능 개선, 비용 절감, 런북 자동화, 모니터링 기능 향상 등을 예로 들 수 있습니다. 요구 사항이 바뀌면 우선순위를 업데이트합니다.

거버넌스 요구 사항 평가: 조직에서 특정 사항을 준수하거나 강조하기 위해 정의한 지침이나 의무 사항을 알고 있어야 합니다. 조직 정책, 표준 및 요구 사항과 같은 내부 요인을 평가합니다. 거버넌스에 대한 변경 사항을 식별할 수 있는 메커니즘이 있는지 확인합니다. 거버넌스 요구 사항이 식별되지 않는 것으로 결론을 내릴 때에는 신중하게 판단하여 내린 결론인지 재차 확인해야 합니다.

외부 규정 준수 요구 사항 평가: 특정 사항을 규정하거나 강조하는 지침 또는 의무 사항을 파악해야 합니다. 규정 준수 요구 사항 및 산업 표준과 같은 외부 요인을 평가합니다. 규정 준수 요구 사항의 변경 내용을 식별할 수 있는 메커니즘이 있는지 확인합니다. 규정 준수 요구 사항이 식별되지 않는 것으로 결론을 내릴 때에는 신중하게 판단하여 내린 결론인지 재차 확인해야 합니다.

만약 조직에 적용되는 외부 규제 또는 규정 준수 요구 사항이 있다면, 팀원이 우선순위에 대한 영향도를 확인할 수 있도록 AWS 클라우드 규정 준수에서 제공하는 리소스를 활용하여 관련 정보를 제공해야 합니다.

위협 환경 평가: 비즈니스에 대한 위협 요소(예: 경쟁, 비즈니스상의 위험 및 법적 책임, 운영상의 위험, 정보 보안 위협)를 평가하고 위험 목록에서 최신 정보를 관리합니다. 노력을 집중해야 할 영역을 결정할 때 위험의 영향도를 포함시킵니다.

Well-Architected 프레임워크에서는 학습, 평가 및 개선을 강조합니다. 아키텍처를 평가하고 시간에 따라 확장 가능한 설계를 구현할 수 있는 일관된 접근 방식을 제공합니다. AWS 에서 제공하는 AWS Well-Architected Tool 은 개발 전의 접근 방식, 프로덕션 환경에 적용하기 전의 워크로드 상태, 프로덕션 환경에서의 워크로드 상태를 검토합니다. 이를 최신 AWS 아키텍처 모범 사례와 비교하고, 워크로드의 전반적인 상태를 모니터링하며, 잠재적 위험에 대한 분석 정보를 얻을 수 있습니다.

Enterprise Support 고객은 AWS 모범 사례를 기준으로 하여 아키텍처를 평가할 수 있는 미션 크리티컬 워크로드에 대한 안내형 Well-Architected Review 서비스를 이용할 수 있습니다.

또한 클라우드 운영 방식의 차이를 파악하는 데 사용할 수 있는 Operations Review 를 이용할 수도 있습니다.

여러 팀의 구성원이 이러한 검토에 참여하면 팀에서 각 역할을 맡은 구성원이 효율적인 워크로드 처리에 기여할 수 있는 방법을 공통된 방식으로 파악할 수 있습니다. 검토를 통해 확인된 요구 사항을 참조하여 우선순위를 결정할 수 있습니다.

AWS Trusted Advisor 는 우선순위 결정에 도움이 될 수 있는 최적화 방안을 알려주는 핵심 점검 세트에 접근할 수 있는 도구입니다. Business 및 Enterprise Support 고객에게는 우선순위를 더욱 정확히 결정하는 데 사용할 수 있는 추가 검사 기능이 제공됩니다. 이러한 기능을 사용하면 보안, 안정성, 성능 및 비용 최적화 영역을 중점적으로 확인할 수 있습니다.

트레이드 오프(절충안) 평가: 운영 역량을 집중할 영역을 결정하거나 또는 수행할 조치를 선택할 때 정보를 토대로 결정을 내릴 수 있도록 충돌하는 이해 관계나 대안 사이의 장단점을 평가합니다. 예를 들어, 비용 최적화보다 새로운 기능의 시장 출시를 앞당기는 데 더 역점을 둘 수 있습니다. 아니면 데이터 유형에 최적화된 데이터베이스로 마이그레이션하고 애플리케이션을 업데이트하는 대신, 시스템 마이그레이션 작업을 간소화하기 위해 비관계형 데이터 솔루션으로 관계형 데이터베이스를 선택할 수도 있습니다.

AWS 를 활용하면 선택한 방식이 워크로드에 어떤 영향을 주는지를 더 자세히 파악할 수 있도록 팀에게 AWS 및 해당 서비스 관련 정보를 제공할 수 있습니다. AWS Support 에서 제공하는 리소스(AWS Knowledge Center, AWS Discussion Forms 및 AWS 지원 센터) 및 AWS 설명서를 사용하여 팀을 교육시켜야 합니다. AWS 관련 문의 사항에 대해 지원을 받으려면 AWS 지원 센터를 통해 AWS Support 에 문의하십시오.

또한 AWS 는 AWS 의 운영을 통해 학습한 모범 사례와 패턴을 Amazon Builders' Library 에서 공유합니다. AWS 블로그 및 공식 AWS 팟캐스트에서도 기타 여러 가지 유용한 정보를 확인할 수 있습니다.

이점 및 위험 관리: 역량을 집중할 영역을 결정할 때 정보를 토대로 적절한 결정을 내릴 수 있도록 이점과 위험을 관리합니다. 예를 들어 새 기능을 고객에게 제공하기 위해 해결되지 않은 문제가 남아 있는 워크로드를 배포하는 것이 이득일 수 있습니다. 관련 위험을 완화할 수도 있거나, 위험을 감수할 수 없는 경우에는 위험을 해결하기 위한 조치를 취해야 합니다.

특정 시점에 우선순위 중 일부를 중점적으로 처리해야 할 수도 있습니다. 필요한 기능을 개발하고 위험을 관리할 수 있도록 장기적으로 균형 잡힌 접근 방식을 사용하십시오. 우선순위를 정기적으로 검토하고 요구 사항이 바뀌면 우선순위를 업데이트합니다.

리소스

조직의 우선순위와 관련한 AWS 모범 사례에 대해 자세히 알아보려면 다음 리소스를 참조하십시오.

설명서

- AWS Trusted Advisor
- AWS Cloud Compliance
- AWS Well-Architected 프레임워크
- AWS Business Support
- AWS Enterprise Support
- AWS Enterprise Support Entitlements
- AWS Support Cloud Operations Reviews
- AWS 클라우드 채택 프레임워크

운영 모델

팀은 비즈니스 성과를 달성하기 위해 맡은 역할을 이해해야 합니다. 그리고 다른 팀의 성공을 위해 자신의 팀이 해야 할 역할과 해당 팀이 해야 할 역할을 파악하고, 목표를 공유해야 합니다. 맡은 의무, 책임, 의사 결정 방식 및 의사 결정권자를 파악하면 역량을 집중하고, 팀의 이점을 극대화할 수 있습니다.

팀의 요구 사항은 종사하는 업계, 소속된 조직, 팀 구성, 그리고 워크로드의 특성에 따라 결정됩니다. 당연히 단일 운영 모델로는 모든 팀과 해당 워크로드를 지원할 수 없습니다.

조직에 존재하는 운영 모델의 수는 개발 팀의 수에 따라 늘어날 수 있습니다. 여러 운영 모델을 조합하여 사용해야 할 수도 있습니다.

표준을 채택하고 서비스를 사용하면 운영 모델에서 지원 부담을 줄이고 운영을 간소화할 수 있습니다. 공유 표준을 개발하면 표준을 이미 채택했고 새로운 기능을 채택하려는 팀이 많을수록 그 효과가 더 커집니다.

팀의 활동을 지원하는 데 있어서는 표준의 추가, 변경 및 예외 처리를 요청하는 메커니즘을 갖추어야 합니다. 이 옵션이 없으면 표준이 혁신의 제약 요인이 됩니다. 이점과 위험을 평가한 후 요청이 적절한지 판단하고 실현 가능한 경우에 요청을 승인해야 합니다.

책임을 잘 정의하면 상충하거나 중복되는 작업의 빈도를 줄일 수 있습니다. 사업, 개발 및 운영 팀 간에 강력한 연계와 관계를 쌓으면 비즈니스 성과를 달성하기가 더 쉽습니다.

운영 모델 2x2 표현

이러한 운영 모델 2x2 표현을 살펴보면 조직의 환경 내에서 팀 간 관계를 이해하는 데 도움이 됩니다. 이 그림은 각 팀이 담당하는 역할과 팀 간 관계를 중점적으로 다루지만, 이 예제의 맥락에서 거버넌스와 의사 결정에 대해서도 설명합니다.

팀은 지원하는 워크로드에 따라 다양한 모델의 여러 부분에서 책임을 맡을 수 있습니다. 설명된 상위 영역보다 더 전문적인 분야로 세분화하기를 원할 수 있습니다. 활동을 분리 또는 합치하거나, 또는 팀을 추가하고 더 구체적인 세부 정보를 제공하면서 이러한 모델을 무한하게 변형시킬 수 있습니다.

여러 팀 간에 중복된 부분을 확인할 수 있거나, 추가적인 이점을 제공 또는 효율성을 높일 수 있지만 아직 알려지지 않은 역량이 있음을 깨달을 수 있습니다. 또한 조직에서 충족되지 않은 요구 사항을 확인하고 이를 해결하려고 계획할 수도 있습니다.

조직의 변화를 평가할 때는 모델 간의 장단점은 무엇인지, 변화 전후의 모델에서 개별 팀은 어떤 역할을 맡는지, 팀의 관계와 책임이 어떻게 바뀌는지, 그리고 변화에 따른 이점이 조직에 영향을 미치는지 여부를 조사합니다.

다음과 같은 네 가지 각각의 운영 모델을 사용하면 성공적으로 운영할 수 있습니다. 일부 모델은 개발 과정의 특정 시점이나 특정 사용 사례에 더 적합할 수 있습니다. 이러한 모델 중 일부는 현재 조직 환경에서 사용하는 모델보다 더 유용할 수 있습니다.

- 완전분리형 운영 모델
- 중앙 집중식 거버넌스에서 분리된 AEO(애플리케이션 엔지니어링 및 운영) 및 IEO(인프라 엔지니어링 및 운영)
- 중앙 집중식 거버넌스 및 서비스 공급자와 분리된 AEO 및 IEO
- 분산형 거버넌스와 분리된 AEO 및 IEO

완전분리형 운영 모델

다음 그림의 세로 축에는 애플리케이션과 인프라가 있습니다. 애플리케이션은 비즈니스 성과를 지원하는 워크로드를 말하며, 맞춤형으로 개발하거나 구매한 소프트웨어일 수 있습니다. 인프라는 물리적 인프라와 및 가상 인프라, 그리고 해당 워크로드를 지원하는 기타 소프트웨어를 말합니다.

가로 축에는 엔지니어링과 운영이 있습니다. 엔지니어링은 애플리케이션 및 인프라의 개발, 구축 및 테스트를 말합니다. 운영은 애플리케이션 및 인프라의 배포, 업데이트 및 지속적 지원을 말합니다.

많은 조직에 이러한 "완전분리형" 모델이 존재합니다. 각 사분면의 활동은 별도의 팀이 수행합니다. 작업 요청, 작업 대기열 및 티켓과 같은 메커니즘 또는 ITSM(IT 서비스 관리 시스템)을 사용하여 팀 간에 작업이 전달됩니다.

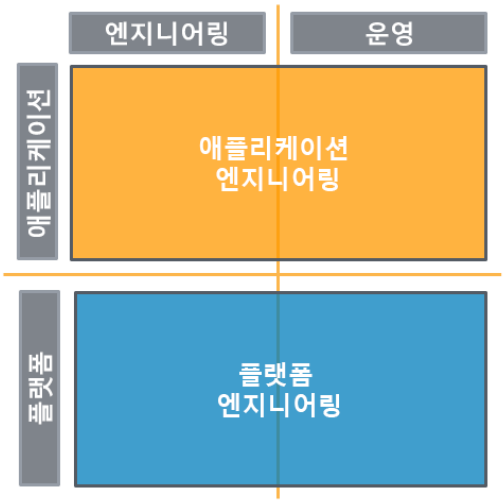
다른 팀으로 또는 여러 팀 간에 작업을 이양하면 복잡해지고 병목 현상과 지연이 발생합니다. 요청은 해당 우선순위에 이를 때까지 지연될 수 있습니다. 결함이 뒤늦게 발견되면 상당한 재작업이 필요하며, 동일한 팀과 부서의 검사를 다시 거쳐야 할 수도 있습니다. 엔지니어링 팀의 조치가 필요한 인시던트가 있다면 이미 착수된 활동으로 인해 대응이 지연됩니다.

수행 중인 활동이나 기능을 중심으로 비즈니스, 개발 및 운영 팀을 구성하면 서로 조율되지 않을 위험이 큽니다. 이로 인해 팀이 사업 성과가 아니라, 맡은 책임에만 집중하게 될 수 있습니다. 팀의 전문 분야가 협소하거나, 팀이 물리적으로 또는 논리적으로 격리되어 있으면, 커뮤니케이션과 협업을 저해할 수 있습니다.

중앙 집중식 거버넌스와 분리된 AEO 및 IEO

이 "분리된 AEO 및 IEO" 모델은 "자체적으로 구축하고 실행"하는 방법론을 따릅니다.

조직의 애플리케이션 엔지니어와 개발자가 워크로드의 운영과 엔지니어링을 모두 수행합니다. 마찬가지로, 인프라 엔지니어는 애플리케이션 팀을 지원하는 데 사용하는 플랫폼의 운영과 엔지니어링을 모두 수행합니다.



이 예제에서는 거버넌스를 중앙 집중식으로 처리하려고 합니다. 표준은 여러 애플리케이션 팀에 제공되거나 분산되거나 공유됩니다.

AWS Organizations 와 같이 여러 계정에 걸쳐 환경을 중앙 집중식으로 관리할 수 있는 도구나 서비스를 사용해야 합니다. AWS Control Tower 와 같은 서비스는 계정 설정을 위한 청사진(운영 모델 지원)를 정의하고, AWS Organizations 를 통해 지속적으로 거버넌스를 적용하며, 새로운 계정의 프로비저닝을 자동화할 수 있도록 관리 기능을 확장합니다.

"자체적으로 구축하고 실행"한다고 해서 애플리케이션 팀이 전체 스택, 도구 체인 및 플랫폼을 책임진다는 의미는 아닙니다.

플랫폼 엔지니어링 팀이 표준화된 서비스 세트(예: 개발 도구, 모니터링 도구, 백업 및 복구 도구, 네트워크)를 애플리케이션 팀에 제공합니다. 플랫폼 팀은 승인된 클라우드 공급자 서비스, 동일한 서비스의 특정 구성 또는 둘 모두에 대한 접근 권한을 애플리케이션 팀에 제공할 수도 있습니다.

AWS Service Catalog 와 같은 승인된 서비스 및 구성을 배포하기 위한 셀프 서비스 기능을 제공하는 메커니즘을 활용하면, 거버넌스를 적용하면서 이행 요청으로 인한 지연을 최소화할 수 있습니다.

플랫폼 팀은 전체 스택의 가시성을 지원하므로, 이를 통해 애플리케이션 팀은 애플리케이션 구성 요소에서 발생한 문제와 애플리케이션에 사용되는 서비스 및 인프라 구성 요소에서 발생한 문제를

서로 구분할 수 있습니다. 또한 플랫폼 팀은 이러한 서비스 구성을 지원하고 애플리케이션 팀의 운영을 개선하는 방법에 대한 지침을 제공할 수 있습니다.

앞서 언급한 바와 같이, 애플리케이션 팀이 팀의 활동과 애플리케이션 혁신을 지원하는 데 있어서 표준의 추가, 변경 및 예외 처리를 요청할 수 있는 메커니즘이 갖추어져야 합니다.

분리된 AEO IEO 모델은 애플리케이션 팀에 강력한 피드백 루프를 제공합니다. 일상적인 워크로드 운영에서는 지원 및 기능 요청을 통한 직접적 또는 간접적인 상호 작용으로 인해 고객과의 접촉이 늘어나기 마련입니다. 이와 같이 가시성이 높아지는 덕분에 애플리케이션 팀은 문제를 보다 신속하게 해결할 수 있습니다. 더 긴밀해지는 고객의 참여와 고객과의 관계는 고객 요구 사항에 대한 인사이트를 얻고 더 빠르게 혁신할 수 있게 해줍니다.

애플리케이션 팀을 지원하는 플랫폼 팀의 경우도 마찬가지입니다.

채택된 표준이 사전에 사전 승인을 받게 되므로, 프로덕션 환경에 적용하는 데 필요한 검토 작업량을 줄일 수 있습니다. 플랫폼 팀에서 제공하는 테스트를 거쳐 지원되는 표준을 사용하면 이러한 서비스 관련 문제의 발생 빈도가 줄어들 수 있습니다. 표준을 채택하면 애플리케이션 팀이 워크로드를 차별화하는 데 집중할 수 있습니다.

중앙 집중식 거버넌스 및 서비스 공급자와 분리된 AEO 및 IEO

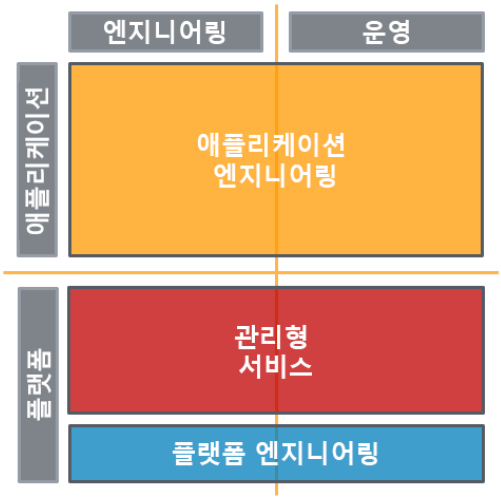
이 "분리된 AEO 및 IEO" 모델은 "자체적으로 구축하고 실행"하는 방법론을 따릅니다.

조직의 애플리케이션 엔지니어와 개발자가 워크로드의 운영과 엔지니어링을 모두 수행합니다.

조직에 전담 플랫폼 엔지니어링 및 운영 팀을 지원할 기술이나 팀원이 갖추어져 있지 않거나, 여기에 시간과 노력을 투자하고 싶지 않을 수도 있습니다.

또는 비즈니스를 차별화하는 기능을 제작하는 데 초점을 맞춘 플랫폼 팀을 두고 싶지만, 획일적인 일상 업무를 아웃소싱 업체에 맡기고 싶을 수도 있습니다.

Managed Services 공급자(예: AWS Managed Services, AWS Managed Services 파트너 또는 AWS Partner Network 의 Managed Services 공급자)는 클라우드 환경을 구현하는 전문성을 제공하며, 보안 및 규정 준수 요구 사항과 비즈니스 목표를 지원합니다.



이 모델의 경우 AWS Organizations 및 AWS Control Tower 에서 계정 생성과 정책을 관리하고, 플랫폼 팀은 중앙 집중식으로 거버넌스를 관리하도록 구현하려고 합니다.

이 모델에서는 서비스 공급자의 업무에 맞추어 메커니즘을 수정해야 합니다. 이 모델에서는 서비스 공급자를 비롯한 팀 간의 작업 이양으로 인해 발생하는 병목 현상과 지연, 또는 뒤늦은 결함 발견과 관련된 잠재적 재작업 문제가 해결되지 않습니다.

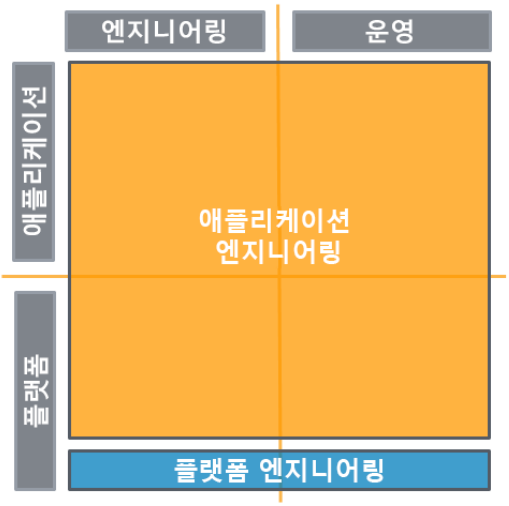
공급자의 표준, 모범 사례, 프로세스 및 전문성을 활용하여 이 문제를 해결할 수 있습니다. 또한 이들이 지속적으로 개발하는 서비스 오퍼링의 이점을 얻을 수 있습니다.

Managed Services 를 운영 모델에 추가하면 시간과 리소스를 투자할 필요 없이, 시간과 리소스를 절약하고, 팀은 새로운 기술과 성능을 개발하는 대신 팀 내부를 간소화하고 비즈니스를 차별화하는 전략적 성과에 집중할 수 있습니다.

분산형 거버넌스와 분리된 AEO 및 IEO

이 "분리된 AEO 및 IEO" 모델은 "자체적으로 구축하고 실행"하는 방법론을 따릅니다.

조직의 애플리케이션 엔지니어와 개발자가 워크로드의 운영과 엔지니어링을 모두 수행합니다. 마찬가지로, 인프라 엔지니어는 애플리케이션 팀을 지원하는 데 사용하는 플랫폼의 운영과 엔지니어링을 모두 수행합니다.



이 예제에서는 거버넌스를 분산형으로 처리하려고 합니다.

표준은 여전히 플랫폼 팀에 의해 애플리케이션 팀에 배포, 제공 또는 공유되지만, 애플리케이션 팀이 워크로드를 지원하는 데 있어서 새로운 플랫폼 기능을 자유롭게 운영하고 엔지니어링할 수 있습니다.

이 모델에서는 애플리케이션 팀의 제약이 더 적지만, 책임은 더 늘어납니다. 플랫폼 기능을 추가로 지원하려면 추가적인 기술이 필요하고 잠재적으로 팀원이 더 필요할 수도 있습니다. 기술 세트가 적절하지 않고 결함이 조기에 파악되지 않으면 재작업 발생의 위험이 상당히 커집니다.

애플리케이션 팀에 명시적으로 위임되지 않은 정책을 적용해야 합니다. AWS Organizations 와 같이 여러 계정에 걸쳐 환경을 중앙 집중식으로 관리할 수 있는 도구나 서비스를 사용합니다. AWS Control Tower 와 같은 서비스는 계정 설정을 위한 청사진(운영 모델 지원)을 정의하고, AWS Organizations 를 통해 지속적으로 거버넌스를 적용하며, 새로운 계정의 프로비저닝을 자동화할 수 있도록 관리 기능을 확장합니다.

애플리케이션 팀이 표준에의 추가 및 변경을 요청할 수 있는 메커니즘을 갖추는 것이 좋습니다. 애플리케이션 팀이 새로운 표준을 만드는 데에도 기여할 수 있으며, 이는 다른 애플리케이션 팀에 도움이 될 수 있습니다. 플랫폼 팀은 이러한 추가 기능에 대한 직접적인 지원을 제공하는 것이 비즈니스 성과를 효과적으로 지원하는 방법이라고 판단할 수 있습니다.

이 모델은 기술 및 팀원에 대한 요구 사항이 상당히 높으므로 혁신에 있어서 제약이 덜합니다. 팀 간의 작업 이양으로 인해 발생하는 많은 병목 현상과 지연을 해결하는 동시에, 팀과 고객 간 관계 구축을 효과적으로 촉진할 수 있습니다.

관계 및 소유권

운영 모델은 팀 간의 관계를 정의하고 식별 가능한 소유권과 책임을 뒷받침합니다.

리소스 책임자 식별: 각 애플리케이션, 워크로드, 플랫폼 및 인프라 구성 요소를 소유한 사람, 해당 구성 요소가 제공하는 비즈니스 가치, 그리고 소유권이 존재하는 이유를 파악합니다. 이러한 개별 구성 요소의 비즈니스 가치를 파악하고 이러한 구성 요소가 비즈니스 성과를 어떻게 지원하는지 이해하면 해당 구성 요소에 적용되는 프로세스와 절차를 알 수 있습니다.

프로세스 및 절차의 책임자 식별: 개별 프로세스와 절차의 정의를 담당하는 사람, 그러한 특정 프로세스와 절차가 사용되는 이유, 그리고 소유권이 존재하는 이유를 파악합니다. 특정 프로세스 및 절차가 사용되는 이유를 이해하면 개선 기회를 파악할 수 있습니다.

운영 활동의 성과에 대한 책임이 있는 책임자 식별: 정의된 워크로드에 대해 특정 활동을 수행할 책임이 있는 사람과 그러한 책임이 존재하는 이유를 파악합니다. 운영 작업의 성과에 대한 책임 소재를 파악하면 누가 작업을 수행하고, 누가 결과를 검증하며, 누가 작업 책임자에게 피드백을 제공하게 되는지 알 수 있습니다.

팀원이 담당해야 하는 업무 파악: 역할을 이해하면 작업의 우선순위를 알 수 있습니다. 이를 통해 팀원은 요구 사항을 인식하고 적절하게 대응할 수 있습니다.

책임과 소유권을 식별하는 메커니즘: 개인이나 팀을 식별하지 못할 경우, 소유권을 할당하거나 해결해야 할 소유권 문제에 대해 계획을 수립할 권한이 있는 사람에게 정해진 에스컬레이션 경로를 통해 사안을 에스컬레이션할 수 있습니다.

추가, 변경 및 예외 처리를 요청하는 메커니즘: 프로세스, 절차 및 리소스의 책임자에게 요청을 보낼 수 있습니다. 이점과 위험을 평가한 후 요청이 적절한지 판단하고 정보에 입각한 의사 결정을 통해 실현 가능한 경우에 요청을 승인해야 합니다.

미리 정의되었거나 협상된 팀 간 책임: 팀 간에 서로 협력하고 지원하는 방식에 관한 내용을 정의하거나 협상해둡니다(예: 응답 시간, 서비스 수준 목표 또는 서비스 수준 계약). 팀의 작업이

비즈니스 성과에 미치는 영향, 그리고 다른 팀과 조직의 성과에 미치는 영향을 이해하면 작업의 우선순위를 파악할 수 있고 적절하게 대응할 수 있습니다.

책임과 소유권을 정의하지 않았거나 알지 못하는 경우 필요한 활동을 적시에 처리하지 못하게 되며 해당 요구 사항을 해결하기 위한 작업이 중복되고 잠재적으로는 상충될 위험이 있습니다.

리소스

운영 설계와 관련한 AWS 모범 사례에 대해 자세히 알아보려면 다음 리소스를 참조하십시오.

동영상

- AWS re:Invent 2019: [REPEAT 1] How to ensure configuration compliance(MGT303-R1)
- AWS re:Invent 2019: Automate everything: Options and best practices(MGT304)

설명서

- AWS Managed Services
- AWS Organizations 기능
- AWS Control Tower 기능

조직 문화

팀원이 효과적으로 조치를 취하고 비즈니스 성과를 지원할 수 있도록 팀원에 대한 지원을 제공합니다.

경영진의 지원: 최고 경영진은 조직에 대한 기대치를 명확하게 설정하고 성공 여부를 평가합니다. 최고 경영진은 조직이 발전하고 모범 사례를 도입하도록 하는 감독이자 후원자이며 지지자입니다.

성과 실현에 실패할 위험이 있는 경우 조치를 취하도록 팀원 권한 강화: 워크로드 소유자는 성과 실현에 실패할 위험이 있는 경우 문제에 대응할 수 있도록 팀원의 권한을 강화하는 지침과 범위를 정의합니다. 에스컬레이션 메커니즘은 이벤트에서 정의된 범위를 벗어날 경우 수행할 조치를 정할 때 사용됩니다.

에스컬레이션 장려: 성과 실현에 실패할 위험이 있는 경우 의사 결정권자와 이해관계자에게 문제를 에스컬레이션하는 메커니즘이 마련되어 있으며 팀원은 이를 적극적으로 활용해야 합니다. 위험을 식별하고 인시던트를 방지할 수 있도록 에스컬레이션을 조기에, 그리고 자주 수행해야 합니다.

조치 가능한 방식으로 적시에 명확하게 커뮤니케이션: 알려진 위험과 예정된 이벤트를 팀원에게 적시에 알리는 데 사용되는 메커니즘이 마련되어 있습니다. 조치가 필요한지 여부와 어떤 조치가 필요한지 판단하고 적시에 조치를 취할 수 있도록 필요한 컨텍스트, 세부 정보 및 시간(가능한 경우)이 제공됩니다. 예를 들어 소프트웨어 취약성에 대한 알림을 제공하여 패치를 신속하게 적용하도록 하거나, 예정된 판매 프로모션에 대한 알림을 제공하여 서비스 중단의 위험을 막기 위한 변경 방지 기능을 구현하게 할 수 있습니다.

대기 중인 활동을 팀원이 식별할 수 있도록 예정된 이벤트를 변경 일정이나 유지 관리 일정에 기록할 수 있습니다.

AWS 에서 AWS Systems Manager Change Calendar 를 사용하여 이러한 세부 정보를 기록할 수 있습니다. 일정 상태를 프로그래밍 방식으로 검사하여 특정 시점에 활동에 대한 일정이 개시되었는지, 아니면 마감되었는지 확인합니다. 운영 활동은 잠재적으로 중단이 발생될 수 있는 활동을 위해 예약되는 "승인된" 특정 시간대를 중심으로 계획할 수 있습니다. AWS Systems Manager Maintenance Windows 에서는 인스턴스 및 다른 지원되는 리소스에 대한 활동을 예약하여 활동을 자동화하고 해당 활동을 검색할 수 있습니다.

실험 권장: 실험은 학습을 가속화하고 팀원의 참여와 관심을 유지해 줍니다. 원치 않는 결과가 나와도, 성공하지 못하는 경로를 알게 되므로 실험에는 성공한 것입니다. 원치 않는 결과가 나온 성공한 실험에 대해 팀원에게 불이익을 가하지 않습니다. 혁신을 하고 아이디어를 실현하려면 실험이 필요합니다.

기술 세트를 관리하고 개발할 수 있도록 팀원 지원 및 독려: 팀은 기술 세트를 발전시켜 새로운 기술을 채택하고 변화하는 요구 사항을 지원하며 워크로드 운영에 책임감을 갖도록 도와줘야 합니다. 새로운 기술 영역에서 기술을 발전시키면 팀원의 만족도를 높이고 혁신을 뒷받침할 수 있습니다. 발전하는 기술을 검증하고 인증하는 업계 자격증을 획득 및 관리하도록 팀원을 독려합니다. 지식이 효과적으로 전달되도록 하고, 제도적 지식을 갖춘 경험 많고 숙련된 직원을

잃은 경우에 중대한 영향이 발생할 위험을 줄일 수 있도록 교차 교육을 실시합니다. 학습을 위해 체계적으로 정해진 교육 시간을 제공합니다.

AWS 에서는 AWS 시작하기 리소스 센터, AWS 블로그, AWS Online Tech Talks, AWS 이벤트 및 웹 세미나, AWS Well-Architected Labs 등의 다양한 리소스를 제공합니다. 이러한 리소스에서는 팀을 대상으로 교육을 진행하는 데 활용할 수 있는 지침, 예제 및 자세한 연습 과정을 제공합니다.

AWS 는 Amazon Builders' Library 에서 AWS 의 운영을 통해 학습한 모범 사례와 패턴을 공유하며, AWS 블로그 및 공식 AWS 팟캐스트를 통해 도움이 되는 방대한 기타 교육 자료를 제공합니다.

Well-Architected Lab 에서 제공하는 교육 리소스, AWS Support(AWS Knowledge Center, AWS Discussion Forms 및 AWS 지원 센터) 및 AWS 설명서를 사용하여 팀을 교육시켜야 합니다. AWS 관련 문의 사항에 대해 지원을 받으려면 AWS 지원 센터를 통해 AWS Support 에 문의하십시오.

AWS Training and Certification 에서는 AWS 기초에 관한 자습형 디지털 교육 과정을 통해 무료 교육을 제공합니다. 팀이 AWS 기술을 발전시켜 나갈 수 있도록 강의식 교육에 등록할 수도 있습니다.

팀에 적절한 리소스 제공: 팀원의 역량을 관리하고, 도구와 리소스를 제공하여 워크로드 요구 사항을 지원합니다. 과중한 업무를 수행하는 팀원은 인적 오류로 인한 인시던트를 일으킬 위험이 큽니다. 자주 실행하는 활동을 자동화하는 등, 도구 및 리소스에 투자하면 팀의 효율성이 높아져 팀원이 더 많은 활동을 지원할 수 있게 됩니다.

팀 내부 및 여러 팀 간에 의견의 다양성 추구 및 장려: 조직 내의 다양성을 활용하여 독창적이고 다양한 관점을 얻습니다. 이러한 관점을 통해 혁신을 증진하고, 기존의 추정 사항에 의문을 제기하며, 확인 편향의 위험을 줄일 수 있습니다. 팀 내에서 포용성, 다양성 및 접근성을 높여 유익한 관점을 확보합니다.

조직 문화는 팀원의 업무 만족도와 팀원 이직률에 직접적인 영향을 미칩니다. 팀원의 참여와 역량을 통해 비즈니스의 성공을 뒷받침할 수 있습니다.

리소스

운영 설계와 관련한 AWS 모범 사례에 대해 자세히 알아보려면 다음 리소스를 참조하십시오.

동영상

- AWS re:Invent 2019: [REPEAT 1] How to ensure configuration compliance(MGT303-R1)
- AWS re:Invent 2019: Automate everything: Options and best practices(MGT304)

설명서

- AWS Managed Services
- AWS Managed Services 서비스 설명
- AWS Organizations 기능
- AWS Control Tower 기능

준비

운영이 잘 되도록 준비하려면 워크로드 및 예상되는 워크로드 동작을 파악해야 합니다. 그러면 워크로드가 상태 관련 분석 정보를 제공하도록 설계할 수 있으며, 워크로드를 지원하는 절차를 작성할 수 있습니다.

운영 우수성 달성을 준비하기 위해 수행해야 하는 사항은 다음과 같습니다.

- 원격 측정 설계
- 흐름 개선
- 배포 위험 완화
- 운영 준비 상태 파악

원격 측정 설계

문제를 관찰하고 조사할 수 있도록 모든 구성 요소에서 지표, 로그, 이벤트, 추적 등 내부 상태를 파악하는 데 필요한 정보를 제공할 수 있게 워크로드를 설계합니다. 반복을 통해, 워크로드 상태를 모니터링하고, 성과 실현에 실패할 위험이 있는 경우 이를 식별하며, 효과적으로 대응하는 데 필요한 원격 측정을 개발합니다.

AWS 에서 애플리케이션 및 워크로드 구성 요소로부터 로그, 지표 및 이벤트를 생성하고 수집하여 내부 상태를 파악할 수 있습니다. 분산 추적을 통합하여 워크로드에서 요청이 이동할 때 요청을 추적할 수 있습니다. 이 데이터를 사용하여 애플리케이션과 기본 구성 요소가 상호 작용하는 방식을 이해하고 문제와 성과를 분석합니다.

워크로드를 계측할 때 상태를 파악할 수 있는 광범위한 정보 세트를 캡처합니다(예: 상태 변경 사항, 사용자 활동, 접근 권한, 사용률 카운터). 이때 필터를 사용하여 시간 경과에 따라 가장 유용한 정보를 선택할 수 있습니다.

애플리케이션 원격 측정 구현: 내부 상태 및 비즈니스 성과 달성 관련 정보(예: 대기열 길이, 오류 메시지, 응답 시간)를 생성하도록 애플리케이션 코드를 설계합니다. 이 정보를 사용하여 대응이 필요한지 여부를 판단합니다.

EC2 인스턴스 및 물리적 서버에서 Amazon CloudWatch 로 애플리케이션 로그 및 고급 지표를 전송하도록 Unified Amazon CloudWatch Logs Agent 를 설치하고 구성해야 합니다.

AWS CLI 또는 CloudWatch API 를 사용하여 맞춤형 지표를 생성하고 게시합니다. 이때 인사이트를 제공하는 비즈니스 지표와 기술 지표를 모두 게시해야 합니다. 이렇게 하면 고객의 행동을 쉽게 파악할 수 있습니다.

CloudWatch Logs API 를 사용하여 애플리케이션에서 CloudWatch 로 로그를 직접 전송하거나 AWS SDK 및 Amazon EventBridge 를 사용하여 이벤트를 전송할 수 있습니다. AWS Lambda 코드에 logging 문을 삽입하여 CloudWatch Logs 에 자동으로 저장합니다.

워크로드 원격 측정 구현 및 구성: 내부 상태와 현재 상태 관련 정보를 내보내도록 워크로드를 설계 및 구성합니다. API 호출 횟수, HTTP 상태 코드, 확장/축소 이벤트 등의 정보를 내보낼 수 있습니다. 이 정보를 사용하면 응답이 필요한 경우를 확인할 수 있습니다.

Amazon CloudWatch 와 같은 서비스를 사용하여 워크로드 구성 요소(예: AWS CloudTrail 의 API 로그, AWS Lambda 지표, Amazon VPC 흐름 로그 및 기타 서비스)에서 로그와 지표를 집계합니다.

사용자 활동 원격 측정 구현: 사용자 활동 관련 정보(예: 클릭 스트림 또는 시작/중단/완료된 트랜잭션)를 생성하도록 애플리케이션 코드를 설계합니다. 이 정보를 사용하면 애플리케이션 사용 방법과 사용 패턴을 파악하고 응답이 필요한 경우를 확인할 수 있습니다.

종속성 원격 측정 구현: 종속된 리소스의 상태에 대한 정보(도달 가능성 또는 응답 시간)를 생성하도록 워크로드를 설계 및 구성합니다. 외부 종속성의 예로는 외부 데이터베이스, DNS, 네트워크 연결 등이 있습니다. 이 정보를 사용하여 대응이 필요한지 여부를 판단합니다.

트랜잭션 추적 기능 구현: 워크로드 전반에 걸친 트랜잭션 흐름 관련 정보를 생성하도록 애플리케이션 코드를 구현하고 워크로드 구성 요소를 구성합니다. 이 정보를 사용하면 응답이 필요한 경우를 확인하고 문제의 원인을 파악할 수 있습니다.

AWS 에서 AWS X-Ray 와 같은 분산 추적 서비스를 사용하여 워크로드에서 트랜잭션이 이동할 때 추적을 수집 및 기록하고, 워크로드 및 서비스에서 트랜잭션의 흐름을 알 수 있는 맵을 생성하며, 구성 요소 간 관계에 대한 인사이트를 얻고, 실시간으로 문제를 식별하고 분석할 수 있습니다.

워크로드가 발전함에 따라 원격 측정을 개발 및 반복하며 워크로드 상태에 대한 인사이트를 얻는 데 필요한 정보를 지속적으로 수신할 수 있도록 합니다.

리소스

운영을 고려한 설계와 관련한 AWS 모범 사례에 대해 자세히 알아보려면 다음 리소스를 참조하십시오.

동영상

- AWS re:Invent 2016: Infrastructure Continuous Delivery Using AWS CloudFormation(DEV313)
- AWS re:Invent 2016: DevOps on AWS: Accelerating Software Delivery with AWS Developer Tools(DEV201)
- AWS CodeStar: The Central Experience to Quickly Start Developing Applications on AWS

문서

- AWS Lambda 에 대한 Amazon CloudWatch Logs 액세스
- Amazon CloudWatch Logs 를 사용하여 CloudTrail 로그 파일 모니터링
- CloudWatch Logs 에 플로우 로그 게시

설명서

- Enhancing workload observability using Amazon CloudWatch Embedded Metric Format
- Amazon CloudWatch 시작하기
- Store and Monitor OS & Application Log Files with Amazon CloudWatch
- High-Resolution Custom Metrics and Alarms for Amazon CloudWatch
- Monitoring AWS Health Events with Amazon CloudWatch Events
- AWS CloudFormation 설명서
- AWS 개발자 도구
- AWS 에서 CI/CD 파이프라인 설정
- AWS X-Ray
- AWS 태그 지정 전략
- Enhancing workload observability using Amazon CloudWatch Embedded Metric Format

운영을 고려한 설계

프로덕션 환경으로 변경 사항을 전달하는 흐름을 개선할 수 있는 방식을 도입합니다. 이 방식은 리팩터링, 품질과 관련된 빠른 피드백 및 버그 수정을 지원해야 합니다. 이러한 방식을 도입하면 유용한 변경 사항을 프로덕션 환경으로 빠르게 전달할 수 있고, 배포 이슈 가능성을 제한할 수 있으며, 배포 활동을 통해 발생하는 문제를 빠르게 파악하고 해결할 수 있습니다.

AWS 에서는 전체 워크로드(애플리케이션, 인프라, 정책, 거버넌스, 운영)를 코드로 확인할 수 있습니다. 즉, 코드를 사용하여 모든 워크로드를 정의하고 업데이트할 수 있습니다. 그러면 애플리케이션 코드에 사용하는 것과 같은 엔지니어링 원칙을 스택의 모든 요소에 적용할 수 있습니다.

버전 관리 사용: 버전 관리를 사용하면 변경 사항과 릴리스를 추적할 수 있습니다.

많은 AWS 서비스가 버전 관리 기능을 제공합니다. AWS CodeCommit 와 같은 수정본 또는 소스 제어 시스템을 사용하여 코드 및 버전을 관리하는 인프라의 AWS CloudFormation 템플릿과 같은 기타 아티팩트를 관리합니다.

변경 사항 테스트 및 확인: 변경 사항을 테스트하고 확인하면 오류를 제한하고 감지할 수 있습니다. 그리고 테스트를 자동화하면 수동 프로세스에서 발생하는 오류와 테스트해야 하는 작업량을 줄일 수 있습니다.

AWS 에서는 실험과 테스트의 위험, 작업량 및 비용을 줄일 수 있는 임시 병렬 환경을 생성할 수 있습니다. AWS CloudFormation 을 사용하여 이러한 환경의 배포를 자동화하면 임시 환경을 일관되게 구현할 수 있습니다.

구성 관리 시스템 사용: 구성 관리 시스템을 사용하면 구성을 변경하고 변경 사항을 추적할 수 있습니다. 이러한 시스템에서는 수동 프로세스에서 발생하는 오류와 변경 사항 배포를 위한 작업량을 줄일 수 있습니다.

빌드 및 배포 관리 시스템 사용: 빌드 및 배포 관리 시스템을 사용합니다. 이러한 시스템에서는 수동 프로세스에서 발생하는 오류와 변경 사항 배포를 위한 작업량을 줄일 수 있습니다.

AWS 에서 AWS CodeCommit, AWS CodeBuild, AWS CodePipeline, AWS CodeDeploy 및 AWS CodeStar 등의 AWS 개발자 도구를 사용하여 CI/CD(지속적 통합/지속적 전달) 파이프라인을 구축할 수 있습니다.

패치 관리 수행: 패치 관리를 수행하면 기능을 확인하고, 문제를 해결하며, 거버넌스 규정 준수 상태를 유지할 수 있습니다. 그리고 패치 관리를 자동화하면 수동 프로세스에서 발생하는 오류와 패치를 위한 작업량을 줄일 수 있습니다.

패치 및 취약성 관리는 이점 관리 및 위험 관리 활동의 일부입니다. 변경이 불가능한 인프라를 보유하고 검증된 정상 상태의 워크로드를 배포하는 것이 좋습니다. 이 방식을 실현할 수 없으면 남은 방법은 패치를 적용하는 것입니다.

취약성을 없애기 위해 머신 이미지, 컨테이너 이미지 또는 Lambda 사용자 지정 런타임 및 추가 라이브러리를 업데이트하는 방법도 패치 관리에 속합니다. EC2 Image Builder 를 사용하여 Linux 또는 Windows Server 용 Amazon Machine Image(AMI)에 대한 업데이트를 관리해야 합니다. 기존 파이프라인과 함께 Amazon Elastic Container Registry 를 사용하여 Amazon ECS 이미지를 관리하고 Amazon EKS 이미지를 관리할 수 있습니다. AWS Lambda 에는 버전 관리 기능이 있습니다.

패치는 먼저 안전한 환경에서 테스트를 거치지 않고는 프로덕션 환경에서 수행해서는 안 됩니다. 패치는 운영 또는 비즈니스 성과를 지원하는 경우에만 적용해야 합니다. AWS 에서는 AWS Systems Manager 패치 관리자를 사용하여 관리형 시스템에 패치를 적용하는 프로세스를 자동화하고 AWS Systems Manager Maintenance Windows 를 사용하여 이 활동을 예약할 수 있습니다.

설계 표준 공유: 여러 팀이 모범 사례를 공유하면 표준에 대한 인지도를 높이고 개발 작업의 이점을 극대화할 수 있습니다.

AWS 에서는 코드 방법론을 사용해 애플리케이션, 컴퓨팅, 인프라 및 운영을 정의하고 관리할 수 있습니다. 따라서 기능을 쉽게 릴리스, 공유 및 도입할 수 있습니다.

많은 AWS 서비스와 리소스는 여러 계정 간에 공유가 가능하도록 설계되어 있으므로, 만들어진 자산과 파악한 정보를 여러 팀이 공유할 수 있습니다. 예를 들어, CodeCommit 리포지토리, Lambda 함수, Amazon S3 버킷 및 AMI 를 특정 계정과 공유할 수 있습니다.

새 리소스 또는 업데이트를 게시할 때는 Amazon SNS 를 사용하여 교차 계정 알림을 제공합니다. 구독자는 Lambda 를 사용하여 새 버전을 얻을 수 있습니다.

조직에 공유 표준이 적용되면 팀 활동을 지원하기 위해 표준에 대한 추가, 변경 및 예외 처리를 요청하는 메커니즘을 갖추어야 합니다. 이 옵션이 없으면 표준이 혁신의 제약 요인이 됩니다.

코드 품질을 개선하는 사례 구현: 코드 품질을 개선하고 결함을 최소화하는 사례를 구현합니다. 테스트 중심 개발, 코드 검토, 표준 도입 등을 예로 들 수 있습니다.

여러 환경 사용: 여러 환경을 사용하여 워크로드를 실험, 개발 및 테스트합니다. 프로덕션 환경에 다가감에 따라 제어 수준을 계속 높이면 배포 시 워크로드가 의도한 대로 운영될지 신뢰할 수 있습니다.

되돌릴 수 있는 작은 단위의 변경 내용 자주 적용: 되돌릴 수 있는 작은 단위의 변경 내용을 자주 적용하면 변경의 범위와 그로 인한 영향을 줄일 수 있습니다. 이렇게 하면 문제를 더 빠르고 쉽게 해결할 수 있으며 변경 사항 롤백 옵션을 사용할 수 있습니다.

통합 및 배포 완전 자동화: 워크로드 빌드, 배포 및 테스트를 자동화합니다. 이렇게 하면 수동 프로세스에서 발생하는 오류와 변경 사항 배포를 위한 작업을 줄일 수 있습니다.

리소스 태그 및 AWS 리소스 그룹을 사용하여 메타데이터를 적용하고 일관된 태그 지정 전략을 시행하면 리소스를 식별할 수 있습니다. 리소스에 조직, 회계 비용, 접근 제어에 대한 태그를 지정하여 자동화된 운영 활동을 실행할 대상을 설정합니다.

리소스

운영을 고려한 설계와 관련한 AWS 모범 사례에 대해 자세히 알아보려면 다음 리소스를 참조하십시오.

동영상

- AWS re:Invent 2016: Infrastructure Continuous Delivery Using AWS CloudFormation(DEV313)
- AWS re:Invent 2016: DevOps on AWS: Accelerating Software Delivery with AWS Developer Tools(DEV201)
- AWS CodeStar: The Central Experience to Quickly Start Developing Applications on AWS

설명서

- AWS 리소스 그룹이란 무엇입니까?
- Amazon CloudWatch 시작하기
- Store and Monitor OS & Application Log Files with Amazon CloudWatch
- High-Resolution Custom Metrics and Alarms for Amazon CloudWatch
- Amazon CloudWatch Events 로 AWS 상태 이벤트 모니터링
- AWS CloudFormation 설명서
- AWS 개발자 도구
- AWS 에서 CI/CD 파이프라인 설정
- AWS X-Ray
- AWS 태그 지정 전략

배포 위험 완화

품질과 관련한 피드백을 빠르게 제공하며, 적절한 성과를 달성하는 데 도움이 되지 않는 변경을 수행한 경우 신속하게 복구할 수 있는 방식을 도입합니다. 이러한 사례를 사용하면 변경 사항 배포로 인해 발생하는 문제의 영향을 완화할 수 있습니다.

워크로드 설계에는 워크로드를 배포, 업데이트 및 동작할 방법을 포함해야 합니다. 결함을 줄이고 신속하고 안전하게 결함을 수정하는 엔지니어링 방식을 구현해야 합니다.

실패한 변경에 대한 계획: 변경을 수행했는데 적절한 성과를 달성하는 데 도움이 되지 않는다면 잘 동작했던 정상 상태로 되돌릴 수 있는 계획을 세우거나 프로덕션 환경에서 관련 문제를 해결합니다. 이와 같이 준비를 하면 문제에 더욱 신속하게 응답함으로써 복구 시간을 단축할 수 있습니다.

변경 테스트 및 검증: 수명 주기의 모든 단계에서 변경 사항을 테스트하고 결과를 검증하여 새 기능을 확인하고 배포 실패의 영향과 위험을 최소화합니다.

AWS 에서는 실험과 테스트의 위험, 작업량 및 비용을 줄일 수 있는 임시 병렬 환경을 생성할 수 있습니다. AWS CloudFormation 을 사용하여 이러한 환경의 배포를 자동화하면 임시 환경을 일관되게 구현할 수 있습니다.

배포 관리 시스템 사용: 배포 관리 시스템을 사용하여 변경을 추적하고 구현합니다. 이렇게 하면 수동 프로세스에서 발생하는 오류와 변경 사항 배포를 위한 작업을 줄일 수 있습니다.

AWS 에서 AWS CodeCommit, AWS CodeBuild, AWS CodePipeline, AWS CodeDeploy 및 AWS CodeStar 등의 AWS 개발자 도구를 사용하여 CI/CD(지속적 통합/지속적 전달) 파이프라인을 구축할 수 있습니다.

변경 구현에 영향을 줄 수 있는 중요한 비즈니스나 운영 활동 또는 이벤트가 계획된 경우 변경 일정을 정하고 이를 추적합니다. 활동을 조정하여 이러한 계획에 관한 위험을 관리합니다. AWS Systems Manager Change Calendar 에서는 변경의 개시 또는 종료에 해당하는 시간 블록과 변경 이유를 문서화하고, 다른 AWS 계정과 이 정보를 공유하는 메커니즘을 제공합니다. 그리고 변경 일정 상태를 준수하도록 AWS Systems Manager Automation 스크립트를 구성할 수 있습니다.

AWS Systems Manager Maintenance Windows 를 사용하면 AWS SSM Run Command 나 자동화 스크립트, AWS Lambda 호출 또는 AWS Step Function 활동을 지정된 시간에 수행하도록 예약할 수 있습니다. 이러한 활동이 평가에 포함될 수 있도록 변경 일정에서 활동을 표시합니다.

제한된 배포를 사용한 테스트: 전체 배포를 진행하기 전에 기존 시스템과 함께 제한된 배포를 사용해 테스트를 진행하여 원하는 결과를 달성할 수 있는지를 확인합니다. 예를 들어, Canary 배포 테스트나 원박스 배포를 사용합니다.

병렬 환경을 사용한 배포: 병렬 환경에 변경 사항을 구현한 다음 새 환경으로 이전합니다. 배포가 정상 완료되었음이 확인될 때까지는 이전 환경을 유지합니다. 그러면 만일의 경우 이전 환경으로 롤백할 수 있으므로 복구 시간을 최소화할 수 있습니다.

되돌릴 수 있는 작은 단위의 변경 내용 자주 배포: 되돌릴 수 있는 작은 단위의 변경 내용을 자주 사용하면 변경의 범위를 줄일 수 있습니다. 그러면 문제를 더 쉽게 해결할 수 있으며 변경 사항 롤백 옵션을 사용해 문제 해결 시간을 단축할 수 있습니다.

통합 및 배포 완전 자동화: 워크로드 빌드, 배포 및 테스트를 자동화합니다. 이렇게 하면 수동 프로세스에서 발생하는 오류와 변경 사항 배포를 위한 작업을 줄일 수 있습니다.

테스트 및 롤백 자동화: 배포된 환경에서 테스트를 자동화하여 원하는 성과를 달성할 수 있는지를 확인합니다. 원하는 결과를 달성할 수 없는 경우 잘 동작됐던 정상 상태로 롤백하는 과정을 자동화하면 수동 프로세스에서 발생하는 오류를 줄이고 복구 시간을 최소화할 수 있습니다.

리소스

운영을 고려한 설계와 관련한 AWS 모범 사례에 대해 자세히 알아보려면 다음 리소스를 참조하십시오.

동영상

- AWS re:Invent 2016: Infrastructure Continuous Delivery Using AWS CloudFormation(DEV313)
- AWS re:Invent 2016: DevOps on AWS: Accelerating Software Delivery with AWS Developer Tools(DEV201)
- AWS CodeStar: The Central Experience to Quickly Start Developing Applications on AWS

설명서

- Amazon CloudWatch 시작하기
- Store and Monitor OS & Application Log Files with Amazon CloudWatch
- High-Resolution Custom Metrics and Alarms for Amazon CloudWatch
- Amazon CloudWatch Events 로 AWS 상태 이벤트 모니터링
- AWS CloudFormation 설명서
- AWS 개발자 도구
- AWS 에서 CI/CD 파이프라인 설정
- AWS X-Ray
- AWS 태그 지정 전략

운영 준비

워크로드, 프로세스, 절차 및 직원의 운영 준비 상태를 평가하여 워크로드와 관련된 운영 위험을 파악합니다.

수동 또는 자동화된 체크리스트를 비롯한 일관된 프로세스를 사용해 워크로드 또는 변경에 응답하는 준비 여부를 확인해야 합니다. 이렇게 하면 문제 해결 계획을 세워야 하는 영역도 파악할 수 있습니다. 일상 활동을 문서화한 런북과 문제 해결 프로세스를 안내하는 플레이북이 제공됩니다.

직원의 역량 확보: 운영상의 요구 사항을 지원하기 위한 적절한 수의 숙련된 인력이 있는지 확인하는 메커니즘을 확보합니다. 효과적인 지원을 유지하기 위해 필요한 경우 직원을 교육하고 인원을 조정합니다.

업무 대기 중인 인력을 포함하여 모든 활동을 다룰 수 있는 팀원을 충분히 보유해야 합니다. 팀이 워크로드, 운영 도구 및 AWS 에 대한 교육을 성공적으로 이수하는 데 필요한 기술을 갖추었는지 확인합니다.

AWS 에서는 AWS 시작하기 리소스 센터, AWS 블로그, AWS Online Tech Talks, AWS 이벤트 및 웹 세미나, AWS Well-Architected Labs 등의 다양한 리소스를 제공합니다. 이러한 리소스에서는 팀 대상으로 교육을 진행하는 데 활용할 수 있는 지침, 예제 및 자세한 연습 과정을 제공합니다. 또한 AWS Training and Certification 에서는 AWS 기초에 관한 자습형 디지털 교육 과정을 통해 몇 가지 무료 교육을 제공합니다. 팀이 AWS 기술을 발전시켜 나갈 수 있도록 강의식 교육에 등록할 수도 있습니다.

일관된 방식으로 운영 준비 검토: 워크로드를 운영할 준비가 되었는지를 일관된 방식으로 검토합니다. 검토에서는 최소한 팀 및 워크로드의 운영 준비 상태와 보안 요구 사항을 파악해야 합니다. 코드에서 검토 활동을 구현하고 해당하는 경우 이벤트 응답 과정에서 자동화된 검토를 트리거하면 일관성을 유지하고, 실행 속도를 높이고, 수동 프로세스에서 발생하는 오류를 줄일 수 있습니다.

또한 AWS Config 를 사용하여 기준을 설정하고 AWS Config 규칙을 사용해 구성을 확인하여 워크로드 구성 테스트를 자동화해야 합니다. AWS Security Hub 의 서비스 및 기능을 사용하여

보안 요구 사항과 규정 준수를 평가할 수 있습니다. 이러한 서비스를 활용하면 워크로드가 모범 사례와 표준을 따르는지를 쉽게 확인할 수 있습니다.

런북을 사용하여 절차 수행: 런북은 특정 결과를 달성하기 위한 문서화된 절차입니다. 절차를 런북으로 문서화하면 적절하게 파악한 이벤트에 일관된 방식으로 신속하게 대응할 수 있습니다. 런북을 코드로 구현하고, 해당하는 경우 이벤트 응답 과정에서 런북 실행을 트리거하여 일관성을 유지하고, 응답 속도를 높이고, 수동 프로세스에서 발생하는 오류를 줄일 수 있습니다.

플레이북을 사용하여 문제 식별: 플레이북은 문제 조사를 위한 문서화된 프로세스입니다. 플레이북에 조사 프로세스를 문서화하면 장애 발생 시나리오에 일관적이고 빠르게 대응할 수 있습니다. 플레이북을 코드로 구현하고, 해당하는 경우 이벤트 응답 과정에서 플레이북 실행을 트리거하여 일관성을 유지하고, 응답 속도를 높이고, 수동 프로세스에서 발생하는 오류를 줄일 수 있습니다.

AWS 를 사용하면 운영 작업을 코드로 처리할 수 있습니다. 즉, 런북과 플레이북 활동을 스크립트로 작성해 인적 오류 위험을 줄일 수 있습니다. 리소스 태그 또는 리소스 그룹과 스크립트를 함께 사용하면 환경, 책임자, 역할, 버전 등의 직접 정의한 기준에 따라 코드를 선택적으로 실행할 수 있습니다.

스크립팅된 절차를 사용하면 이벤트에 대한 응답으로 스크립트를 트리거하여 활동을 자동화할 수 있습니다. 운영 작업과 워크로드를 모두 코드로 처리하면 환경 평가 작업도 스크립트로 작성하여 자동화할 수 있습니다.

AWS Systems Manager(SSM) Run Command 를 사용하여 인스턴스에서 수행할 절차를 스크립트로 작성하거나 AWS Systems Manager Automation 을 사용하여 인스턴스 및 기타 리소스에서 조치를 스크립트로 작성하고 워크플로를 생성하거나 AWS Lambda 서버리스 컴퓨팅 기능을 사용하여 AWS 서비스 API 및 사용자 지정 인터페이스에서 이벤트에 대한 응답을 스크립트로 작성해야 합니다. 또한 AWS Step Functions 를 사용하면 스크립트를 통해 여러 AWS 서비스를 서버리스 워크플로로 조정할 수 있습니다. CloudWatch 이벤트를 사용하여 이러한 스크립트를 트리거하여 응답을 자동화하고, Amazon EventBridge 를 사용하여 추가 운영 지원 시스템에 원하는 이벤트를 제공합니다.

게임 데이를 진행하고 실제 가동 전에 테스트를 진행하는 등의 방법을 통해 절차, 장애 시나리오 및 대응 방식의 적절성 여부를 테스트하여 문제 해결 계획을 세워야 하는 영역을 파악해야 합니다.

AWS 에서는 실험과 테스트의 위험, 작업량 및 비용을 줄일 수 있는 임시 병렬 환경을 생성할 수 있습니다. AWS CloudFormation 을 사용하여 이러한 환경의 배포를 자동화하면 임시 환경을 일관되게 구현할 수 있습니다. 고객에 영향을 주지 않거나 허용가능한 환경에서 오류 삽입 테스트를 수행하고, 적절한 응답을 개발하거나 수정합니다.

정보에 입각하여 시스템 및 변경 사항 배포 결정: 워크로드를 지원할 수 있는 팀의 능력과 워크로드의 거버넌스 준수 여부를 평가합니다. 배포의 이점을 기준으로 하여 이러한 평가를 수행해 시스템 또는 변경 사항을 프로덕션 환경으로 전환할지 여부를 결정합니다. 이점과 위험을 파악하면 정보에 입각한 결정을 내릴 수 있습니다.

해당하는 경우에는 "사전 분석(pre-mortem)" 기능을 사용하여 장애를 예측하고 절차를 생성합니다. 워크로드를 평가하는 데 사용하는 체크리스트를 변경할 때는 해당 변경으로 인해 더 이상 규정에 맞지 않는 라이브 시스템에 대해 수행할 작업을 계획합니다.

리소스

운영 준비와 관련한 AWS 모범 사례에 대해 자세히 알아보려면 다음 리소스를 참조하십시오.

설명서

- AWS Lambda
- AWS Systems Manager
- AWS Config Rules – Dynamic Compliance Checking for Cloud Resources
- How to track configuration changes to CloudFormation stacks using AWS Config
- Amazon Inspector 업데이트 블로그 게시물
- AWS 이벤트 및 웹 세미나
- AWS 교육
- Well-Architected 실습
- AWS launches Tag Policies
- Using AWS Systems Manager Change Calendar to prevent changes during critical events

운영

성공은 직접 정의한 지표를 기준으로 측정된 비즈니스 성과를 달성했음을 의미합니다. 워크로드 및 운영 상태를 이해하면 조직 및 비즈니스 성과가 지금 위험한지, 아니면 앞으로 위험해질 것인지를 파악하고 적절히 대응할 수 있습니다.

성공하려면 다음을 수행할 수 있어야 합니다.

- 워크로드 상태 파악
- 운영 상태 파악
- 이벤트에 대응

워크로드 상태 파악

워크로드 지표를 정의, 캡처 및 분석하면 워크로드 이벤트에 대한 가시성을 확보하여 적절한 조치를 취할 수 있습니다.

팀이 워크로드의 상태를 쉽게 파악할 수 있어야 합니다. 그러려면 워크로드 성과를 기준으로 하는 지표를 사용하여 유용한 분석 정보를 습득해야 합니다. 이러한 지표를 사용해 비즈니스 및 기술적 시각이 반영된 대시보드를 구현해야 합니다. 팀원은 이러한 대시보드를 통해 정보를 파악하여 적절한 결정을 내릴 수 있습니다.

AWS 에서는 워크로드 로그를 더욱 쉽게 취합하고 분석하여 지표를 생성하고, 워크로드 상태를 확인하며, 시간이 지남에 따라 운영으로부터 인사이트를 확보할 수 있습니다.

핵심 성과 지표 파악: 원하는 비즈니스 성과(예: 주문율, 고객 유지율, 수익 및 운영 지출 비교)와 고객 성과(예: 고객 만족도)를 기반으로 KPI(핵심 성과 지표)를 파악합니다. 그리고 KPI 를 평가하여 워크로드의 성공 여부를 결정합니다.

워크로드 지표 정의: KPI 달성(예: 포기한 장바구니, 제출된 주문, 비용, 가격 및 할당된 워크로드 지출)을 측정하도록 워크로드 지표를 정의합니다. 워크로드 상태(예: 인터페이스 응답 시간, 오류 발생률, 제출된 요청, 완료된 요청 및 사용률)를 측정하도록 워크로드 지표를 정의합니다. 그런

다음, 해당 지표를 평가해 워크로드에서 적절한 성과를 달성할 수 있는지를 확인하고 워크로드의 상태를 파악합니다.

CloudWatch Logs 와 같은 서비스로 로그 데이터를 전송하고 필요한 로그 콘텐츠를 관찰하여 지표를 생성해야 합니다.

CloudWatch 에는 Amazon CloudWatch Insights for .NET and SQL Server 및 Container Insights 와 같은 특별한 기능이 있습니다. 이 기능을 사용하면 특별히 지원되는 애플리케이션 리소스 및 기술 스택에서 핵심 지표, 로그 및 경보를 파악하고 설정할 수 있습니다.

워크로드 지표 수집 및 분석: 사전 예방 차원에서 지표를 정기적으로 점검하여 트랜드를 확인하고 어느 부분에 적절한 대응이 필요한지를 파악합니다.

애플리케이션, 워크로드 구성 요소, 서비스 및 API 호출의 로그 데이터를 CloudWatch Logs 와 같은 서비스로 집계해야 합니다. 운영 활동의 성과에 대한 인사이트를 얻을 수 있도록 필요한 로그 콘텐츠를 관찰하여 지표를 생성합니다.

AWS 공동 책임 모델에서는 AWS Personal Health Dashboard 를 통해 모니터링 일부를 고객에게 제공합니다. 이 대시보드는 AWS 가 고객에게 영향을 미칠 수 있는 이벤트를 겪고 있을 때 이를 알리고 수정 지침을 제공합니다. Business 및 Enterprise Support 구독 고객은 PHD API 도 사용할 수 있으며 이벤트 관리 시스템에 AWS Health API 를 통합할 수 있습니다.

AWS 에서 Amazon S3 로 로그 데이터를 내보내거나 장기 보관을 위해 Amazon S3 로 로그를 직접 전송할 수 있습니다. AWS Glue 를 사용하면 분석을 위해 Amazon S3 의 로그 데이터를 검색하고 준비하여 AWS Glue Data Catalog 에 관련된 메타데이터를 저장할 수 있습니다. 그리고 Amazon Athena 에서 Glue 와의 기본 통합을 통해 로그 데이터를 분석하고 표준 SQL 을 사용해 쿼리할 수 있습니다. Amazon QuickSight 와 같은 비즈니스 인텔리전스 도구를 사용하면 데이터를 시각화하고 탐색하며 분석할 수 있습니다.

Amazon Elasticsearch Service 및 Kibana 를 사용하여 여러 계정과 AWS 리전에서 AWS 의 로그를 수집, 분석 및 표시하는 것도 대안이 될 수 있습니다.

워크로드 지표 기준선 설정: 지표의 기준을 설정해 성능이 기준보다 높은/낮은 구성 요소를 확인하고 각 구성 요소의 성능을 비교할 수 있는 기준으로 필요한 값을 제공합니다. 개선, 조사 및 개입을 위한 임계값을 파악합니다.

워크로드에 대한 예상 활동 패턴 파악: 필요한 경우 적절하게 대응할 수 있도록 비정상적인 동작을 식별할 워크로드 활동 패턴을 설정합니다.

CloudWatch 에서는 CloudWatch 이상 탐지 기능을 통해 통계 및 기계 학습 알고리즘을 적용해 정상 지표 동작을 나타내는 예상되는 값의 범위를 생성합니다.

워크로드 성과가 위험한 상태이면 알림 생성: 워크로드 성과가 위험한 상태이면 알림을 생성합니다. 그러면 필요할 때 적절하게 대응할 수 있습니다.

이전에는 자동화된 응답을 트리거하는 데 사용할 수 있는 이벤트 또는 경보를 알릴 수 있는 지표 임계값을 식별했습니다.

또한 CloudWatch Logs Insights 에서 특별히 구축된 쿼리 언어를 사용해 로그 데이터를 대화식으로 검색하고 분석할 수 있습니다. CloudWatch Logs Insights 는 AWS 서비스에서 로그의 필드와 사용자 지정 로그 이벤트를 JSON 형식으로 자동 검색합니다. 그러면 로그 볼륨 및 쿼리 복잡성에 대한 지원을 확장하고 몇 초 안에 답변을 제공하므로 인시던트의 원인을 파악하는 데 도움이 됩니다.

워크로드에 이상이 감지되면 알림 생성: 워크로드에서 이상 상태가 감지되면 알림을 생성합니다. 그러면 필요할 때 적절하게 대응할 수 있습니다.

시간에 따른 워크로드 지표를 분석하면 이벤트를 정의하거나 이벤트 응답으로 경보를 울리기 위해 정량화할 수 있는 동작의 패턴을 설정할 수 있습니다.

학습된 후에는 CloudWatch Anomaly Detection 기능을 사용하여 탐지된 이상현상에 대한 경보를 생성하거나 비교를 위해 지표 데이터의 그래프에서 중첩된 예상되는 값을 제공할 수 있습니다.

성과 달성 여부와 KPI 및 지표의 효율성 확인: 워크로드 운영을 실무 수준에서 확인할 수 있는 보기를 생성합니다. 그러면 요구를 충족하고 있는지 확인할 수 있으며 업무 목표 달성을 위해 개선해야 하는 영역을 파악할 수 있습니다. 또한 KPI 와 지표의 효율성을 확인하고, 필요한 경우 수정합니다.

AWS 는 AWS 서비스 API 및 SDK 를 통해 서드파티 로그 분석 시스템 및 비즈니스 인텔리전스 도구(예: Grafana, Kibana, Logstash)도 지원합니다.

리소스

워크로드 상태 파악과 관련한 AWS 모범 사례에 대해 자세히 알아보려면 다음 리소스를 참조하십시오.

동영상

- AWS re:Invent 2015: Log, Monitor, and Analyze your IT with Amazon CloudWatch(DVO315)
- AWS re:Invent 2016: Amazon CloudWatch Logs and AWS Lambda: A Match Made in Heaven(DEV301)

설명서

- Amazon CloudWatch Application Insights for .NET and SQL Server 란 무엇입니까?
- Store and Monitor OS & Application Log Files with Amazon CloudWatch
- API & CloudFormation Support for Amazon CloudWatch Dashboards
- AWS Answers: Centralized Logging

운영 상태 파악

운영 지표를 정의, 캡처 및 분석하면 워크로드 이벤트에 대한 가시성을 확보하여 적절한 조치를 취할 수 있습니다.

팀이 운영 상태를 쉽게 파악할 수 있어야 합니다. 그러려면 운영 성과를 기준으로 하는 지표를 사용하여 유용한 분석 정보를 습득해야 합니다. 이러한 지표를 사용해 비즈니스 및 기술적 시각이 반영된 대시보드를 구현해야 합니다. 팀원은 이러한 대시보드를 통해 정보를 파악하여 적절한 결정을 내릴 수 있습니다.

AWS 에서는 운영 로그를 더욱 쉽게 취합하여 지표를 생성하고, 운영 상태를 확인하며, 시간의 흐름에 따라 운영으로부터 인사이트를 확보할 수 있습니다.

핵심 성과 지표 파악: 원하는 비즈니스 성과(예: 새로운 기능 제공) 및 고객 성과(예: 고객 지원 사례)를 기반으로 KPI(핵심 성과 지표)를 파악합니다. 그리고 KPI 를 평가하여 운영의 성공 여부를 결정합니다.

운영 지표 정의: KPI 성과(예: 성공한 배포와 실패한 배포)를 측정하는 데 사용할 운영 지표를 정의합니다. 운영 활동 상태(예: 인시던트의 MTTD(평균 탐지 시간) 및 인시던트의 MTTR(평균 복구 시간))를 측정하는 데 사용할 운영 지표를 정의합니다. 그런 다음, 해당 지표를 평가해 운영 과정에서 적절한 성과를 달성할 수 있는지를 확인하고 운영 활동 상태를 파악합니다.

운영 지표 수집 및 분석: 사전 예방 차원에서 지표를 정기적으로 점검하여 트랜드를 확인하고 어느 부분에 적절한 대응이 필요한지를 파악합니다.

운영 활동 및 운영 API 호출의 실행에서 CloudWatch Logs 와 같은 서비스로 로그 데이터를 집계해야 합니다. 운영 활동의 성과에 대한 인사이트를 얻을 수 있도록 필요한 로그 콘텐츠를 관찰하여 지표를 생성합니다.

AWS 에서 Amazon S3 로 로그 데이터를 내보내거나 장기 보관을 위해 Amazon S3 로 로그를 직접 전송할 수 있습니다. AWS Glue 를 사용하면 분석을 위해 Amazon S3 의 로그 데이터를 검색하고 준비하여 AWS Glue Data Catalog 에 관련된 메타데이터를 저장할 수 있습니다. 그리고 Amazon Athena 에서 Glue 와의 기본 통합을 통해 로그 데이터를 분석하고 표준 SQL 을 사용해 쿼리할 수 있습니다. Amazon QuickSight 와 같은 비즈니스 인텔리전스 도구를 사용하면 데이터를 시각화하고 탐색하며 분석할 수 있습니다.

운영 지표 기준 설정: 지표의 기준을 설정해 성능이 기준보다 높은/낮은 프로세스를 확인하고 각 운영 활동 성과(초과 또는 미달)를 비교할 수 있는 기준으로 필요한 값을 제공합니다.

워크로드에 대한 예상 활동 패턴 파악: 필요한 경우 적절하게 대응할 수 있도록 비정상적인 활동을 식별할 운영 활동 패턴을 설정합니다.

워크로드 성과가 위험한 상태이면 알림 생성: 운영 성과가 위험한 상태이면 알림을 생성합니다. 그러면 필요할 때 적절하게 대응할 수 있습니다.

이전에는 자동화된 응답을 트리거하는 데 사용할 수 있는 이벤트 또는 경보를 알릴 수 있는 지표를 식별했습니다.

또한 CloudWatch Logs Insights 에서 특별히 구축된 쿼리 언어를 사용해 로그 데이터를 대화식으로 검색하고 분석할 수 있습니다. CloudWatch Logs Insights 는 AWS 서비스에서 로그의 필드와 사용자 지정 로그 이벤트를 JSON 형식으로 자동 검색합니다. 그러면 로그 볼륨 및

쿼리 복잡성에 대한 지원을 확장하고 몇 초 안에 답변을 제공하므로 인시던트의 원인을 파악하는 데 도움이 됩니다.

운영에 이상이 감지되면 알림 생성: 운영에서 이상 상태가 감지되면 알림을 생성합니다. 그러면 필요할 때 적절하게 대응할 수 있습니다.

시간에 따른 운영 지표를 분석하면 이벤트를 정의하거나 이벤트 응답으로 경보를 울리기 위해 정량화할 수 있는 동작의 패턴을 설정할 수 있습니다.

학습된 후에는 CloudWatch Anomaly Detection 기능을 사용하여 탐지된 이상 현상에 대한 경보를 생성하거나 비교를 위해 지표 데이터의 그래프에서 중첩된 예상되는 값을 제공할 수 있습니다.

성과 달성 여부와 KPI 및 지표의 효율성 확인: 운영 활동을 실무 수준에서 확인할 수 있는 뷰(view)를 생성합니다. 그러면 요구를 충족하고 있는지 확인할 수 있으며 업무 목표 달성을 위해 개선해야 하는 영역을 파악할 수 있습니다. 또한 KPI 와 지표의 효율성을 확인하고, 필요한 경우 수정합니다.

AWS 는 AWS 서비스 API 및 SDK 를 통해 서드파티 로그 분석 시스템 및 비즈니스 인텔리전스 도구(예: Grafana, Kibana, Logstash)도 지원합니다.

리소스

운영 상태 파악과 관련한 AWS 모범 사례에 대해 자세히 알아보려면 다음 리소스를 참조하십시오.

동영상

- AWS re:Invent 2015: Log, Monitor, and Analyze your IT with Amazon CloudWatch(DVO315)
- AWS re:Invent 2016: Amazon CloudWatch Logs and AWS Lambda: A Match Made in Heaven(DEV301)

설명서

- Store and Monitor OS & Application Log Files with Amazon CloudWatch
- API & CloudFormation Support for Amazon CloudWatch Dashboards
- AWS Answers: Centralized Logging

이벤트에 응답

계획된 운영 이벤트(예: 판매 프로모션, 배포 및 장애 테스트)와 계획되지 않은 운영 이벤트(예: 사용량 및 구성 요소 장애의 급증)를 모두 예상해야 합니다. 알림에 응답할 때는 일관된 결과가 제공되도록 기존 런북과 플레이북을 사용해야 합니다. 대응과 에스컬레이션을 담당하는 역할 또는 팀이 정의된 알림을 보유해야 합니다. 또한 시스템 구성 요소가 업무에 주는 영향을 확인하고, 해당 정보를 활용해 필요시의 작업 대상을 지정할 수 있습니다. 이벤트 후에는 RCA(근본 원인 분석)를 수행해야 하며 장애 재발을 방지하거나 해결 방법을 문서로 작성해야 합니다.
AWS 에서는 워크로드와 운영의 모든 측면을 코드로 지원하는 도구가 제공되므로 이벤트에 손쉽게 대응할 수 있습니다. 이러한 도구를 사용하면 운영 이벤트 관련 응답을 스크립트로 작성할 수 있으며, 모니터링 데이터에 대한 응답에서 스크립트 실행을 트리거할 수 있습니다.

AWS 에서는 장애가 발생한 구성 요소의 복구를 시도하는 대신 잘 동작하는 정상 버전으로 교체함으로써 복구 시간을 줄일 수 있습니다. 그런 후에는 외부 환경에서 장애 발생 리소스 분석을 수행할 수 있습니다.

이벤트, 인시던트 및 문제 관리에 대한 프로세스 사용: 관찰되는 이벤트, 개입이 필요한 이벤트(인시던트), 개입이 필요하며 반복되거나 현재 해결할 수 없는 이벤트(문제)를 처리하기 위한 프로세스를 마련합니다. 이러한 프로세스를 사용하면 적시에 적절한 응답을 보장하여 비즈니스와 고객에 대한 해당 이벤트의 영향을 완화할 수 있습니다.

AWS 에서 AWS Systems Manager OpsCenter 를 중앙에 위치시켜 사용하면 AWS 리소스와 관련된 운영 문제를 보고, 조사하며, 해결할 수 있습니다. 운영 문제를 집계하고 상황에 맞는 관련 데이터를 제공하여 인시던트 응답을 지원합니다.

알림당 프로세스 지정: 경계심을 가져야 하는 이벤트가 있는 경우, 특정하게 식별된 책임자를 지정함과 동시에 명확하게 정의된 대응 방법(런북 또는 플레이북)을 마련합니다. 이렇게 하면 운영 이벤트에 빠르고 효과적으로 대응할 수 있으며 중요하지 않은 알림 때문에 실행 가능한 이벤트를 제대로 확인하지 못하는 상황을 방지할 수 있습니다.

비즈니스 영향에 기반하여 운영 이벤트 우선순위 지정: 여러 이벤트에 대해 조치를 취해야 할 때는 실무에 가장 큰 영향을 주는 이벤트를 먼저 해결해야 합니다. 예를 들어, 이러한 영향에는 생명 또는 부상, 재정적 손실 또는 평판이나 신뢰의 손상이 포함될 수 있습니다.

에스컬레이션 경로 정의: 에스컬레이션을 트리거하는 요소와 에스컬레이션 절차를 포함한 에스컬레이션 경로를 런북과 플레이북에 정의합니다. 운영 이벤트에 즉시 효율적으로 응답할 수 있도록 각 작업의 책임자를 구체적으로 명시합니다.

작업을 수행하기 전에 사람의 결정이 필요한 경우를 확인합니다. 의사 결정권자와 협력하여 미리 의사 결정을 내리고 작업을 사전에 승인합니다. 그러면 답변을 기다리기 위한 MTTR 이 연장되지 않습니다.

푸시 알림 활성화: 사용자가 사용 중인 서비스가 이벤트의 영향을 받을 때 사용자에게 이메일이나 SMS 등을 통해 직접 알리고, 정상 작동 상태로 되돌아갈 때 다시 알립니다. 그러면 사용자가 적절한 조치를 취할 수 있습니다.

대시보드를 통해 상태 전달: 목표 대상(예: 내부 기술 팀, 리더십 및 고객)에게 맞춤화된 대시보드를 제공하여 비즈니스의 현재 운영 상태를 전달하고 관심 있는 지표를 제공합니다.

CloudWatch 콘솔을 통해 사용자 지정 가능한 홈 페이지에서 Amazon CloudWatch 대시보드를 사용하여 대시보드를 생성할 수 있습니다. Amazon QuickSight 와 같은 비즈니스 인텔리전스 서비스를 사용하면 워크로드 및 운영 상태(예: 주문율, 연결된 사용자 수 및 트랜잭션 시간)에 대한 대화형 대시보드를 생성하고 게시할 수 있습니다. 그리고 이러한 지표의 시스템 및 비즈니스 수준의 보기를 제공하는 대시보드를 생성합니다.

이벤트에 대한 응답 자동화: 이벤트 응답을 자동화하면 수동 프로세스에서 발생하는 오류를 줄일 수 있으며 일관된 방식으로 즉시 대응할 수 있습니다.

다양한 방식으로 AWS 에서 런북 및 플레이북 작업 실행을 자동화할 수 있습니다. AWS 리소스의 상태 변경으로 인해 발생하는 이벤트나 사용자 지정 이벤트에 응답하려는 경우 CloudWatch 대상(예: Lambda 함수, Amazon Simple Notification Service(Amazon SNS) 주제, Amazon EC2 태스크, AWS Systems Manager Automation)을 통해 응답을 트리거하는 CloudWatch 이벤트 규칙을 생성해야 합니다.

리소스의 임계값(예: 대기 시간)을 초과하는 지표에 응답하려는 경우에는 Amazon EC2 작업이나 Auto Scaling 작업을 사용해 작업을 하나 이상 수행하거나 Amazon SNS 주제로 알림을 전송하는 CloudWatch 경보를 생성해야 합니다. 경보에 대한 응답으로 사용자 지정 작업을 수행해야 하는

경우에는 Amazon SNS 알림을 통해 Lambda 를 호출합니다. 직원이 정보를 계속 확인할 수 있도록 Amazon SNS 를 사용하여 이벤트 알림 및 에스컬레이션 메시지를 게시합니다.

AWS 는 AWS 서비스 API 및 SDK 를 통해 서드파티 시스템도 지원합니다. APN 파트너와 서드파티에서 제공하는 다양한 모니터링 도구를 모니터링, 알림 및 응답에 사용할 수 있습니다. 이러한 도구의 예로는 New Relic, Splunk, Loggly, SumoLogic, Datadog 등이 있습니다.

자동 절차에서 오류가 발생하는 경우를 대비해 중요 수동 절차를 사용 가능한 상태로 유지해야 합니다.

리소스

이벤트 대응과 관련한 AWS 모범 사례에 대해 자세히 알아보려면 다음 리소스를 참조하십시오.

동영상

- AWS re:Invent 2016: Automating Security Event Response, from Idea to Code to Execution(SEC313)

설명서

- What is Amazon CloudWatch Events?
- How to Automatically Tag Amazon EC2 Resources in Response to API Events
- Amazon EC2 Systems Manager Automation is now an Amazon CloudWatch Events Target
- EC2 Run Command is Now a CloudWatch Events Target
- Automate remediation actions for Amazon EC2 notifications and beyond using EC2 Systems Manager Automation and AWS Health
- High-Resolution Custom Metrics and Alarms for Amazon CloudWatch

개선

개선은 장기적으로 진행되는 지속적인 향상 주기입니다. 운영 활동에서 파악한 내용을 토대로 하여 소규모 증분 방식 변경을 자주 구현하고, 성공적으로 개선했는지를 평가해야 합니다.

시간에 따라 운영을 개선하려면 다음을 수행할 수 있어야 합니다.

- 학습, 공유 및 개선

학습, 공유 및 개선

운영 활동 분석, 장애 분석, 실험 및 운영 방식 향상을 정기적으로 수행해야 합니다. 장애 발생 시에는 팀과 전체 엔지니어링 커뮤니티가 해당 장애로부터의 유용한 내용을 파악할 수 있어야 합니다. 그리고 장애를 분석하여 파악한 내용을 확인하고 운영 방식의 향상을 계획해야 합니다. 그리고 다른 팀이 파악한 내용도 정기적으로 검토하여 기존에 파악했던 내용을 다시 검증해야 합니다.

지속적 개선을 위한 프로세스 마련: 개선 기회를 정기적으로 평가하고 우선 순위를 지정해 가장 큰 이점을 얻을 수 있는 영역에서 작업을 중점적으로 수행합니다.

인시던트 이후 분석 수행: 고객에게 영향을 주는 이벤트를 검토하고 원인과 예방 조치 항목을 식별합니다. 이 정보를 사용하여 재발을 제한하거나 예방하는 완화 기능을 개발합니다. 신속하고 효과적인 응답 절차를 개발합니다. 목표 대상에 맞게 적절히 원인과 수정 조치를 전달합니다.

피드백 루프 구현: 개선이 필요한 영역과 문제를 확인할 수 있도록 절차와 워크로드에 피드백 루프를 포함합니다.

지식 관리 수행: 팀원이 적시에 원하는 정보를 검색하고 접근하며, 최신 상태의 완전한 정보인지 식별할 수 있는 메커니즘을 제공합니다. 필요한 콘텐츠, 갱신이 필요한 콘텐츠, 더 이상 참조되지 않도록 보관해야 하는 콘텐츠를 식별하는 메커니즘도 제공합니다.

개선 추진 요인 정의: 개선 기회를 평가하고 우선 순위를 지정할 수 있도록 개선 추진 요인을 파악합니다.

AWS 에서는 모든 운영 활동, 워크로드 및 인프라의 로그를 집계해 상세 활동 이력을 생성할 수 있습니다. 그런 후에는 AWS 도구를 사용하여 시간별 운영 및 워크로드 상태를 분석함으로써 추진 요인을 기반으로 향상 기회를 파악할 수 있습니다. 예를 들어, 트렌드를 파악하고, 이벤트/활동과 성과의 상관 관계를 지정하고, 여러 환경과 시스템을 비교/대조할 수 있습니다.

CloudTrail 을 사용해 AWS Management Console, CLI, SDK 및 API 를 통해 API 활동을 추적하여 모든 계정에서 수행되는 작업을 확인해야 합니다. CloudTrail 및 CloudWatch 를 사용하여 AWS 개발자 도구 배포 활동을 추적합니다. 이렇게 하면 배포의 자세한 활동 이력과 해당 결과가 CloudWatch Logs 로그 데이터에 추가됩니다.

장기간 저장을 위해 Amazon S3 로 로그 데이터를 내보냅니다. AWS Glue 를 사용하면 분석을 위해 Amazon S3 의 로그 데이터를 검색 및 준비할 수 있습니다. Glue 와의 기본 통합을 통해 Amazon Athena 에서 로그 데이터를 분석합니다. Amazon QuickSight 와 같은 비즈니스 인텔리전스 도구를 사용하면 데이터를 시각화하고 탐색하며 분석할 수 있습니다.

인사이트 검증: 여러 부문의 팀 및 비즈니스 책임자와 함께 분석 결과와 응답을 검토합니다. 이러한 검토에서는 개선 가능성을 공통적으로 파악하며, 추가적인 영향을 확인하고, 조치 과정을 결정할 수 있습니다. 필요에 따라 대응 내용을 조정합니다.

운용 지표 검토 수행: 다양한 비즈니스 영역의 경영진을 비롯해 여러 팀 구성원과 함께 인시던트 및 운영 지표의 후행 분석을 주기적으로 수행합니다. 이러한 검토에서는 개선 기회와 진행 가능한 조치 과정을 파악하고 배운 내용을 공유할 수 있습니다.

개발, 테스트, 프로덕션 등 모든 환경에서 향상 기회를 모색해야 합니다.

파악한 내용 공유 및 문서화: 운영 활동 실행 과정에서 파악한 내용을 문서화하고 공유하여 내부적으로, 그리고 여러 팀 간에 사용할 수 있도록 하십시오.

조직 전체에서 관련 이점을 더욱 효율적으로 활용하려면 팀에서 파악한 내용을 공유해야 합니다. 피할 수 있는 오류를 방지하고 개발 작업을 쉽게 수행하기 위해 정보와 리소스를 공유할 수 있습니다. 이렇게 하면 원하는 기능을 제공하는 데 집중할 수 있습니다.

AWS Identity and Access Management(IAM)를 사용하여 계정 내/계정 간에 공유할 리소스의 접근을 제어할 수 있는 권한을 정의합니다. 그런 다음, 버전 제어 AWS CodeCommit 리포지토리를 사용하여 애플리케이션 라이브러리, 스크립팅된 절차, 절차 설명서 및 기타 시스템 설명서를

공유해야 합니다. 여러 계정에 Lambda 함수 사용 권한을 부여하고 AMI 접근을 공유하는 방식을 통해 컴퓨팅 표준을 공유합니다. 인프라 표준도 CloudFormation 템플릿으로 공유해야 합니다.

AWS API 및 SDK 를 사용하면 GitHub, BitBucket, SourceForge 등의 외부 도구 및 서드파티 도구와 리포지토리를 통합할 수 있습니다.

파악한 내용과 개발한 기능을 공유할 때는 공유 리포지토리 무결성을 유지할 수 있도록 권한의 구조를 지정해야 합니다.

개선을 위한 시간 할애: 프로세스 내에서 리소스와 시간을 할애하여 가능한 범위 내에서 점진적이고 지속적으로 개선을 수행합니다.

AWS 에서는 실험과 테스트의 위험, 작업량 및 비용을 줄일 수 있는 환경의 임시 복제본을 생성할 수 있습니다. 이렇게 복제된 환경을 사용하여 분석의 결론을 테스트하고, 실험을 진행하고, 계획된 개선 사항을 개발/테스트할 수 있습니다.

리소스

경험을 통한 배움과 관련한 AWS 모범 사례에 대해 자세히 알아보려면 다음 리소스를 참조하십시오.

설명서

- Amazon VPC 흐름 로그 쿼리
- Amazon CloudWatch 도구를 사용하여 배포 모니터링
- Analyzing VPC Flow Logs with Amazon Kinesis Data Firehose, Amazon Athena, and Amazon QuickSight
- AWS CodeCommit 리포지토리 공유
- 리소스 기반 정책을 사용하면 다른 계정과 AWS 서비스에 Lambda 리소스를 사용할 수 있는 권한을 부여할 수 있습니다.
- 지정한 AWS 계정과 AMI 공유
- Amazon SNS 에서 AWS Lambda 사용

결론

운영 우수성은 지속적으로 반복해서 수행해야 하는 작업입니다.

목표를 공유하여 조직이 성공할 수 있도록 지원합니다. 모든 사람이 비즈니스 성과를 달성하는 데 자신의 역할을 파악하고 어떻게 다른 사람의 역량에 영향을 미치는지 이해해야 합니다. 비즈니스 성과를 달성할 수 있도록 팀원을 지원합니다.

모든 운영 이벤트와 장애는 아키텍처 운영을 개선할 수 있는 기회로 간주해야 합니다. 워크로드의 요구 사항을 파악하고, 일상적인 활동을 위한 런북과 문제 해결 과정을 안내하는 플레이북을 미리 정의합니다. 그리고, 코드 기능을 사용하여 AWS 를 운영하고, 상황을 지속적으로 인지하면 운영 준비 역량을 높이고 인시던트가 발생한 경우 더 효과적으로 대응할 수 있습니다.

요구 사항이 바뀌면 우선순위와 이벤트 대응 및 후행 분석에서 파악한 내용에 따라 증분식 개선을 집중적으로 수행합니다. 이러한 작업을 통해, 활동의 효율성을 높여 비즈니스 성공을 달성할 수 있습니다.

AWS 는 응답성이 뛰어난 적응형 배포를 구축하는 동시에 효율성을 최대화하는 아키텍처를 구축하고 운영할 수 있도록 지원합니다. 워크로드의 운영 우수성을 향상시키려면 이 백서에서 설명한 모범 사례를 사용해야 합니다.

기고자

- Brian Carlson, Well-Architected 운영 부문 담당자, Amazon Web Services
- Jon Steele, Sr. 기술 계정 관리자, Amazon Web Services
- Ryan King, 기술 프로그램 관리자, Amazon Web Services
- Philip Fitzsimons, Sr. Well-Architected 관리자, Amazon Web Services

추가 자료

자세한 내용은 다음 출처를 참조하십시오.

- AWS Well-Architected 프레임워크

문서 개정

날짜	설명
2020 년 4 월	새 AWS 서비스 및 기능과 최신 모범 사례를 반영하여 업데이트되었습니다.
2018 년 7 월	새 AWS 서비스 및 기능과 업데이트된 참조를 반영하여 업데이트되었습니다.
2017 년 11 월	초판