관찰성 결과 가속화

AWS 권장 가이드

AWS 권장 가이드: 관찰성 결과 가속화

Table of Contents

관찰성 결과 가속화

Jyothi Madanlal, Abhaya Chauhan, Jose Augusto Ferronato, Amazon Web Services

2025년 7월(문서 기록)

Amazon의 부사장 겸 최고 기술 책임자인 Werner Vogels 박사는 “모든 것이 항상 실패하므로 실패할 계획은 아무것도 실패하지 않을 것입니다”라고 말합니다(AWS re:Invent 2024 기조연설 프레젠테이션). 이러한 사고 방식을 채택하는 것은 효과적이고 강력한 관찰성 구현의 핵심입니다.

이 전략 문서에서는 관찰성 태세를 높여 더 탄력적이고 효율적인 시스템을 만드는 방법을 살펴봅니다. 또한 조직 전체에 효과적인 조정을 제공하여 고객 신뢰, 브랜드 평판 및 팀 사기를 높이는 앵커 목표(North Star)를 정의하는 방법도 설명합니다. 이 문서에서는 고려해야 할 주요 사항과 관찰성 결과를 가속화하기 위한 권장 접근 방식을 강조합니다. 이 접근 방식에는 새로운 비즈니스 기회 창출, 더 나은 사용자 경험 제공, 성능 및 비용 최적화, 조직 전반의 운영 효율성 개선이 포함됩니다. 관찰성 구현의 성숙도에 관계없이이 문서는 관찰성 목표를 추구하는 데 필요한 증분적이고 실행 가능한 인사이트를 제공합니다.

대상 독자

이 정보는 다음 대상을 대상으로 합니다.

- 관찰성 전략을 정의할 책임이 있는 기술 리더 및 아키텍트
- 운영 우수성 감독을 담당하는 엔지니어링 관리자 및 DevOps 리더
- 관찰성 기반 구축을 담당하는 플랫폼 엔지니어링 팀
- 관찰성 사례 구현을 담당하는 사이트 신뢰성 엔지니어(SREs)

목표

이 가이드의 정보는 관찰성 관행을 개선하여 조직 전체에 걸쳐 보다 탄력적이고 효율적인 시스템을 만드는 데 도움이 됩니다. 사람, 프로세스, 기술이라는 세 가지 원칙에 걸쳐 바람직한 관찰성 관행을 채택하기 위한 프레임워크를 제공합니다. 이 정보는 사용하기로 선택한 도구에 관계없이 광범위하게 적용됩니다. 관찰성을 지원하는 AWS 서비스에 관심이 있는 경우 AWS 웹 사이트의 모니터링 및 관찰성을 참조하세요.

개요

관찰성 전략을 재고해야 하는 이유

관찰성은 애플리케이션 디버깅에 도움이 되도록 로그, 지표, 추적과 같은 원격 측정 신호 모음에 초점을 맞춘 모니터링에서 발전했습니다. 이러한 연결로 인해 관찰성은 종종 사후 생각으로 인해 계측이 너무 많거나 너무 적고, 신호의 상관관계를 파악할 수 없으며, 가시성이 연결 해제되고, 종종 응집적으로 통합되지 않는 여러 도구가 생성되었습니다. 이로 인해 관찰성의 이점을 능가하는 것으로 보이는 가치 및 비용이 부족한 것으로 인식되었습니다. 비즈니스 관점에서 이러한 문제는 평균 식별 시간(MTTI), 평균 복구 시간(MTTR) 및 사용자 경험, 신뢰, 브랜드 평판 및 수익의 저하를 의미했습니다. 오늘날의 관찰성은 애플리케이션을 디버깅하고 진단하는 기능뿐만 아니라 애플리케이션이 의도한 대로 정확하게 작동하는지 검증하는 기능에 관한 것입니다.

사용자에게 최상의 경험과 관찰성 도구 및 기능의 발전을 제공하려는 비즈니스 간의 융합을 위해서는 관찰성을 재고하고 우선 순위를 다시 지정해야 합니다.

관찰성 도구 및 프레임워크

2019년에 OpenTelemetry를 사용할 수 있게 되기 전에는 애플리케이션 성능 모니터링(APM) 및 디지털 경험 모니터링(DEM)을 위한 관찰성 솔루션을 제공하는 특수 도구를 통해 원격 측정 신호 간의 연결 해제가 더 잘 표시되고 좋지 않은 사용자 경험을 강조 표시했습니다.

- APM은 소프트웨어 애플리케이션 동작을 실시간으로 추적하고 분석합니다. 애플리케이션 구성 요소 전반의 사용자 트랜잭션을 모니터링하면서 응답 시간, 오류율 및 리소스 사용량과 같은 주요 지표를 측정합니다. APM 도구를 사용하면 성능 문제, 병목 현상 및 오류가 사용자에게 영향을 미치기 전에 팀이 신속하게 식별할 수 있습니다. 주요 목표는 문제를 해결하는 데 필요한 시간을 줄이면서 최적의 애플리케이션 성능과 사용자 경험을 유지하는 것입니다.
- DEM은 사용자의 관점에서 디지털 서비스와의 상호 작용 품질을 측정하고 분석합니다. 실제 사용자 모니터링(RUM), 합성 모니터링 및 엔드포인트 모니터링을 결합하여 사용자 경험에 대한 전체 보기를 제공합니다. DEM은 다양한 디바이스, 브라우저 및 위치에서 페이지 로드 시간, 애플리케이션 응답성 및 사용자 여정 완료와 같은 지표를 추적합니다. 이를 통해 조직은 사용자가 디지털 서비스를 어떻게 경험하는지 이해하고, 사용자 만족도에 영향을 미치는 성능 문제를 식별하고, 디지털 접점을 최적화할 수 있습니다. 인사이트를 통해 기업은 데이터 기반 결정을 내려 고객 경험을 개선하고 경쟁 우위를 유지할 수 있습니다.

2019년에 OpenTelemetry가 출시 되면서 원격 측정 데이터를 생성, 수집, 관리 및 내보내기 위한 오픈 소스 통합 표준이 제공되었습니다. 이 프레임워크는 컨텍스트를 추가하고, 신호 간에 더 나은 상관관계를 제공하고, 더 나은 파생 가치를 제공하여 원격 측정 신호 간의 격차를 해소하는 데 중점을 둡니다. 예를 들어 컨텍스트가 추가된 구조화된 로그를 사용하면 수집된 로그에서 지표를 도출하고 다양한 방식으로 정보를 분석하여 근본 원인을 더 빠르게 파악할 수 있습니다. OpenTelemetry 이전에는 신호를 개별적으로 확인했습니다. 기능을 추가하려면 기존 지표에 새 차원을 추가하거나 새 지표를 생성하도록 코드를 수정하고 코드가 개발 수명 주기를 통과할 때까지 기다린 다음 적절한 환경에서 지표가 관찰될 때까지 기다렸다가 공제해야 합니다. 이 프로세스는 가시성을 지연시키고 필요한 경우 데이터를 로그 또는 추적과 연관시키는 기능에 영향을 미쳤습니다.

OpenTelemetry에 대한 지원과이 지원에서 비롯된 도구 개선 사항은 관찰성 플랫폼에서 더 나은 가치를 도출하고, 사용자 경험을 개선하고, 운영 효율성과 팀 사기를 개선하는 데 도움이 됩니다.

관찰성 태세를 개선하고 개선하려면 실제로 어디서 어떻게 시작해야 합니까? 이 가이드에서 자세히 설명하는 세 단계로 구성된 접근 방식을 권장합니다.

- 1단계: North Star 정의
- 2단계: 관찰성 구현
- 3단계: 검사, 조정 및 반복

1단계: North Star 정의

관찰성의 성공적인 구현은 운영 및 도구뿐만 아니라 소유권, 지속적인 개선 및 선제적 문제 해결 문화를 조성하는 것입니다. 성공적인 전략과 마찬가지로 관찰성 전략에는 사람, 프로세스, 기술이라는 세 가지 원칙을 전체적으로 고려해야 합니다.

관찰성 태세를 설정하거나 개선하려면 먼저 무엇이 중요한지 정의하고, 비즈니스 성과에서 다시 작업하고, 비즈니스, 팀 및 제품이 발전함에 따라 전략을 지속적으로 검토, 조정 및 재조정하는 것이 좋습니다.

이 첫 번째 단계에서는 North Star를 정의하고 설정합니다. North Star는 조직에 좋은 모습이 무엇인지에 대해 합의되고 잘 이해된 정의입니다. 비즈니스가 발전하거나, 새 제품, 애플리케이션 또는 서비스를 시작하거나, 주요 아키텍처 변경을 설계할 때이 단계의 일부 또는 모든 활동을 다시 검토하여 관찰성 플랫폼과 조직 요구 사항을 재평가하는 것이 좋습니다.

개발 수명 주기 초기에 관찰성 통합(교대-왼쪽 접근 방식)

관찰성을 엔지니어링, 운영 및 제품 팀의 모든 구성원의 책임으로 삼고 단위 테스트 또는 보안을 처리하는 방식과 유사한 주요 기능 요구 사항으로 취급합니다. 이는 운영 팀에서 개발 팀으로 책임을 이전하지 않지만 여러 팀에서 필요한 협업을 강조합니다. 팀이 개발 수명 주기 초기에 공동 작업으로 다음 활동을 수행하는 것이 좋습니다. 이러한 작업은 티켓별, 기능별 또는 제품별로 수행할 수 있습니다.

- 이해관계자를 식별합니다. 이해관계자는 누구이며이 기능이나 제품이 예상대로 작동하지 않는 경우 이해관계자에게 중요한 것은 무엇입니까? 이해관계자를 식별할 때는 기능, 가용성, 보안, 비용, 판매 및 제품 사용과 같은 측면을 고려하세요. 이해관계자에는 팀, 제품 고객, 내부 비즈니스 이해관계자, 플랫폼 운영 팀 구성원, 애플리케이션 개발자가 포함될 수 있습니다. 시나리오에 따라 보안 및 재무 팀도 이해관계자가 될 수 있습니다.
- 주요 결과를 식별합니다. 주요 성과와 주요 성과가 비즈니스 및 각 이해관계자에게 미치는 영향을 결정합니다. 각 성과 및 이해관계자의 성공과 실패를 식별합니다. 결과는 일반적으로 서비스 수준 목표(SLOs)로 정의되며 정량화 가능해야 합니다. SLO는 각 결과에 대한 척도입니다. 좋은 SLO에는 목표로 노력하거나 유지해야 하는 목표 값이 있습니다. SLO는 사용자 만족도의 척도일 수 있습니다. 서비스 수준 지표(SLI)는 SLO를 충족하는지 확인하는 데 사용되는 실제 측정 또는 지표입니다. 이는 목표에 대해 추적하는 정량화 가능한 데이터 포인트입니다. 예를 들어 MTTR을 60% 줄이거나, 애플리케이션 가용성을 99.99%로 유지하거나, 개발자 생산성을 30% 개선합니다.

  애플리케이션 가용성을 99.99%로 유지하는 예를 살펴보고 성공을 측정하고 검증하는 데 필요한 SLO, SLI 및 지표를 정의해 보겠습니다. 이 예제에서는 RESTful 애플리케이션을 고려하고 애플리케

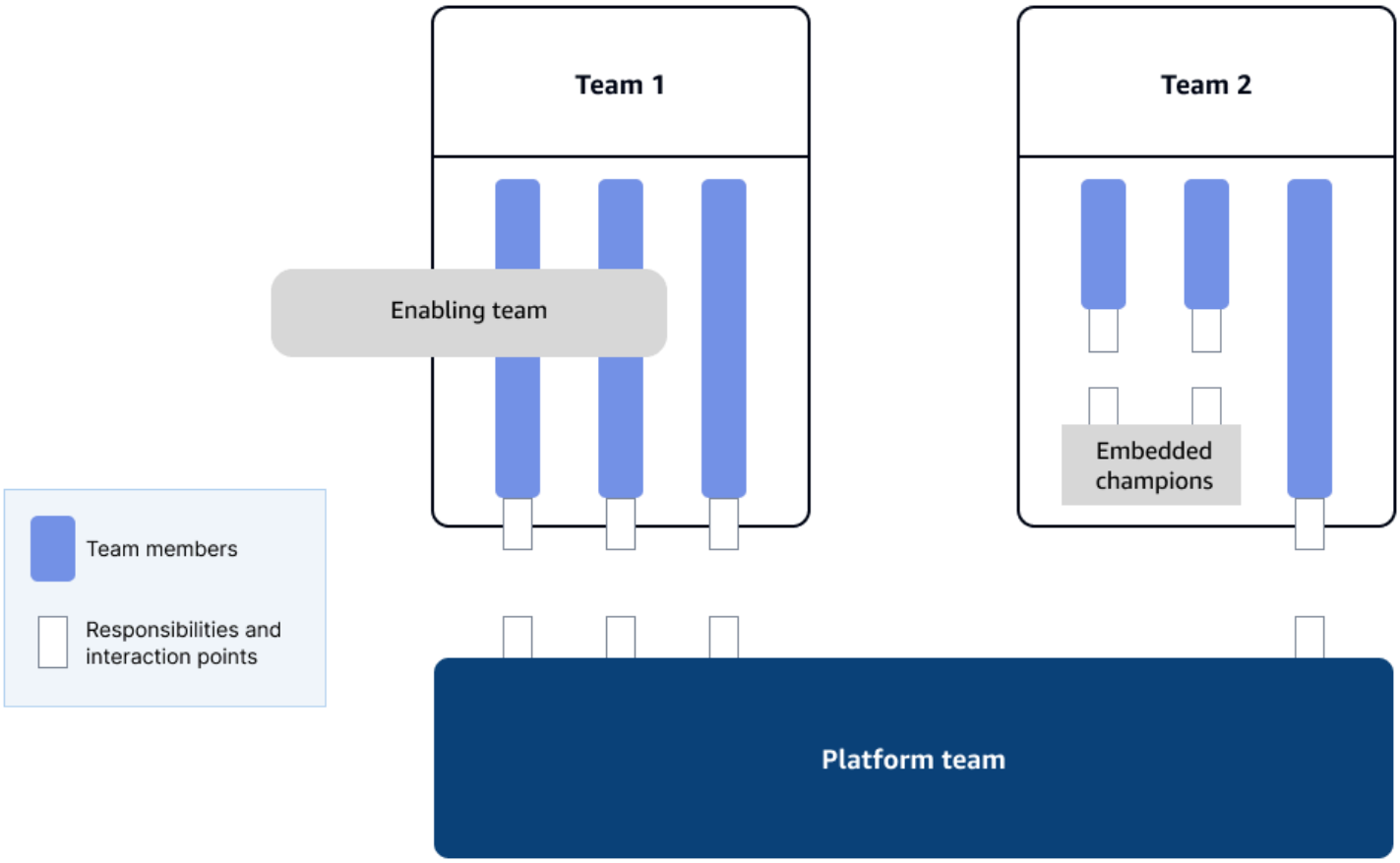
이션 가용성을 모든 수신 요청의 성공적인 완료로 정의해 보겠습니다. 이를 위해서는 애플리케이션에 대한 총 요청 수와 각 요청의 완료 상태를 측정해야 합니다. 이를 SLO 및 SLI로 변환할 때는 수신 요청을 캡처하는 지표 하나와 요청 상태를 캡처하는 지표 하나가 필요합니다. 모든 요청이 성공적으로 완료되면 애플리케이션을 사용할 수 있는 것으로 간주됩니다. 하나 이상의 요청에 오류가 발생하면 애플리케이션을 사용할 수 없는 것으로 간주됩니다. 따라서 SLI는 오류 상태인 요청 완료의 합계를 5분 간격으로 수신되는 요청의 합계로 나눈 값입니다. 실제로 오류율입니다. 이 SLI에 목표를 추가하여 SLO로 전환할 수 있습니다. 예를 들어 3회 연속 5분 간격에서 오류율이 0.1% 미만이 되도록 Strive를 실행할 수 있습니다.

- 주요 결과의 우선순위를 정합니다. 각 결과에 설정한 우선 순위에 따라 모든 작업을 동시에 수행하는 대신 가장 큰 영향을 미치는 결과에 먼저 집중하도록 선택할 수 있습니다. 작게 시작하고 반복하여 관찰성 태세를 조금씩 개선합니다. 관찰성은 성숙도와 이점을 높이기 위해 지속적인 검토, 감사, 개선 및 개선이 필요한 프로세스입니다. 또한 우선순위 지정을 통해 식별된 결과에 대한 점진적 마일스톤을 정의할 수 있습니다.
- 필요한 계측을 식별합니다. 이전 단계에서 식별된 대로 중요한 결과에 영향을 미칠 수 있는 아키텍처 또는 구현의 구성 요소 및 관련 기능은 무엇입니까? 예를 들어 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에서 애플리케이션을 실행하면 코어 수와 사용 가능한 RAM이 애플리케이션의 응답성과 처리량에 영향을 미칠 수 있습니다. 이 단계에서는 사용하는 도구 또는 라이브러리가 이미이 계측 중 일부를 제공하는지 확인하는 것도 도움이 될 수 있습니다. 일련의 예비 검토를 수행하거나 티켓의 준비 완료(DoR) 정의에 다음과 같은 질문을 추가하면이 활동이 표준 프로세스의 일부가 될 수 있습니다.
  - 이 작업이 실패할 경우 실패를 해결하기 위해 알아야 할 사항은 무엇입니까? 일반적인 작업이나 문제가 있는 작업은 관련된 구성 요소에 어떤 영향을 미치나요? 로그, 지표 또는 추적 등이 작업은 어떤 종류의 신호를 전송해야 합니까? 값과 비교하여이 계측의 비용은 얼마입니까? SLOs를 위반하지 않으면 어떤 종류의 집계가 허용됩니까?
  - 이 작업에서 장애를 일으킬 수 있는 구성 요소와 종속성은 무엇입니까? 어떤 구성 요소 또는 종속성이 장애를 일으켰는지 어떻게 식별할 수 있습니까? 이러한 구성 요소와 종속성의 다양한 구성 레버는 무엇이며 각각 작업에 어떤 영향을 미치나요?
  - SLI 및 SLO를 정확하게 측정하는 데 필요한 지표 세부 수준 및 샘플링 속도는 얼마입니까?
- 성공 기준을 정의합니다. 우선순위가 지정된 각 결과에 대해 목표 달성 여부에 따른 영향과 일치하는 임계값을 정의합니다. 성공 기준은 팀이 알림에 응답할 때 팀에 추가 컨텍스트를 제공합니다. 또한 필요한 가시성을 위해 계측 비용을 예측하고 절충할 수 있는 기능도 제공합니다.

효과적인 조직 및 팀 구조 설정

아키텍처 복잡성과 비즈니스 규모에 따라 관찰성에 초점을 맞춘 전담 팀을 구성해야 할 수 있습니다. 이 팀은 관찰성 도구를 구성하고 다른 팀을 위한 관찰성 플랫폼을 설정할 책임이 있습니다. 또한 표준 OpenTelemetry 구현을 선택하는 경우 전용 팀을 설정하는 것이 좋습니다. 소규모 조직에서는 관찰성을 모든 팀원의 추가 책임으로 할당하고 팀 전체에서 모범 사례를 전파하고 적용하는 관찰성 챔피언을 지정할 수도 있습니다. 이러한 챔피언은 하루의 일부를 자원하여 프로세스를 정의하고 조직의 표준을 설정합니다. 이들은 자율 조정 팀으로 근무하거나 전담 관찰성 전문가가 주도할 수 있습니다. 다음 다이어그램은 투자가 조직의 접근 방식을 결정하는 방법을 보여줍니다.

챔피언은 팀에 완전히 포함되거나(다음 그림의 팀 2에 나와 있음) 팀 전체에서 순환하여 모범 사례를 수립하고 홍보하는 지원 팀의 일원이 될 수 있습니다(그림의 팀 1).



비용 할당 추적

조직은 리소스 사용 및 비용에 대한 팀별 책임을 설정하면서 지표, 로그 및 추적 전반에 걸쳐 포괄적인 비용 추적 및 가시성을 구현해야 합니다. 재무 운영(FinOps) 사례를 성공적으로 통합하려면 체계적인 데이터 보존 및 수집 최적화와 결합된 예산 알림이 포함된 자동 모니터링 시스템이 필요합니다. 엔지니어링 및 재무 팀은 공유 대시보드와 정기적인 검토를 통해 목표를 조정해야 합니다. 조직은 명확한 차지백 모델과 비용 할당 전략을 구현하여 소유권과 책임을 높일 수 있습니다.

표준 정의

알림 및 대시보드 전략을 포함하여 애플리케이션에 필요한 기본 신호와 원격 측정을 식별하고 정의합니다. 모든 애플리케이션에 대한 체크리스트 또는 공식 검토 프로세스를 생성합니다. AWS 관찰성 모범 사례 웹 사이트에서는 적절한 알림 임계값 설정, 알림 피로 최소화, 각 페르소나에 대해 충분한 컨텍스트가 있는 대시보드 생성 등과 같은 알림 및 대시보드 생성 지침을 제공합니다. 연결되고 선별된 관찰성 경험은 Amazon CloudWatch 설명서의 애플리케이션 신호를 참조하세요.

에스컬레이션 프로세스 설정

에스컬레이션 메커니즘, 알림 소유권 및 대응 절차를 설정하고 적용하는 것이 중요합니다. 에스컬레이션이 중단되지 않는 문화를 장려하는 것이 좋습니다.

훈련을 통한 기술 향상

기존 및 신규 팀원의 기술을 향상시키고, 관찰성의 중요성을 강화하고, 지속적인 개선 문화를 조성하는 가장 좋은 방법을 식별합니다. 조직의 요구 사항에 따라 관찰성 챔피언 또는 전문가가 제공하는 사전 녹화된 온디맨드 교육 또는 강의실 교육 중에서 선택할 수 있습니다. AWS 계정 팀은 One Observability 워크숍 또는 GameDays와 같은 심층적인 실습 훈련 세션을 제공하여 관찰성 기술과 모범 사례를 지도하고 개선할 수 있습니다. 또한 모범 사례를 강화하고 조직에서 정의한 표준을 홍보하는 메커니즘을 통합합니다.

2단계: 관찰성 구현

이 단계에서는 팀이 North Star로 점진적으로 이동하는 프로세스를 시작합니다.

관찰성 플랫폼 선택

첫 번째 단계는 신호를 수집, 시각화 및 분석하고 알림을 전송하는 데 적합한 도구를 식별하는 것입니다. 도구를 선택할 때 기능 세트, 라이선스 모델, 가격, 기술 요구 사항 및 유지 관리를 고려합니다.

기능 세트

고려해야 할 몇 가지 질문은 다음과 같습니다.

- 구성 가능성 및 사용자 지정. 도구는 조사 경험을 간소화하고 MTTR을 줄이는 데 도움이 되는 어떤 기능을 제공합니까? 도구가 경보 상관관계, 지표 수학, 누락된 원격 측정 처리의 유연성 또는 이상 탐지를 제공합니까?
- 세분화. 원격 측정 신호 수집 및 시각화에서 지원되는 세분화는 무엇입니까?
- 페르소나. 도구가 개발자, 플랫폼 엔지니어 및 기타 페르소나에게 제공하고자 하는 경험을 지원하나요? 기술 페르소나와 비즈니스 페르소나 모두에서 작동하나요?
- 위젯. 대시보드는 어떤 종류의 위젯을 지원하나요? 도구에서 사용자 지정 위젯 생성을 허용하나요?
- 사전 구축된 솔루션. 도구에서는 가치 실현 시간을 줄이기 위해 어떤 종류의 사전 구축된 관찰성 솔루션을 제공하나요?
- 자동화 및 생성형 AI. 이 도구는 사용자와 팀의 토일을 자동화하거나 줄이는 데 도움이 되는 어떤 기능을 제공하나요? 예를 들어 자동 이상 탐지, 예측 분석 및 기타 생성형 AI 기능은 가정 및 알 수 없는 것(즉, 인식하지 못하거나 완전히 이해하지 못하는 것)의 스트레스를 줄이는 데 도움이 될 수 있습니다. 도구가 생성형 AI/ML 모델을 사용하여 대규모 데이터 분석을 개선하는 것을 지원하나요? AIOps를 자동화하고 구현할 수 있는 옵션이 제공되나요?
- 보안. 도구는 어떤 종류의 인증 및 권한 부여 통합을 지원하나요? 사용자 및 로그인 경험이 조직의 요구 사항을 충족하나요?
- OpenTelemetry 지원. 도구와 에이전트가 OpenTelemetry를 지원하나요? 대부분의 관찰성 플랫폼은 OpenTelemetry 호환 신호 수집을 지원하지만 모든 에이전트가 이러한 신호를 관찰성 플랫폼에 전달하는 구성 옵션을 제공하는 것은 아닙니다.
- 통합. 도구는 어떤 통합을 제공하나요? Slack에게 알림을 전송해야 하는지, 팀원을 호출해야 하는지 또는 해결을 자동화해야 하는지 고려합니다.

- 확장성. 도구의 확장성과 성능은 어느 정도입니까? 관찰성 솔루션은 수요와 사용량이 증가함에 따라 확장해야 하므로 애플리케이션을 사용할 수 없는 경우에도 진단을 제공할 수 있습니다.
- 지원. 어떤 지원 모델이 제공되나요? MTTR 및 애플리케이션 가용성 목표 또는 서비스 수준 계약(SLAs)을 충족할 수 있도록 애플리케이션이 실패하더라도 관찰성 도구를 사용할 수 있어야 합니다. 오픈 소스 솔루션은 제한된 공식 지원을 제공할 수 있습니다.

라이선스 및 배포 모델

솔루션의 라이선스 모델(오픈 소스 또는 상용) 및 배포 모델(자체 호스팅 또는 클라우드 기반)을 고려합니다. 오픈 소스 옵션은 종종 선결제 비용이 낮지만 가치를 제공하기 전에 배포, 설정 및 구성, 유지 관리 및 팀 업그레이드에 더 많은 시간이 필요할 수 있습니다. 오픈 소스 옵션을 고려하는 경우 관찰성 전문가로 구성된 전담 팀이 필요할 수 있습니다. 상용 소프트웨어는 일반적으로 더 높은 선결제 비용으로 더 빠른 가치 실현 시간을 제공하며, 채택, 복잡성 및 성숙도가 증가함에 따라 전담 관찰성 팀의 필요성이 시간이 지남에 따라 진화합니다. 자체 호스팅 솔루션은 클라우드 기반 솔루션에 비해 배포, 설정 및 구성, 유지 관리 및 운영 오버헤드에 더 많은 시간이 필요합니다.

차원 사용

애플리케이션이 새로운 사용자, 더 큰 아키텍처 공간 또는 새로운 기능 및 애플리케이션을 확보함에 따라 도구의 요금 모델이 총 소유 비용(TCO)에 어떤 영향을 미칠까요? 예를 들어, 일부 일반적인 라이선스 모델은 영구적이거나 구독, 명명된 사용자 수, 소비 또는 볼륨을 기반으로 합니다. 애플리케이션과 관찰성 도구가 사용량을 어떻게 확장하고 라이선스 모델이 도구 비용에 어떤 영향을 미칠 수 있는지 고려합니다.

팀 스킬

팀의 현재 스킬 세트와 성숙도에 따라 필요한 기술 향상 정도를 결정해야 합니다. 공급업체에서 팀을 교육하기 위해 어떤 종류의 지원을 제공하는지 고려합니다. 또한 조직 구조가 선택한 도구의 구성 및 관리를 지원하는지 여부를 고려합니다. 예를 들어 OpenTelemetry를 선택하는 경우 관찰성을 전문으로 하는 별도의 팀을 설정하는 것이 좋습니다.

운영 및 유지 관리

다음 질문을 평가합니다.

- 관찰성 에이전트 또는 수집기는 어떤 배포 옵션을 제공합니까? 이러한 옵션이 아키텍처의 요구 사항을 충족하나요? 예를 들어 관찰성 도구에 컨테이너화된 배포를 사용하는 경우 데몬 세트 또는 사이

드카를 지원하나요? 보안 및 기타 모든 프로세스에 부합하도록 운영 팀이 수행하거나 사용해야 하는 추가 단계 또는 도구는 무엇입니까?

- 솔루션을 유지 관리하는 데 필요한 노력은 무엇입니까? 에이전트 또는 수집기를 업데이트하는 프로세스는 얼마나 간단하거나 자동화되어 있습니까? 에이전트 또는 수집기의 관리는 동일하게 유지되지만 완전 관리형 및 클라우드 기반 관찰성 인터페이스는 일반적으로 자체 배포 및 호스팅 애플리케이션에 비해 운영 오버헤드가 낮습니다. 팀 구조를 고려하고 선택한 옵션을 유지 관리하는 데 드는 인적 비용을 고려합니다.

애플리케이션 계측

이전 섹션의 질문에 대한 답변은 애플리케이션을 계측하는 데 필요한 정보, 즉 애플리케이션에 원격 측정 신호를 캡처하고 동작을 측정, 관찰 및 검증하는 코드를 추가하는 데 필요한 정보를 제공합니다. 애플리케이션 언어에 대한 OpenTelemetry SDKs와 같은 SDK를 사용하여 애플리케이션을 자동으로 계측할 수 있습니다. 격차를 해소하고 end-to-end 가시성을 보장하기 위해 수동 계측 코드를 추가해야 할 수도 있습니다. 추가하는 원격 측정에 대해 의도적으로 생각하고 이전 단계에서 설정한 하나 이상의 SLIs 및 SLOs에 다시 연결할 수 있는지 확인합니다.

원격 측정 수집

1단계에서 우선순위를 지정한 결과에 따라 관련 원격 측정 신호를 수집하도록 원격 측정 수집기 또는 에이전트를 구성합니다.

관찰성 구성 요소 구현

원격 측정이 흐르고 관찰성 플랫폼으로 수집되면 대시보드, 알림, 플레이북 및 런북을 생성합니다.

- 대시보드: 우선순위가 지정된 결과와 관련된 현재 및 과거 추세의 시각적 표현을 포함하여 관련 정보가 포함된 대시보드를 생성합니다. 1단계에서 정의한 이해관계자가 이러한 대시보드를 사용할 수 있도록 합니다. 자세한 내용은 Amazon Builders' Library 웹 사이트에서 운영 가시성을 위한 대시보드 구축을 참조하세요.
- 알림: 결과가 위험하거나 위반될 때 팀에 알리도록 알림을 정의합니다. 보안 및 성능 문제에 대한 알림을 추가하는 것이 좋습니다. 다음을 채택하여 알림을 최적화하여 알림 피로와 비용을 줄입니다.
  - 이상 탐지를 사용하면 자주 조정해야 하는 하드 임계값을 설정하지 않고 알 수 없는 항목의 발생을 줄일 수 있습니다.
  - 각 지표에 대해 개별 알림을 설정하는 대신 여러 관련 지표를 함께 살펴보는 지능형 알림 조합을 사용합니다. 예를 들어 CPU, 메모리 및 응답 시간에 대해 별도의 알림을 설정하는 대신 이러한 지

표가 집합적으로 실제 문제를 나타내는 경우에만 트리거되는 하나의 의미 있는 알림을 생성합니다. 이 접근 방식은 알림 노이즈를 크게 줄이고 팀이 격리된 지표 스파이크에 대응하는 대신 서비스에 영향을 미치는 실제 문제에 집중할 수 있도록 도와줍니다.

  - 사용자 경험 또는 결과가 위험한 경우에만 알림을 생성합니다. 예를 들어 애플리케이션에 활성 사용자가 없는 경우 자동 업그레이드로 인한 CPU 급증에 대한 알림을 받지 마세요.
- 플레이북: 플레이북은 인시던트 또는 알림에 대응하는 사람에게 사고 모델과 컨텍스트를 제공하고 근본 원인을 더 빠르게 식별하는 데 도움이 됩니다. 고도로 결합되고 복잡한 애플리케이션과 계측이 부족하지만 1단계에서 식별하고 우선순위를 지정한 결과에 직접적인 영향을 미치는 애플리케이션을 위한 플레이북을 만드는 것이 좋습니다.
- 런북: 런북을 사용하여 인시던트 또는 알림을 해결하는 데 필요한 단계를 정의합니다.

관찰성 시스템 검증

소프트웨어 개발 수명 주기(SDLC) 동안 대시보드가 시스템 테스트 중에 예상되는 동작과 업데이트를 제공하는지 확인합니다. 카오스 엔지니어링을 구현하고 플레이북 및 런북에 문서화된 단계를 검증하여 정확성과 용도를 충족하는지 확인합니다. 또한 알림 소유권 및 에스컬레이션 경로를 검증해야 합니다.

3단계: 검사, 조정 및 반복

관찰성 시스템을 구현한 후에는 구현을 지속적으로 검토, 평가, 학습, 조정 및 개선하는 것이 좋습니다. AWS 관찰성 성숙도 모델을 도구로 사용하여 구현의 성숙도를 평가하고 개선이 필요한 영역을 식별하고 우선순위를 지정할 수 있습니다.

정기 검토 구현

관찰성은 반복적인 프로세스입니다. 이를 위해서는 기존 구성 요소에 대한 정기적인 감사 및 평가와 지속적인 개선을 위한 변경 및 개선이 필요합니다. 정기적인 검토를 수행하여 SLOs, 알림 임계값, 대시보드, 지표 세분화, 보존 정책, 샘플링 전략 등을 재평가하여 팀과 비즈니스의 가치를 높이는지 확인하는 것이 좋습니다. 관찰성 비용을 특정 팀 및 서비스에 연결하여 적용 범위 및 리소스 할당에 대한 데이터 기반 결정을 활성화할 수 있습니다.

Amazon은 주간 운영 준비 검토(ORRs)를 수행하여 모범 사례에 대한 팀의 프로세스 및 관찰성 태세를 감사합니다. 이는 Amazon의 서비스 수 및 릴리스 빈도에 맞는 비차단 연습입니다.

조직의 규모에 따라 각 팀의 구성원 한 명이 이상 및 추세를 보고하고, 알려지지 않은 사항을 발견하고, 원치 않는 계측 및 알림을 제거하고, 대시보드를 개선하고, 관찰성 솔루션이 팀에 계속 작동하고 팀의 목표 및 성공 지표에 부합하는지 확인할 책임이 있는 평소와 같은 비즈니스(BAU) 명단을 보유할 수도 있습니다. 또한 알림 전략을 재평가하여 보다 대응력이 뛰어나고 선제적이며 사용자에게 더 가까워질 수 있습니다. 이러한 검토의 목표는 다음 그림과 같이 선순환을 생성하고 관찰성 성숙도 AWS 모델에 설명된 대로 관찰성 태세 성숙도의 성숙도를 개선하는 것입니다.

가장 자주 액세스하는 플레이북을 식별하고 애플리케이션을 개선하거나 계측을 추가하는 것이 좋습니다. 가장 자주 실행되는 런북을 식별하고 해당 런북을 자동화하는 것이 좋습니다.

이러한 리뷰의 학습 내용은 관찰성 스쿼드 및 전문가와도 공유되어 중앙 프로그램 및 관찰성 플랫폼의 개선 사항을 강조합니다. 예를 들어 배포 트리거 이벤트의 빈도에 따라 다른 구성 요소보다 배포 파이프라인 개선의 우선순위를 지정할 수 있습니다. 모니터링 격차로 인해 MTTR이 더 높은 경우 관찰성 플랫폼 및 해당 구성 개선의 우선순위를 지정할 수 있습니다.

승자 축하

관찰성 도구를 사용하는 팀의 성공 사례를 공유합니다. 예를 들어 관찰성 지표를 사용하여 보다 효율적이고 지연 시간이나 비용을 낮추는 대체 솔루션을 구현한 팀의 성공을 강조합니다. 이러한 성공을 알리는 것은 관찰성의 중요성을 강조하고 다른 팀이 관찰성 태세를 개선하고 유사한 성공을 위해 노력하도록 동기를 부여합니다.

인시던트에서 알아보기

Amazon에서 오류 수정(COE) 프로세스와 유사한 비난 없는 인시던트 후 연습을 수행하여 개선이 필요한 영역을 식별하고 향후 문제를 방지합니다. 성공과 마찬가지로이 연습의 학습 내용을 다른 팀과 광범위하게 공유하여 관찰성과 모범 사례의 가치를 강화할 수 있습니다.

다음 단계 및 리소스

사용자 및 비즈니스 성과부터 시작하여 영향에 집중하면 비용을 제어하고 도구 확산을 방지하는 동시에 관찰성 솔루션에서 더 나은 가치를 도출할 수 있습니다. 관찰성은 대상이 아닌 반복적인 프로세스입니다. 잘 정의된 목표를 가이드로 사용하여 관찰성 태세를 지속적으로 반복 및 개선하고, 모범 사례를 통합하고, 아키텍처, 비즈니스 요구 사항 및 팀이 발전함에 따라 격차를 좁히도록 팀과 프로세스를 설정합니다. 실패를 계획하고, 지속적인 개선 문화를 통합하고,이 가이드에서 설명하는 모범 사례를 채택하면 Werner Vogels 박사의 지침을 연습할 수 있습니다. “모든 것이 항상 실패합니다. 따라서 실패 계획을 세우고 아무것도 실패하지 않습니다.”

관찰성 전략을 정의하는 데 도움이 되는 실습 워크숍을 보려면 담당자에게 문의하여 AWS 관찰성 전략 워크숍을 제공하십시오.

리소스

효과적인 관찰성 전략을 구현하기 위한 자세한 내용과 지침은 다음 리소스를 AWS참조하세요.

- 모니터링 및 관찰성(AWS 관찰성 서비스)
- AWS 관찰성 모범 사례
- AWS OpenTelemetry용 배포판
- 운영 우수성 원칙(AWS Well-Architected Framework)
- 운영 가시성을 위한 대시보드 구축(Amazon Builders' Library)
- 관찰성(DevOps 지침, AWS Well-Architected Framework)

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	—	2025년 7월 16일

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