Amazon CloudWatch를 사용한 로깅 및 모니터링 설계 및 구현

AWS 권장 가이드

AWS 권장 가이드: Amazon CloudWatch를 사용한 로깅 및 모니터링 설계 및 구현

Table of Contents

소개 1
목표 비즈니스 성과 5
운영 준비 가속화 5
운영 우수성 개선 5
운영 가시성 향상 5
운영 규모 조정 및 오버헤드 비용 절감 6
CloudWatch 배포 계획 7
중앙 집중식 또는 분산 계정에서 CloudWatch 사용 7
CloudWatch 에이전트 구성 파일 관리 10
CloudWatch 구성 관리 11
예: S3 버킷에 CloudWatch 구성 파일 저장 13
EC2 인스턴스 및 온프레미스 서버에 대한 CloudWatch 에이전트 구성 15
CloudWatch 에이전트 구성 15
EC2 인스턴스에 대한 로그 캡처 구성 16
EC2 인스턴스에 대한 지표 캡처 구성 18
시스템 수준 CloudWatch 구성 20
시스템 수준 로그 구성 20
시스템 수준 지표 구성 22
애플리케이션 수준 CloudWatch 구성 23
애플리케이션 수준 로그 구성 23
애플리케이션 수준 지표 구성 24
Amazon EC2 및 온프레미스 서버에 대한 CloudWatch 에이전트 설치 접근 방식 26
Systems Manager Distributor 및 State Manager를 사용하여 CloudWatch 에이전트 설치 26
CloudWatch 에이전트 배포 및 구성을 위한 State Manager 및 Distributor 설정 28
Systems Manager 빠른 설정을 사용하고 생성된 Systems Manager 리소스를 수동으로 업데이트합니다. 29
빠른 설정 CloudFormation 대신 사용 30
CloudFormation 스택이 있는 단일 계정 및 리전의 사용자 지정 빠른 설정 31
CloudFormation StackSets를 사용한 여러 리전 및 여러 계정의 사용자 지정 빠른 설정 32
온프레미스 서버 구성 시 고려 사항 33
임시 EC2 인스턴스에 대한 고려 사항 35
자동화된 솔루션을 사용하여 CloudWatch 에이전트 배포 35
사용자 데이터 스크립트를 사용하여 인스턴스 프로비저닝 중에 CloudWatch 에이전트 배포 36
AMIs에 CloudWatch 에이전트 포함 36

Amazon ECS에서 로깅 및 모니터링 38
EC2 시작 유형으로 CloudWatch 구성 38
EC2 및 Fargate 시작 유형에 대한 Amazon ECS 컨테이너 로그 40
FireLens for Amazon ECS에서 사용자 지정 로그 라우팅 사용 41
Amazon ECS에 대한 지표 41
Amazon ECS에서 사용자 지정 애플리케이션 지표 생성 42
Amazon EKS의 로깅 및 모니터링 44
Amazon EKS에 대한 로깅 44
Amazon EKS 컨트롤 플레인 로깅 45
Amazon EKS 노드 및 애플리케이션 로깅 45
Fargate의 Amazon EKS에 대한 로깅 47
Amazon EKS 및 Kubernetes에 대한 지표 48
Kubernetes 컨트롤 플레인 지표 48
Kubernetes에 대한 노드 및 시스템 지표 48
애플리케이션 지표 49
Fargate의 Amazon EKS 지표 50
Amazon EKS에서 Prometheus 모니터링 51
에 대한 로깅 및 지표 AWS Lambda 53
Lambda 함수 로깅 53
CloudWatch에서 다른 대상으로 로그 전송 54
Lambda 함수 지표 54
시스템 수준 지표 55
애플리케이션 지표 55
Searching and analyzing logs in CloudWatch 56
CloudWatch Application Insights를 사용하여 애플리케이션을 종합적으로 모니터링 및 분석 56
CloudWatch Logs Insights를 사용하여 로그 분석 수행 58
Amazon OpenSearch Service를 사용하여 로그 분석 수행 60
CloudWatch를 사용한 경보 옵션 63
CloudWatch 경보를 사용하여 모니터링 및 경보 63
CloudWatch 이상 탐지를 사용하여 모니터링 및 경보 64
여러 리전 및 계정에서 경보 발생 64
EC2 인스턴스 태그를 사용하여 경보 생성 자동화 65
애플리케이션 및 서비스 가용성 모니터링 66
를 사용하여 애플리케이션 추적 AWS X-Ray 67
Amazon EC2에서 애플리케이션 및 서비스를 추적하기 위한 X-Ray 데몬 배포 67

Amazon ECS 또는 Amazon EKS에서 애플리케이션 및 서비스를 추적하기 위한 X-Ray 데몬 배포 68
요청을 X-Ray로 추적하도록 Lambda 구성 68
X-Ray용 애플리케이션 계측 69
X-Ray 샘플링 규칙 구성 69
Dashboards and visualizations with CloudWatch 70
교차 서비스 대시보드 생성 70
애플리케이션 또는 워크로드별 대시보드 생성 70
교차 계정 또는 교차 리전 대시보드 생성 71
지표 수학을 사용하여 관찰성 및 경보 미세 조정 71
CloudWatchContainer Insights 및 CloudWatch Lambda Insights와 함께 Amazon ECS, Amazon EKS 및 Lambda에 자동 대시보드 사용 72
AWS 서비스와 CloudWatch 통합 73
대시보드 및 시각화를 위한 Amazon Managed Grafana 74
FAQ 77
CloudWatch 구성 파일은 어디에 저장하나요? 77
경보가 발생할 때 서비스 관리 솔루션에서 티켓을 생성하려면 어떻게 해야 합니까? 77
CloudWatch를 사용하여 컨테이너의 로그 파일을 캡처하려면 어떻게 해야 하나요? 77
AWS 서비스의 상태 문제를 모니터링하려면 어떻게 해야 합니까? 78
에이전트 지원이 없는 경우 사용자 지정 CloudWatch 지표를 생성하려면 어떻게 해야 합니까? ... 78
기존 로깅 및 모니터링 도구를와 통합하려면 어떻게 해야 하나요 AWS? 78
리소스 79
소개 79
목표 비즈니스 성과 79
CloudWatch 배포 계획 79
EC2 인스턴스 및 온프레미스 서버에 대한 CloudWatch 에이전트 구성 79
Amazon EC2 및 온프레미스 서버에 대한 CloudWatch 에이전트 설치 접근 방식 80
Amazon ECS에서 로깅 및 모니터링 80
Amazon EKS의 로깅 및 모니터링 81
에 대한 로깅 및 지표 AWS Lambda 81
Searching and analyzing logs in CloudWatch 82
CloudWatch를 사용한 경보 옵션 82
애플리케이션 및 서비스 가용성 모니터링 83
를 사용하여 애플리케이션 추적 AWS X-Ray 83
Dashboards and visualizations with CloudWatch 83
CloudWatch와 AWS 서비스 통합 83

대시보드 및 시각화를 위한 Amazon Managed Grafana 83
문서 기록 85
용어집 86
..... 86
A 87
B 89
C 91
D 94
E 98
F 100
G 101
H 102
I 104
L 106
M 107
O 111
P 113
Q 116
R 116
S 119
T 122
U 124
V 124
W 125
Z 126
..... cxxvii

Amazon CloudWatch를 사용한 로깅 및 모니터링 설계 및 구현

Khurram Nizami, Amazon Web Services(AWS)

2023년 4월(문서 기록)

이 가이드는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스, Amazon Elastic Container Service(Amazon ECS), Amazon Elastic Kubernetes Service(Amazon EKS), 및 온프레미스 서버를 사용하는 워크로드에 대해 Amazon CloudWatch 및 관련 Amazon Web Services() 관리 AWS Lambda및 거버넌스 서비스를 사용하여 로깅 및 모니터링을 설계하고 구현하는 데 도움이 됩니다.AWSAmazon EC2 https://docs.aws.amazon.com//AmazonECS/latest/developerguide/Welcome.html 이 가이드는 AWS 클라우드에서 워크로드를 관리하는 운영 팀, DevOps 엔지니어 및 애플리케이션 엔지니어를 대상으로 합니다.

로깅 및 모니터링 접근 방식은 AWS Well-Architected Framework의 6가지 원칙을 기반으로 해야 합니다. 이러한 원칙은 운영 우수성, 보안, 신뢰성, 성능 효율성 및 비용 최적화입니다. 잘 설계된 모니터링 및 경보 솔루션은 인프라를 사전에 분석하고 조정할 수 있도록 지원하여 안정성과 성능을 개선합니다.

보안 또는 비용 최적화에 대한 로깅 및 모니터링은 심층 평가가 필요한 주제이므로이 가이드에서는 이에 대해 광범위하게 설명하지 않습니다. AWS CloudTrail, , AWS Config Amazon Inspector, Amazon Detective, Amazon MacieAmazon Macie Amazon GuardDuty, 등 보안 로깅 및 모니터링을 지원하는 많은 AWS 서비스가 있습니다AWS Security Hub CSPM. 비용 최적화를 위해 AWS Cost ExplorerAWS Budgets, 및 CloudWatch 결제 지표를 사용할 수도 있습니다.

다음 표에는 로깅 및 모니터링 솔루션이 해결해야 하는 6가지 영역이 요약되어 있습니다.

로그 파일 및 지표 캡처 및 수집	다양한 소스의 서비스에서 시스템 및 애플리케이션 로그와 지표를 AWS 식별, 구성 및 전송합니다.
로그 검색 및 분석	로그를 검색하고 분석하여 운영 관리, 문제 식별, 문제 해결 및 애플리케이션 분석을 수행합니다.
지표 모니터링 및 경보	워크로드의 관찰 및 추세를 식별하고 조치를 취합니다.

애플리케이션 및 서비스 가용성 모니터링	서비스 가용성을 지속적으로 모니터링하여 가동 중지 시간을 줄이고 서비스 수준 목표를 충족하는 기능을 개선합니다.
애플리케이션 추적	시스템 및 외부 종속성에서 애플리케이션 요청을 추적하여 성능을 미세 조정하고, 근본 원인 분석을 수행하고, 문제를 해결합니다.
대시보드 및 시각화 생성	시스템 및 워크로드에 대한 관련 지표 및 관찰에 초점을 맞춘 대시보드를 생성하여 문제를 지속적으로 개선하고 사전에 발견할 수 있습니다.

CloudWatch는 대부분의 로깅 및 모니터링 요구 사항을 충족할 수 있으며 안정적이고 확장 가능하며 유연한 솔루션을 제공합니다. 모니터링 및 분석을 위한 CloudWatch 로깅 통합 외에도 많은 AWS 서비스가 CloudWatch 지표를 자동으로 제공합니다. 또한 CloudWatch는 서버(클라우드 및 온프레미스 모두), 컨테이너 및 서버리스 컴퓨팅과 같은 다양한 컴퓨팅 옵션을 지원하는 에이전트 및 로그 드라이버를 제공합니다. 이 안내서에서는 로깅 및 모니터링에 사용되는 다음 AWS 서비스도 다룹니다.

- EC2 인스턴스 및 온프레미스 서버의 CloudWatch 에이전트를 자동화, 구성 및 업데이트하기 위한 AWS Systems Manager Distributor, Systems Manager State Manager 및 Systems Manager Automation
- 고급 로그 집계, 검색 및 분석을 위한 Amazon OpenSearch Service
- 애플리케이션 및 서비스 가용성을 모니터링하기 위한 Amazon Route 53 상태 확인 및 CloudWatch Synthetics
- 컨테이너화된 애플리케이션을 대규모로 모니터링하기 위한 Amazon Managed Service for Prometheus
- AWS X-Ray 애플리케이션 추적 및 런타임 분석
- 여러 소스(예: CloudWatch, Amazon OpenSearch Service, Amazon Timestream)의 데이터를 시각화하고 분석하는 Amazon Managed Grafana

선택한 AWS 컴퓨팅 서비스도 로깅 및 모니터링 솔루션의 구현 및 구성에 영향을 미칩니다. 예를 들어 CloudWatch의 구현 및 구성은 Amazon EC2, Amazon ECS, Amazon EKS 및 Lambda에 대해 다릅니다.

애플리케이션 및 워크로드 소유자는 로깅 및 모니터링을 잊어버리거나 일관되지 않게 구성하고 구현할 수 있습니다. 즉, 워크로드는 관찰성이 제한된 프로덕션 환경에 진입하여 문제 식별이 지연되고 문

제를 해결하고 해결하는 데 걸리는 시간이 늘어납니다. 로깅 및 모니터링 솔루션은 애플리케이션 로그 및 지표에 대한 애플리케이션 계층 외에도 운영 체제(OS) 수준 로그 및 지표에 대한 시스템 계층을 해결해야 합니다. 이 가이드는 다음 표에 설명된 세 가지 컴퓨팅 유형을 포함하여 다양한 컴퓨팅 유형에서 이러한 두 계층을 해결하기 위한 권장 접근 방식을 제공합니다.

장기 실행 및 변경 불가능한 EC2 인스턴스	여러 AWS 리전 또는 계정의 여러 운영 체제(OSs)에 대한 시스템 및 애플리케이션 로그와 지표입니다.
컨테이너	다양한 구성에 대한 예제를 포함하여 Amazon ECS 및 Amazon EKS 클러스터에 대한 시스템 및 애플리케이션 로그와 지표입니다.
서버리스	Lambda 함수에 대한 시스템 및 애플리케이션 로그와 지표 및 사용자 지정 고려 사항.

이 가이드는 다음 영역에서 CloudWatch 및 관련 AWS 서비스를 다루는 로깅 및 모니터링 솔루션을 제공합니다.

- CloudWatch 배포 계획 - CloudWatch 배포 계획 시 고려 사항 및 CloudWatch 구성 중앙 집중화에 대한 지침.
- EC2 인스턴스 및 온프레미스 서버에 대한 CloudWatch 에이전트 구성 - 시스템 수준 및 애플리케이션 수준 로깅 및 지표에 대한 CloudWatch 구성 세부 정보입니다.
- Amazon EC2 및 온프레미스 서버에 대한 CloudWatch 에이전트 설치 접근 방식 - 여러 리전 및 계정에서 Systems Manager를 사용한 자동 배포를 포함하여 CloudWatch 에이전트를 설치하는 접근 방식입니다.
- Amazon ECS에서 로깅 및 모니터링 - Amazon ECS에서 클러스터 수준 및 애플리케이션 수준 로깅과 지표를 위해 CloudWatch를 구성하기 위한 지침입니다.
- Amazon EKS의 로깅 및 모니터링 - Amazon EKS의 클러스터 수준 및 애플리케이션 수준 로깅 및 지표에 대한 CloudWatch 구성 지침입니다.
- Amazon EKS에서 Prometheus 모니터링 - Amazon Managed Service for Prometheus를 Prometheus에 대한 CloudWatch Container Insights 모니터링과 소개하고 비교합니다.
- 에 대한 로깅 및 지표 AWS Lambda - Lambda 함수에 대한 CloudWatch 구성 지침입니다.

- Searching and analyzing logs in CloudWatch - Amazon CloudWatch Application Insights, CloudWatch Logs Insights를 사용하고 로그 분석을 Amazon OpenSearch Service로 확장하여 로그를 분석하는 방법입니다.
- CloudWatch를 사용한 경보 옵션 - CloudWatch 경보 및 CloudWatch 이상 탐지를 소개하고 경보 생성 및 설정에 대한 지침을 제공합니다.
- 애플리케이션 및 서비스 가용성 모니터링 - 자동 가용성 모니터링을 위한 CloudWatch Synthetics 및 Route 53 상태 확인을 도입하고 비교합니다.
- 를 사용하여 애플리케이션 추적 AWS X-Ray - Amazon EC2, Amazon ECS, Amazon EKS 및 Lambda용 X-Ray를 사용한 애플리케이션 추적 소개 및 설정
- Dashboards and visualizations with CloudWatch - AWS 워크로드 간 관찰성을 개선하기 위한 CloudWatch Dashboards 소개.
- AWS 서비스와 CloudWatch 통합 - CloudWatch가 다양한 AWS 서비스와 통합되는 방법을 설명합니다.
- 대시보드 및 시각화를 위한 Amazon Managed Grafana - 대시보드 및 시각화를 위해 Amazon Managed Grafana를 CloudWatch와 소개하고 비교합니다.

구현 예제는이 가이드 전체에서 이러한 영역에 사용되며 AWS 샘플 GitHub 리포지토리에서도 사용할 수 있습니다.

목표 비즈니스 성과

AWS 클라우드 컴퓨팅의 6가지 이점을 달성하려면 클라우드용으로 설계된 로깅 및 모니터링 솔루션을 생성하는 것이 필수적입니다. 로깅 및 모니터링 솔루션은 IT 조직이 비즈니스 프로세스, 비즈니스 파트너, 직원 및 고객에게 도움이 되는 비즈니스 성과를 달성하는 데 도움이 될 것입니다. AWS Well-Architected Framework에 부합하는 로깅 및 모니터링 솔루션을 구현한 후 다음 네 가지 결과를 기대할 수 있습니다.

운영 준비 가속화

로깅 및 모니터링 솔루션을 활성화하는 것은 프로덕션 지원 및 사용을 위해 워크로드를 준비하는 데 중요한 구성 요소입니다. 수동 프로세스에 너무 많이 의존하면 운영 준비 상태가 빠르게 병목 현상이 될 수 있으며 IT 투자의 가치 실현 시간(TTV)을 줄일 수도 있습니다. 또한 비효율적인 접근 방식은 워크로드의 관찰성을 제한합니다. 이로 인해 장기 중단, 고객 불만족 및 비즈니스 프로세스 실패의 위험이 증가할 수 있습니다.

이 가이드의 접근 방식을 사용하여 AWS 클라우드에서 로깅 및 모니터링을 표준화하고 자동화할 수 있습니다. 그러면 새 워크로드는 프로덕션 로깅 및 모니터링을 위한 수동 준비 및 개입을 최소화해야 합니다. 또한 이를 통해 여러 계정 및 리전의 다양한 워크로드에 대해 대규모 로깅 및 모니터링 표준을 생성하는 데 필요한 시간과 단계를 줄일 수 있습니다.

운영 우수성 개선

이 가이드에서는 다양한 워크로드가 비즈니스 목표 및 운영 우수성을 충족하는 데 도움이 되는 로깅 및 모니터링에 대한 여러 모범 사례를 제공합니다. 또한이 가이드에서는 AWS 서비스를 사용하여 잘 설계된 로깅 및 모니터링 솔루션을 구현하기 위해 코드형 인프라(IaC) 접근 방식과 함께 사용할 수 있는 자세한 예제와 재사용 가능한 오픈 소스 템플릿도 제공합니다. 운영 우수성 개선은 반복적이며 지속적인 개선이 필요합니다. 이 가이드에서는 로깅 및 모니터링 관행을 지속적으로 개선하는 방법에 대한 제안을 제공합니다.

운영 가시성 향상

비즈니스 프로세스와 애플리케이션은 다양한 IT 리소스에서 지원될 수 있으며 온프레미스 또는 AWS 클라우드의 다양한 컴퓨팅 유형에서 호스팅될 수 있습니다. 로깅 및 모니터링 전략의 일관되지 않고 불완전한 구현으로 인해 운영 가시성이 제한될 수 있습니다. 포괄적인 로깅 및 모니터링 접근 방식을 채택하면 워크로드 전반의 문제를 신속하게 식별, 진단 및 대응할 수 있습니다. 이 가이드는 완전한 운영

가시성을 개선하고 평균 해결 시간(MTTR) 장애를 줄이기 위한 접근 방식을 설계하고 구현하는 데 도움이 됩니다. 또한 포괄적인 로깅 및 모니터링 접근 방식은 조직이 서비스 품질을 개선하고, 최종 사용자 경험을 개선하고, 서비스 수준 계약(SLAs)을 충족하는 데 도움이 됩니다.

운영 규모 조정 및 오버헤드 비용 절감

이 가이드의 로깅 및 모니터링 사례를 확장하여 여러 리전 및 계정, 수명이 짧은 리소스 및 여러 환경을 지원할 수 있습니다. 이 가이드에서는 수동 단계(예: 에이전트 설치 및 구성, 지표 모니터링, 문제 발생 시 알림 또는 조치 수행)를 자동화하는 접근 방식과 예제를 제공합니다. 이러한 접근 방식은 클라우드 채택이 성숙해지고 성장하며 클라우드 관리 활동이나 리소스를 늘리지 않고 운영 기능을 확장해야 할 때 유용합니다.

CloudWatch 배포 계획

로깅 및 모니터링 솔루션의 복잡성과 범위는 다음과 같은 여러 요인에 따라 달라집니다.

- 사용되는 환경, 리전 및 계정 수와이 수가 증가할 수 있는 방식.
- 기존 워크로드 및 아키텍처의 다양성과 유형입니다.
- 로깅하고 모니터링해야 하는 컴퓨팅 유형 및 OSs.
- 온프레미스 위치와 AWS 인프라가 모두 있는지 여부.
- 여러 시스템 및 애플리케이션의 집계 및 분석 요구 사항입니다.
- 로그 및 지표의 무단 노출을 방지하는 보안 요구 사항입니다.
- 운영 프로세스를 지원하기 위해 로깅 및 모니터링 솔루션과 통합해야 하는 제품 및 솔루션.

로깅 및 모니터링 솔루션을 정기적으로 검토하고 새 워크로드 배포 또는 업데이트된 워크로드 배포로 업데이트해야 합니다. 로깅, 모니터링 및 경보에 대한 업데이트는 문제가 관찰될 때 식별하고 적용해야 합니다. 그런 다음 이러한 문제를 사전에 식별하고 예방할 수 있습니다.

로그 및 지표를 캡처하고 수집하기 위한 소프트웨어 및 서비스를 지속적으로 설치하고 구성해야 합니다. 설정된 로깅 및 모니터링 접근 방식은 다양한 도메인(예: 보안, 성능, 네트워킹 또는 분석)에 대해 여러 AWS 또는 독립 소프트웨어 공급업체(ISV) 서비스 및 솔루션을 사용합니다. 각 도메인에는 고유한 배포 및 구성 요구 사항이 있습니다.

CloudWatch를 사용하여 여러 OSs 및 컴퓨팅 유형에 대한 로그와 지표를 캡처하고 수집하는 것이 좋습니다. 많은 AWS 서비스에서 CloudWatch를 사용하여 추가 구성 없이 로그 및 지표를 로깅, 모니터링 및 게시합니다. CloudWatch는 다양한 OS 및 환경에 설치 및 구성할 수 있는 소프트웨어 에이전트를 제공합니다. OSs 다음 섹션에서는 여러 계정, 리전 및 구성에 대해 CloudWatch 에이전트를 배포, 설치 및 구성하는 방법을 간략하게 설명합니다.

주제

- 중앙 집중식 또는 분산 계정에서 CloudWatch 사용
- CloudWatch 에이전트 구성 파일 관리

중앙 집중식 또는 분산 계정에서 CloudWatch 사용

CloudWatch는 한 계정 및 리전의 AWS 서비스 또는 리소스를 모니터링하도록 설계되었지만 중앙 계정을 사용하여 여러 계정 및 리전의 로그와 지표를 캡처할 수 있습니다. 둘 이상의 계정 또는 리전을 사

용하는 경우 중앙 집중식 계정 접근 방식을 사용할지 아니면 개별 계정을 사용하여 로그 및 지표를 캡처할지 평가해야 합니다. 일반적으로 보안, 분석, 운영 및 워크로드 소유자의 요구 사항을 지원하기 위해 다중 계정 및 다중 리전 배포에는 하이브리드 접근 방식이 필요합니다.

다음 표에는 중앙 집중식, 분산형 또는 하이브리드 접근 방식을 사용하기로 선택할 때 고려해야 할 영역이 나와 있습니다.

계정 구조	조직에 여러 개의 개별 계정(예: 비프로덕션 및 프로덕션 워크로드에 대한 계정) 또는 특정 환경의 단일 애플리케이션에 대한 수천 개의 계정이 있을 수 있습니다. 워크로드가 실행되는 계정에 애플리케이션 로그 및 지표를 유지하여 워크로드 소유자에게 로그 및 지표에 대한 액세스 권한을 부여하는 것이 좋습니다. 이를 통해 로깅 및 모니터링에서 활성 역할을 가질 수 있습니다. 또한 별도의 로깅 계정을 사용하여 분석, 집계, 추세 및 중앙 집중식 작업을 위한 모든 워크로드 로그를 집계하는 것이 좋습니다. 보안, 아카이빙 및 모니터링, 분석에 별도의 로깅 계정을 사용할 수도 있습니다.
액세스 요구 사항	팀원(예: 워크로드 소유자 또는 개발자)은 문제를 해결하고 개선하기 위해 로그 및 지표에 액세스해야 합니다. 액세스 및 문제 해결을 더 쉽게 하려면 워크로드 계정에서 로그를 유지 관리해야 합니다. 로그 및 지표가 워크로드와 별도의 계정에 유지 관리되는 경우 사용자는 정기적으로 계정을 바꿔야 할 수 있습니다. 중앙 집중식 계정을 사용하면 워크로드 계정에 대한 액세스 권한을 부여하지 않고 권한 있는 사용자에게 로그 정보를 제공합니다. 이를 통해 여러 계정에서 실행되는 워크로드에서 집계가 필요한 분석 워크로드의 액세스 요구 사항을 간소화할 수 있습니다. 중앙 집중식 로깅 계정에는 Amazon OpenSearch Service 클러스터와 같은 대체 검색 및 집계 옵션도 있을 수 있습니다. Amazon OpenSearch Service는 로그에 대한 필드 수준까지 세분화된 액세스 제어를 제공합니다. 특수한 액세스 및 권한이 필요한 민감한 데이터나 기밀 데이터가 있는 경우 세분화된 액세스 제어가 중요합니다.
운영	많은 조직에 중앙 집중식 운영 및 보안 팀 또는 모니터링을 위해 로그에 액세스해야 하는 운영 지원을 위한 외부 조직이 있습니다. 중앙 집중식 로깅 및 모니터링을 사용하면 모든 계정 및 워크로드에서 추세를

	쉽게 식별하고, 검색하고, 집계하고, 분석을 수행할 수 있습니다. 조직에서 DevOps에 대해 '구축하여 실행' 접근 방식을 사용하는 경우 워크로드 소유자는 계정에 로깅 및 모니터링 정보가 필요합니다. 분산 워크로드 소유권 외에도 중앙 운영 및 분석을 충족하기 위해 하이브리드 접근 방식이 필요할 수 있습니다.
환경	프로덕션 계정의 중앙 위치에 로그 및 지표를 호스팅하고 보안 요구 사항 및 계정 아키텍처에 따라 다른 환경(예: 개발 또는 테스트)에 대한 로그 및 지표를 동일하거나 별도의 계정에 보관하도록 선택할 수 있습니다. 이렇게 하면 프로덕션 중에 생성된 민감한 데이터에 더 많은 사용자가 액세스하는 것을 방지할 수 있습니다.

CloudWatch는 CloudWatch 구독 필터를 사용하여 로그를 실시간으로 처리할 수 있는 여러 옵션을 제공합니다. 구독 필터를 사용하여 사용자 지정 처리, 분석 및 다른 시스템으로 로드하기 위해 AWS 서비스에 실시간으로 로그를 스트리밍할 수 있습니다. 이는 중앙 집중식 계정 및 리전 외에도 개별 계정 및 리전에서 로그 및 지표를 사용할 수 있는 하이브리드 접근 방식을 취하는 경우 특히 유용할 수 있습니다. 다음 목록은 이를 위해 사용할 수 있는 AWS 서비스의 예를 제공합니다.

- Amazon Data Firehose – Firehose는 생성되는 데이터 볼륨에 따라 자동으로 규모를 조정하고 크기를 조정하는 스트리밍 솔루션을 제공합니다. Amazon Kinesis 데이터 스트림의 샤드 수를 관리할 필요가 없으며 추가 코딩 없이 Amazon Simple Storage Service(Amazon S3), Amazon OpenSearch Service 또는 Amazon Redshift에 직접 연결할 수 있습니다. Firehose는 해당 AWS 서비스의 로그를 중앙 집중화하려는 경우 효과적인 솔루션입니다.
- Amazon Kinesis Data Streams – Kinesis Data Streams는 Firehose가 지원하지 않는 서비스와 통합하고 추가 처리 로직을 구현해야 하는 경우 적절한 솔루션입니다. 중앙 계정의 Kinesis 데이터 스트림과 스트림에 레코드를 배치할 수 있는 권한을 부여하는 AWS Identity and Access Management (IAM) 역할을 지정하는 Amazon CloudWatch Logs 대상을 계정 및 리전에 생성할 수 있습니다. Kinesis Data Streams는 다양한 옵션에서 사용할 수 있는 로그 데이터를 위한 유연한 개방형 랜딩 존을 제공합니다. Kinesis Data Streams 로그 데이터를 계정으로 읽고, 사전 처리를 수행하고, 선택한 대상으로 데이터를 전송할 수 있습니다.

  그러나 스트림의 샤드를 구성하여 생성된 로그 데이터에 맞게 크기를 적절하게 조정해야 합니다. Kinesis Data Streams는 로그 데이터에 대한 임시 중개자 또는 대기열 역할을 하며 1~365일 동안 Kinesis 스트림 내에 데이터를 저장할 수 있습니다. Kinesis Data Streams는 재생 기능도 지원하므로 사용하지 않은 데이터를 재생할 수 있습니다.

- Amazon OpenSearch Service – CloudWatch Logs는 로그 그룹의 로그를 개별 또는 중앙 집중식 계정의 OpenSearch 클러스터로 스트리밍할 수 있습니다. OpenSearch 클러스터로 데이터를 스트리밍하도록 로그 그룹을 구성하면 로그 그룹과 동일한 계정 및 리전에 Lambda 함수가 생성됩니다. Lambda 함수는 OpenSearch 클러스터와 네트워크 연결이 있어야 합니다. Amazon OpenSearch Service로 수집을 사용자 지정하는 것 외에도 추가 사전 처리를 수행하도록 Lambda 함수를 사용자 지정할 수 있습니다. Amazon OpenSearch Service를 사용한 중앙 집중식 로깅을 사용하면 클라우드 아키텍처의 여러 구성 요소에서 문제를 더 쉽게 분석, 검색 및 해결할 수 있습니다.
- Lambda - Kinesis Data Streams를 사용하는 경우 스트림의 데이터를 소비하는 컴퓨팅 리소스를 프로비저닝하고 관리해야 합니다. 이를 방지하기 위해 로그 데이터를 처리를 위해 Lambda로 직접 스트리밍하고 로직에 따라 대상으로 전송할 수 있습니다. 즉, 수신 데이터를 처리하기 위해 컴퓨팅 리소스를 프로비저닝하고 관리할 필요가 없습니다. Lambda를 사용하기로 선택한 경우 솔루션이 Lambda 할당량과 호환되는지 확인합니다.

CloudWatch Logs에 저장된 로그 데이터를 파일 형식으로 처리하거나 공유해야 할 수 있습니다. 내보내기 작업을 생성하여 특정 날짜 또는 시간 범위에 대해 로그 그룹을 Amazon S3로 내보낼 수 있습니다. 예를 들어 분석 및 감사를 위해 매일 Amazon S3로 로그를 내보내도록 선택할 수 있습니다. Lambda를 사용하여이 솔루션을 자동화할 수 있습니다. 또한이 솔루션을 Amazon S3 복제와 결합하여 여러 계정 및 리전의 로그를 하나의 중앙 집중식 계정 및 리전으로 전송하고 중앙 집중화할 수 있습니다.

CloudWatch 에이전트 구성은 `agent` 섹션에서 `credentials` 필드를 지정할 수도 있습니다. 이는 지표와 로그를 다른 계정으로 전송할 때 사용할 IAM 역할을 지정합니다. 지정된 경우이 필드에 `role_arn` 파라미터가 포함됩니다. 이 필드는 특정 중앙 집중식 계정 및 리전에서 중앙 집중식 로깅 및 모니터링만 필요한 경우에 사용할 수 있습니다.

AWS SDK를 사용하여 원하는 언어로 사용자 지정 처리 애플리케이션을 작성하고, 계정에서 로그 및 지표를 읽고, 추가 처리 및 모니터링을 위해 중앙 집중식 계정 또는 기타 대상으로 데이터를 전송할 수도 있습니다.

CloudWatch 에이전트 구성 파일 관리

모든 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 및 온프레미스 서버에서 캡처하려는 시스템 로그 및 지표가 포함된 표준 Amazon CloudWatch 에이전트 구성을 생성하는 것이 좋습니다.Amazon EC2 CloudWatch 에이전트 구성 파일 마법사를 사용하여 구성 파일을 생성할 수 있습니다. 구성 마법사를 여러 번 실행하여 다양한 시스템 및 환경에 대한 고유한 구성을 생성할 수 있습니다. 구성 파일 스키마를 사용하여 구성 파일을 수정하거나 변형을 생성할 수도 있습니다. CloudWatch 에이전트 구성 파일은 AWS Systems Manager 파라미터 스토어 파라미터에 저장할 수 있습니다.

CloudWatch 에이전트 구성 파일이 여러 개 있는 경우 별도의 파라미터 스토어 파라미터를 생성할 수 있습니다. 여러 AWS 계정 또는 AWS 리전을 사용하는 경우 각 계정 및 리전에서 파라미터 스토어 파라미터를 관리하고 업데이트해야 합니다. 또는 CloudWatch 구성을 Amazon S3의 파일 또는 원하는 버전 제어 도구로 중앙에서 관리할 수 있습니다.

CloudWatch 에이전트에 포함된 `amazon-cloudwatch-agent-ctl` 스크립트를 사용하면 구성 파일, 파라미터 스토어 파라미터 또는 에이전트의 기본 구성을 지정할 수 있습니다. 기본 구성은 사전 정의된 기본 지표 세트에 맞게 조정되고 메모리 및 디스크 공간 지표를 CloudWatch에 보고하도록 에이전트를 구성합니다. 그러나 로그 파일 구성은 포함되지 않습니다. CloudWatch 에이전트에 Systems Manager 빠른 설정을 사용하는 경우에도 기본 구성이 적용됩니다.

기본 구성에는 로깅이 포함되지 않고 요구 사항에 맞게 사용자 지정되지 않으므로 요구 사항에 맞게 사용자 지정된 자체 CloudWatch 구성을 생성하고 적용하는 것이 좋습니다.

CloudWatch 구성 관리

기본적으로 CloudWatch 구성은 파라미터 스토어 파라미터 또는 CloudWatch 구성 파일로 저장 및 적용할 수 있습니다. 최선의 선택은 요구 사항에 따라 달라집니다. 이 섹션에서는이 두 옵션의 장단점에 대해 설명합니다. 여러 AWS 계정 및 AWS 리전에 대한 CloudWatch 구성 파일을 관리하기 위한 대표적인 솔루션도 자세히 설명되어 있습니다.

Systems Manager 파라미터 스토어 파라미터

작은 AWS 계정 및 리전 집합에서 적용하고 관리하려는 표준 CloudWatch 에이전트 구성 파일이 하나 있는 경우 파라미터 스토어 파라미터를 사용하여 CloudWatch 구성을 관리하는 것이 좋습니다. CloudWatch CloudWatch 구성을 파라미터 스토어 파라미터로 저장할 때 CloudWatch 에이전트 구성 도구(`amazon-cloudwatch-agent-ctl`Linux 기반)를 사용하여 구성 파일을 인스턴스에 복사할 필요 없이 파라미터 스토어에서 구성을 읽고 적용할 수 있습니다. AmazonCloudWatch-ManageAgent Systems Manager Command 문서를 사용하여 한 번의 실행으로 여러 EC2 인스턴스에서 CloudWatch 구성을 업데이트할 수 있습니다. 파라미터 스토어 파라미터는 리전 파라미터이므로 각 AWS 리전 및 AWS 계정에서 CloudWatch 파라미터 스토어 파라미터를 업데이트하고 유지 관리해야 합니다. 각 인스턴스에 적용하려는 CloudWatch 구성이 여러 개 있는 경우 이러한 파라미터를 포함하도록 AmazonCloudWatch-ManageAgent Command 문서를 사용자 지정해야 합니다.

CloudWatch 구성 파일

AWS 계정과 리전이 많고 여러 CloudWatch 구성 파일을 관리하는 경우 CloudWatch 구성을 파일로 관리하는 것이 효과적일 수 있습니다. 이 접근 방식을 사용하면 폴더 구조에서 탐색, 구성 및 관리할 수 있습니다. 개별 폴더 또는 파일에 보안 규칙을 적용하여 업데이트 및 읽기 권한과 같은 액세스를 제한하고 부여할 수 있습니다. 공동 작업을 위해 AWS 외부에서 공유 및 전송할 수 있습니다. 파일을

버전 관리하여 변경 사항을 추적하고 관리할 수 있습니다. 각 CloudWatch 구성 파일을 개별적으로 적용하지 않고 구성 파일을 CloudWatch 에이전트 구성 디렉터리에 복사하여 CloudWatch 구성을 일괄 적용할 수 있습니다. Linux의 경우 CloudWatch 구성 디렉터리는에 있습니다`/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d`. Windows의 경우 구성 디렉터리는에 있습니다`C:\ProgramData\Amazon\AmazonCloudWatchAgent\Configs`.

CloudWatch 에이전트를 시작하면 에이전트는 이러한 디렉터리에 있는 각 파일을 자동으로 추가하여 CloudWatch 복합 구성 파일을 생성합니다. 구성 파일은 필요한 계정 및 리전에서 액세스할 수 있는 중앙 위치(예: S3 버킷)에 저장해야 합니다. 이 접근 방식을 사용하는 예제 솔루션이 제공됩니다.

CloudWatch 구성 구성

CloudWatch 구성을 관리하는 데 사용되는 접근 방식에 관계없이 CloudWatch 구성을 구성합니다. 다음과 같은 접근 방식을 사용하여 구성을 파일 또는 파라미터 스토어 경로로 구성할 수 있습니다.

/config/standard/windows/ec2	Amazon EC2에 대한 표준 Windows별 CloudWatch 구성 파일을 저장합니다. 이 폴더 아래의 다양한 Windows 버전, EC2 인스턴스 유형 및 환경에 대한 표준 운영 체제(OS) 구성을 추가로 분류할 수 있습니다.
/config/standard/windows/onpremises	온프레미스 서버에 대한 표준 Windows별 CloudWatch 구성 파일을 저장합니다. 또한이 폴더에서 다양한 Windows 버전, 서버 유형 및 환경에 대한 표준 OS 구성을 추가로 분류할 수 있습니다.
/config/standard/linux/ec2	Amazon EC2에 대한 표준 Linux별 CloudWatch 구성 파일을 저장합니다. 이 폴더 아래의 다양한 Linux 배포, EC2 인스턴스 유형 및 환경에 대한 표준 OS 구성을 추가로 분류할 수 있습니다.
/config/standard/linux/온프레미스	온프레미스 서버에 대한 표준 Linux별 CloudWatch 구성 파일을 저장합니다. 이 폴더에서 다양한 Linux 배포, 서버 유형 및 환경에 대한 표준 OS 구성을 추가로 분류할 수 있습니다.
/config/ecs	Amazon ECS 컨테이너 인스턴스를 사용하는 경우 Amazon Elastic Container Service(Amazon

	ECS)와 관련된 CloudWatch 구성 파일을 저장합니다. 이러한 구성은 Amazon ECS 특정 시스템 수준 로깅 및 모니터링을 위한 표준 Amazon EC2 구성에 추가할 수 있습니다.
/config/<application_name>	애플리케이션별 CloudWatch 구성 파일을 저장합니다. 환경 및 버전에 대한 추가 폴더 및 접두사를 사용하여 애플리케이션을 추가로 분류할 수 있습니다.

예: S3 버킷에 CloudWatch 구성 파일 저장

이 섹션에서는 Amazon S3를 사용하여 CloudWatch 구성 파일을 저장하는 예제와 CloudWatch 구성 파일을 검색하고 적용하는 사용자 지정 Systems Manager 실행서를 제공합니다. 이 접근 방식은 CloudWatch 구성에 Systems Manager Parameter Store 파라미터를 대규모로 사용할 때 발생하는 몇 가지 문제를 해결할 수 있습니다.

- 여러 리전을 사용하는 경우 각 리전의 파라미터 스토어에서 CloudWatch 구성 업데이트를 동기화해야 합니다. Parameter Store는 리전 서비스이며 CloudWatch 에이전트를 사용하는 각 리전에서 동일한 파라미터를 업데이트해야 합니다.
- CloudWatch 구성이 여러 개 있는 경우 각 Parameter Store 구성의 검색 및 적용을 시작해야 합니다. 파라미터 스토어에서 각 CloudWatch 구성을 개별적으로 검색하고 새 구성을 추가할 때마다 검색 방법을 업데이트해야 합니다. 반면 CloudWatch는 구성 파일을 저장하기 위한 구성 디렉터리를 제공하며, 구성 파일을 개별적으로 지정할 필요 없이 디렉터리에 각 구성을 적용합니다.
- 여러 계정을 사용하는 경우 각 새 계정에 파라미터 스토어에 필요한 CloudWatch 구성이 있는지 확인해야 합니다. 또한 향후 이러한 계정 및 해당 리전에 구성 변경 사항이 적용되도록 해야 합니다.

CloudWatch 구성을 모든 계정 및 리전에서 액세스할 수 있는 S3 버킷에 저장할 수 있습니다. 그런 다음 Systems Manager Automation 실행서 및 Systems Manager 상태 관리자를 사용하여 S3 버킷에서 CloudWatch 구성 디렉터리로 이러한 구성을 복사할 수 있습니다. cloudwatch-config-s3-bucket.yaml AWS CloudFormation 템플릿을 사용하여 AWS Organizations의 조직 내 여러 계정에서 액세스할 수 있는 S3 버킷을 생성할 수 있습니다. AWS Organizations 템플릿에는 조직 내 모든 계정에 대한 읽기 액세스 권한을 부여하는 파라미터가 포함되어 `OrganizationID` 있습니다.

이 가이드의 상태 관리자 및 Distributor for CloudWatch 에이전트 배포 및 구성 설정 섹션에 제공된 증강 샘플 Systems Manager 실행서는 cloudwatch-config-s3-bucket.yaml AWS CloudFormation 템플릿으로 생성된 S3 버킷을 사용하여 파일을 검색하도록 구성됩니다.

또는 버전 관리 시스템(예: GitHub)을 사용하여 구성 파일을 저장할 수 있습니다. 버전 관리 시스템에 저장된 구성 파일을 자동으로 검색하려면 자격 증명 스토리지를 관리 또는 중앙 집중화하고 계정 및 전체에서 자격 증명을 검색하는 데 사용되는 Systems Manager Automation 런북을 업데이트해야 합니다 AWS 리전.

EC2 인스턴스 및 온프레미스 서버에 대한 CloudWatch 에이전트 구성

많은 조직이 물리적 서버와 가상 머신(VMs) 모두에서 워크로드를 실행합니다. 이러한 워크로드는 일반적으로 지표를 캡처하고 수집하기 위한 고유한 설치 및 구성 요구 사항이 있는 서로 다른 OSs에서 실행됩니다.

EC2 인스턴스를 사용하기로 선택하면 인스턴스 및 OS 구성을 높은 수준으로 제어할 수 있습니다. 그러나 이러한 높은 수준의 제어 및 책임을 위해서는 구성을 모니터링하고 조정하여 보다 효율적으로 사용해야 합니다. 로깅 및 모니터링 표준을 설정하고 로그 및 지표를 캡처하고 수집하기 위한 표준 설치 및 구성 접근 방식을 적용하여 운영 효율성을 개선할 수 있습니다.

IT 투자를 AWS 클라우드로 마이그레이션하거나 확장하는 조직은 CloudWatch를 활용하여 통합 로깅 및 모니터링 솔루션을 달성할 수 있습니다. CloudWatch 요금은 캡처하려는 지표 및 로그에 대해 점진적으로 비용을 지불한다는 의미입니다. Amazon EC2와 유사한 CloudWatch 에이전트 설치 프로세스를 사용하여 온프레미스 서버의 로그 및 지표를 캡처할 수도 있습니다.

CloudWatch 설치 및 배포를 시작하기 전에 시스템 및 애플리케이션의 로깅 및 지표 구성을 평가해야 합니다. 사용하려는 OSs에 대해 캡처해야 하는 표준 로그 및 지표를 정의해야 합니다. 시스템 로그와 지표는 OS에서 생성되며 Linux와 Windows에서는 다르기 때문에 로깅 및 모니터링 솔루션의 기반과 표준입니다. Linux 버전 또는 배포와 관련된 지표 및 로그 파일 외에도 Linux 배포에서 사용할 수 있는 중요한 지표 및 로그 파일이 있습니다. 이 분산은 여러 Windows 버전 간에도 발생합니다.

CloudWatch 에이전트 구성

CloudWatch는 각 OS에 고유한 CloudWatch 에이전트 및 에이전트 구성 파일을 사용하여 Amazon EC2 및 온프레미스 서버에 대한 지표와 로그를 캡처합니다. CloudWatch 계정에 CloudWatch 에이전트를 대규모로 설치하기 전에 조직의 표준 지표 및 로그 캡처 구성을 정의하는 것이 좋습니다.

여러 CloudWatch 에이전트 구성을 결합하여 복합 CloudWatch 에이전트 구성을 구성할 수 있습니다. 시스템 및 애플리케이션 수준에서 로그 및 지표에 대한 구성을 정의하고 나누는 것이 좋습니다. 다음 다이어그램은 다양한 요구 사항에 대한 여러 CloudWatch 구성 파일 유형을 결합하여 복합 CloudWatch 구성을 구성하는 방법을 보여줍니다.

이러한 로그 및 지표는 특정 환경 또는 요구 사항에 맞게 추가로 분류하고 구성할 수도 있습니다. 예를 들어, 규제되지 않은 개발 환경에서는 정밀도가 낮은 더 작은 로그 및 지표 하위 집합을 정의하고, 규제되는 프로덕션 환경에서는 정밀도가 높은 더 크고 완전한 세트를 정의할 수 있습니다.

EC2 인스턴스에 대한 로그 캡처 구성

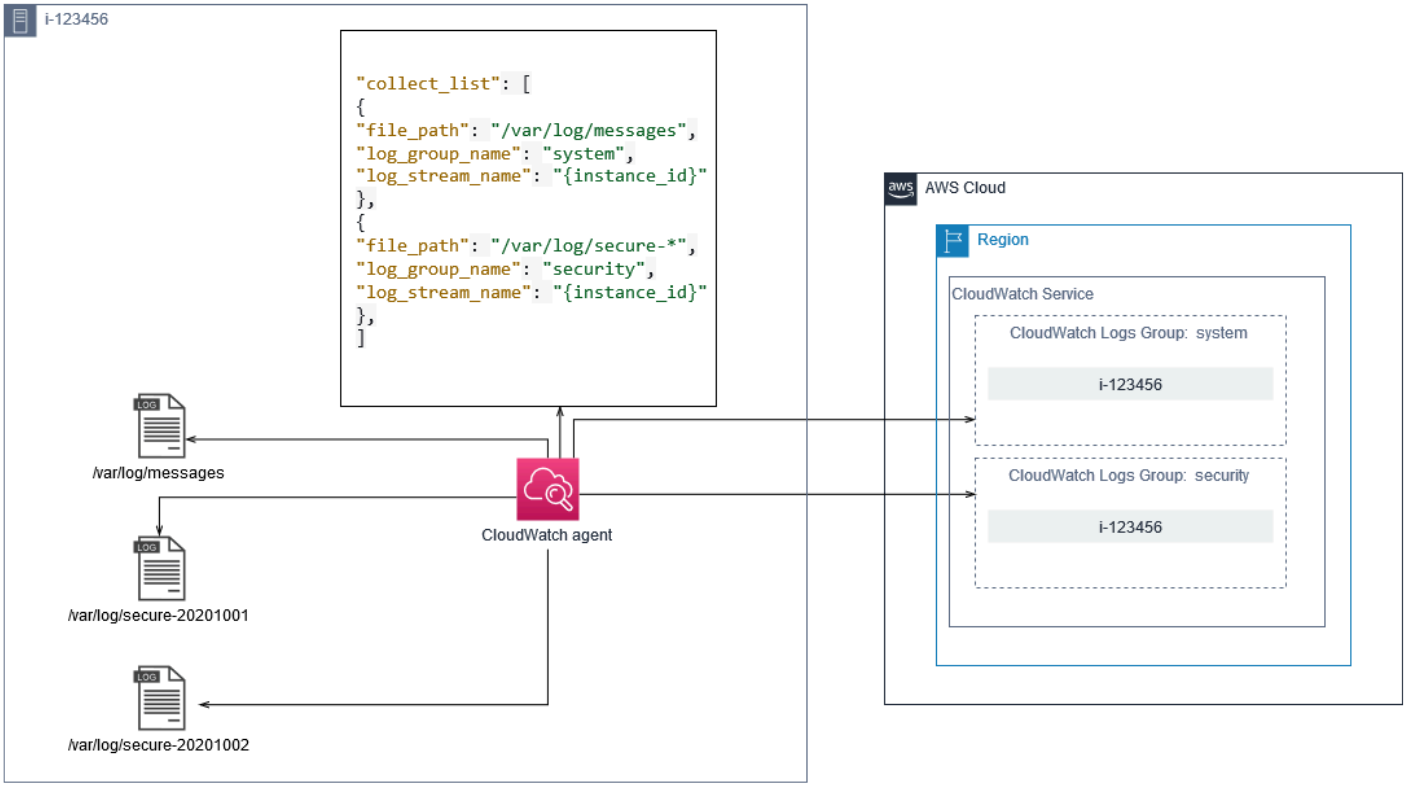
기본적으로 Amazon EC2는 로그 파일을 모니터링하거나 캡처하지 않습니다. 대신 EC2 인스턴스, AWS API 또는 AWS Command Line Interface ()에 설치된 CloudWatch 에이전트 소프트웨어가 로그 파일을 캡처하여 CloudWatch Logs에 수집합니다AWS CLI. CloudWatch 에이전트를 사용하여 Amazon EC2 및 온프레미스 서버용 CloudWatch Logs에 로그 파일을 수집하는 것이 좋습니다.

로그를 검색 및 필터링하고 CloudWatch의 로그 파일에서 패턴 패치를 기반으로 지표를 추출하고 자동화를 실행할 수 있습니다. CloudWatch는 JSON 형식 로그를 통해 가장 유연한 일반 텍스트, 공백으로 구분된 JSON 형식의 필터 및 패턴 구문 옵션을 지원합니다. 필터링 및 분석 옵션을 늘리려면 일반 텍스트 대신 형식이 지정된 로그 출력을 사용해야 합니다.

CloudWatch 에이전트는 CloudWatch로 전송할 로그 및 지표를 정의하는 구성 파일을 사용합니다. 그런 다음 CloudWatch는 각 로그 파일을 로그 스트림으로 캡처하고 이러한 로그 스트림을 로그 그룹으로 그룹화합니다. 이렇게 하면 일치하는 문자열을 검색하는 등 EC2 인스턴스의 로그에서 작업을 수행할 수 있습니다.

기본 로그 스트림 이름은 EC2 인스턴스 ID와 동일하고 기본 로그 그룹 이름은 로그 파일 경로와 동일합니다. 로그 스트림의 이름은 CloudWatch 로그 그룹 내에서 고유해야 합니다. 로그 스트림 및 로그 그룹 이름에서 동적 대체ip_address에 `instance_id`, `local_hostname`, 또는 `hostname`를 사용할 수 있습니다. 즉, 여러 EC2 인스턴스에서 동일한 CloudWatch 에이전트 구성 파일을 사용할 수 있습니다.

다음 다이어그램은 로그 캡처를 위한 CloudWatch 에이전트 구성을 보여줍니다. 로그 그룹은 캡처된 로그 파일에 의해 정의되며 `{instance_id}` 변수가 로그 스트림 이름에 사용되고 EC2 인스턴스 ID가 고유하기 때문에 각 EC2 인스턴스에 대해 별도의 로그 스트림을 포함합니다. IDs



로그 그룹은 포함된 로그 스트림의 보존, 태그, 보안, 지표 필터 및 검색 범위를 정의합니다. 로그 파일 이름을 기반으로 하는 기본 그룹화 동작은 계정 및 리전의 EC2 인스턴스에서 로그 파일과 관련된 데이터를 검색, 지표 생성 및 경보하는 데 도움이 됩니다. 추가 로그 그룹 개선이 필요한지 여부를 평가해야 합니다. 예를 들어 계정이 여러 사업부에서 공유되고 기술 또는 운영 소유자가 다를 수 있습니다. 즉, 분리 및 소유권을 반영하도록 로그 그룹 이름을 추가로 구체화해야 합니다. 이 접근 방식을 사용하면 관련 EC2 인스턴스에 분석 및 문제 해결에 집중할 수 있습니다.

여러 환경에서 하나의 계정을 사용하는 경우 각 환경에서 실행되는 워크로드에 대한 로깅을 분리할 수 있습니다. 다음 표에는 사업부, 프로젝트 또는 애플리케이션 및 환경을 포함하는 로그 그룹 이름 지정 규칙이 나와 있습니다.

로그 그룹 이름	`/<Business unit>/<Project or application name>/<Environment>/<Log file name>`
로그 스트림 이름	`<EC2 instance ID>`

EC2 인스턴스의 모든 로그 파일을 동일한 로그 그룹으로 그룹화할 수도 있습니다. 이렇게 하면 단일 EC2 인스턴스에 대한 로그 파일 집합을 더 쉽게 검색하고 분석할 수 있습니다. 이는 대부분의 EC2 인스턴스가 하나의 애플리케이션 또는 워크로드를 서비스하고 각 EC2 인스턴스가 특정 목적을 수행하는 경우에 유용합니다. 다음 표는이 접근 방식을 지원하도록 로그 그룹 및 로그 스트림 이름 지정의 형식을 지정하는 방법을 보여줍니다.

로그 그룹 이름	`/<Business unit>/<Project or application name>/<Environment>/<EC2 instance ID>`
로그 스트림 이름	`<Log file name>`

EC2 인스턴스에 대한 지표 캡처 구성

기본적으로 EC2 인스턴스는 기본 모니터링을 위해 활성화되며 표준 지표 세트(예: CPU, 네트워크 또는 스토리지 관련 지표)는 5분마다 CloudWatch로 자동 전송됩니다. CloudWatch 지표는 인스턴스 패밀리에 따라 다를 수 있습니다. 예를 들어 버스트 가능한 성능 인스턴스에는 CPU 크레딧에 대한 지표가 있습니다. Amazon EC2 표준 지표는 인스턴스 가격에 포함됩니다. EC2 인스턴스에 대한 세부 모니터링을 활성화하면 1분 간격으로 데이터를 수신할 수 있습니다. 기간 빈도는 CloudWatch 비용에 영향을 미치므로 모든 EC2 인스턴스에 대해 세부 모니터링이 필요한지 아니면 일부 EC2 인스턴스에만 필요한지 평가해야 합니다. 예를 들어 프로덕션 워크로드에 대한 세부 모니터링을 활성화하고 비프로덕션 워크로드에 대한 기본 모니터링을 사용할 수 있습니다.

온프레미스 서버에는 CloudWatch에 대한 기본 지표가 포함되어 있지 않으므로 CloudWatch 에이전트, AWS CLI또는 AWS SDK를 사용하여 지표를 캡처해야 합니다. 즉, CloudWatch 구성 파일에서 캡처하려는 지표(예: CPU 사용률)를 정의해야 합니다. 온프레미스 서버에 대한 표준 EC2 인스턴스 지표가 포함된 고유한 CloudWatch 구성 파일을 생성하고 표준 CloudWatch 구성 외에도 적용할 수 있습니다.

CloudWatch의 지표는 지표 이름과 0개 이상의 차원으로 고유하게 정의되며 지표 네임스페이스에서 고유하게 그룹화됩니다. AWS 서비스에서 제공하는 지표에는 로 시작하는 네임스페이스AWS(예: `AWS/EC2`)가 있으며,AWS 지표가 아닌 지표는 사용자 지정 지표로 간주됩니다. CloudWatch 에이전트를 사용하여 구성하고 캡처하는 지표는 모두 사용자 지정 지표로 간주됩니다. 생성된 지표 수는 CloudWatch 비용에 영향을 미치므로 EC2 인스턴스 전체 또는 일부에만 각 지표가 필요한지 평가해야 합니다. 예를 들어 프로덕션 워크로드에 대한 전체 지표 세트를 정의할 수 있지만 비프로덕션 워크로드에는 이러한 지표 중 더 작은 하위 집합을 사용할 수 있습니다.

CWAgent는 CloudWatch 에이전트가 게시한 지표의 기본 네임스페이스입니다. 로그 그룹과 마찬가지로 지표 네임스페이스는 지표 세트를 구성하여 한 곳에서 함께 찾을 수 있도록 합니다. 사업부, 프로젝트 또는 애플리케이션, 환경(예: /<Business unit>/<Project or application name>/<Environment>)을 반영하도록 네임스페이스를 수정해야 합니다. 이 접근 방식은 관련이 없는 여러 워크로드가 동일한 계정을 사용하는 경우에 유용합니다. 또한 네임스페이스 이름 지정 규칙을 CloudWatch 로그 그룹 이름 지정 규칙과 연관시킬 수 있습니다.

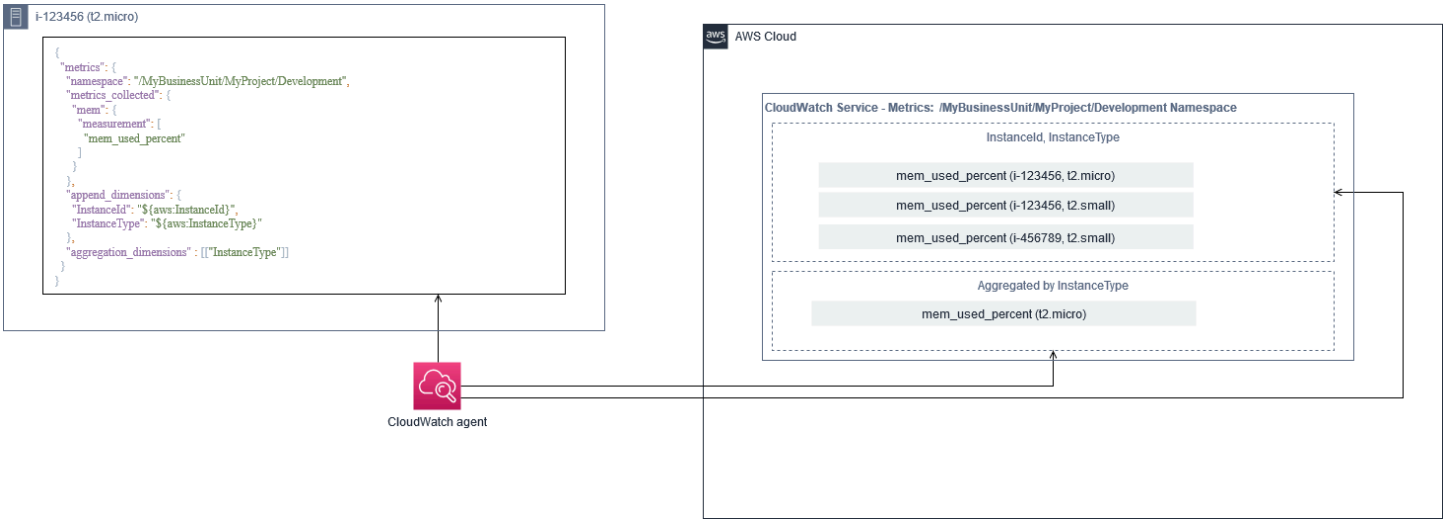
지표는 차원으로도 식별되므로 일련의 조건을 기준으로 지표를 분석하는 데 도움이 되며 관찰이 기록되는 속성입니다. Amazon EC2에는 InstanceId및 AutoScalingGroupName차원이 있는 EC2 인스턴스에 대한 별도의 지표가 포함되어 있습니다. 세부 모니터링을 활성화하면 ImageId 및 InstanceType차원이 포함된 지표도 수신됩니다. 예를 들어 Amazon EC2는 InstanceId차원에 대한 별도의 CPU 사용률 지표 외에도 InstanceType차원과 함께 CPU 사용률에 대한 별도의 EC2 인스턴스 지표를 제공합니다. 이를 통해 특정 인스턴스 유형의 모든 EC2 인스턴스 외에도 각 고유 EC2 인스턴스의 CPU 사용률을 분석할 수 있습니다.

차원을 더 추가하면 각 지표와 고유한 차원 값 조합으로 인해 새 지표가 생성되므로 분석 기능이 향상되지만 전체 비용도 증가합니다. 예를 들어 InstanceId차원에 대한 메모리 사용률에 대한 지표를 생성하는 경우 이는 각 EC2 인스턴스에 대한 새 지표입니다. 조직에서 수천 개의 EC2 인스턴스를 실행하는 경우 수천 개의 지표가 발생하고 비용이 증가합니다. 비용을 제어하고 예측하려면 지표의 카디널리티와 가장 큰 가치를 더하는 차원을 결정해야 합니다. 예를 들어 프로덕션 워크로드 지표에 대한 전체 차원 집합을 정의할 수 있지만 비프로덕션 워크로드에 대한 이러한 차원의 하위 집합은 더 작을 수 있습니다.

append_dimensions 속성을 사용하여 CloudWatch 구성에 정의된 하나 또는 모든 지표에 차원을 추가할 수 있습니다. 또한 CloudWatch 구성의 모든 지표에 ImageId, InstanceType, 및 InstanceIdAutoScalingGroupName를 동적으로 추가할 수 있습니다. 또는 해당 지표의 append_dimensions 속성을 사용하여 특정 지표에 대한 임의의 차원 이름과 값을 추가할 수 있습니다. CloudWatch는 aggregation_dimensions 속성으로 정의한 지표 차원에 대한 통계도 집계할 수 있습니다.

예를 들어 InstanceType차원에 대해 사용된 메모리를 집계하여 각 인스턴스 유형에 대해 모든 EC2 인스턴스에서 사용된 평균 메모리를 확인할 수 있습니다. 리전에서 실행되는 t2.micro 인스턴스를 사용하는 경우 t2.micro 클래스를 사용하는 워크로드가 제공된 메모리를 과도하게 사용하고 있는지 아니면 과소 사용하고 있는지 확인할 수 있습니다. 사용률이 낮은 것은 불필요한 메모리 용량이 있는 EC2 클래스를 사용하는 워크로드의 징후일 수 있습니다. 반면, 과다 사용은 메모리가 부족한 Amazon EC2 클래스를 사용하는 워크로드의 징후일 수 있습니다.

다음 다이어그램은 사용자 지정 네임스페이스, 추가된 차원 및에 의한 집계를 사용하는 샘플 CloudWatch 지표 구성을 보여줍니다InstanceType.



시스템 수준 CloudWatch 구성

시스템 수준 지표 및 로그는 모니터링 및 로깅 솔루션의 핵심 구성 요소이며 CloudWatch 에이전트에는 Windows 및 Linux에 대한 특정 구성 옵션이 있습니다.

CloudWatch 구성 파일 마법사 또는 구성 파일 스키마를 사용하여 지원하려는 각 OS에 대한 CloudWatch 에이전트 구성 파일을 정의하는 것이 좋습니다. 워크로드별 OS 수준 로그 및 지표를 별도의 CloudWatch 구성 파일에 정의하고 표준 구성에 추가할 수 있습니다. 이러한 고유한 구성 파일은 EC2 인스턴스에서 검색할 수 있는 S3 버킷에 별도로 저장해야 합니다. 이 목적을 위한 S3 버킷 설정의 예는이 가이드의 CloudWatch 구성 관리 섹션에 설명되어 있습니다. State Manager 및 Distributor를 사용하여 이러한 구성을 자동으로 검색하고 적용할 수 있습니다.

시스템 수준 로그 구성

시스템 수준 로그는 온프레미스 또는 AWS 클라우드에서 문제를 진단하고 해결하는 데 필수적입니다. 로그 캡처 접근 방식에는 OS에서 생성된 모든 시스템 및 보안 로그가 포함되어야 합니다. OS 생성 로그 파일은 OS 버전에 따라 다를 수 있습니다.

CloudWatch 에이전트는 이벤트 로그 이름을 제공하여 Windows 이벤트 로그 모니터링을 지원합니다. 모니터링할 Windows 이벤트 로그를 선택할 수 있습니다(예: `System`, `Application`또는 `Security`).

Linux 시스템의 시스템, 애플리케이션 및 보안 로그는 일반적으로 `/var/log` 디렉터리에 저장됩니다. 다음 표에서는 모니터링해야 하는 일반적인 기본 로그 파일을 정의하지만 `/etc/rsyslog.conf` 또는 `/etc/syslog.conf` 파일을 확인하여 시스템 로그 파일의 특정 설정을 결정해야 합니다.

Fedora 배포	`/var/log/boot.log*` - 부팅 로그
(Amazon Linux, CentOS, Red Hat Enterprise Linux)	`/var/log/dmesg` - 커널 로그
	`/var/log/secure` - 보안 및 인증 로그
	`/var/log/messages` - 일반 시스템 로그
	`/var/log/cron*` - Cron Logs
	`/var/log/cloud-init-output.log` - Userdata 시작 스크립트의 출력
Debian	`/var/log/syslog` - 부팅 로그
(Ubuntu)	`/var/log/cloud-init-output.log` - Userdata 시작 스크립트의 출력
	`/var/log/auth.log` - 보안 및 인증 로그
	`/var/log/kern.log` - 커널 로그

조직에 모니터링하려는 로그를 생성하는 다른 에이전트 또는 시스템 구성 요소가 있을 수도 있습니다. 이러한 에이전트 또는 애플리케이션에서 생성되는 로그 파일을 평가 및 결정하고 파일 위치를 식별하여 구성에 포함해야 합니다. 예를 들어 구성에 Systems Manager 및 CloudWatch 에이전트 로그를 포함해야 합니다. 다음 표에는 Windows 및 Linux에 대한 이러한 에이전트 로그의 위치가 나와 있습니다.

Windows	CloudWatch 에이전트	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\amazon-cloudwatch-agent.log`
	Systems Manager 에이전트	`%PROGRAMDATA%\Amazon\SSM\Logs\amazon-ssm-agent.log`

		%PROGRAMDATA%\Amazon\SSM\Logs\errors.log %PROGRAMDATA%\Amazon\SSM\Logs\audits\amazon-ssm-agent-audit-YYYY-MM-DD
Linux	CloudWatch 에이전트	/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log
	Systems Manager 에이전트	/var/log/amazon/ssm/amazon-ssm-agent.log /var/log/amazon/ssm/errors.log /var/log/amazon/ssm/audits/amazon-ssm-agent-audit-YYYY-MM-DD

로그 파일이 CloudWatch 에이전트 구성에 정의되어 있지만 찾을 수 없는 경우 CloudWatch는 로그 파일을 무시합니다. 이는 각 배포에 대한 별도의 구성 대신 Linux에 대한 단일 로그 구성을 유지하려는 경우에 유용합니다. 에이전트 또는 소프트웨어 애플리케이션이 실행되기 시작할 때까지 로그 파일이 없는 경우에도 유용합니다.

시스템 수준 지표 구성

메모리 및 디스크 공간 사용률은 Amazon EC2에서 제공하는 표준 지표에 포함되지 않습니다. 이러한 지표를 포함하려면 EC2 인스턴스에 CloudWatch 에이전트를 설치하고 구성해야 합니다. CloudWatch 에이전트 구성 마법사는 미리 정의된 지표가 포함된 CloudWatch 구성을 생성하며 필요에 따라 지표를 추가하거나 제거할 수 있습니다. 사전 정의된 지표 세트를 검토하여 필요한 적절한 수준을 결정해야 합니다.

최종 사용자 및 워크로드 소유자는 서버 또는 EC2 인스턴스에 대한 특정 요구 사항에 따라 추가 시스템 지표를 게시해야 합니다. 이러한 지표 정의는 별도의 CloudWatch 에이전트 구성 파일에 저장, 버전 관리 및 유지 관리해야 하며 재사용 및 자동화를 위해 중앙 위치(예: Amazon S3)에서 공유해야 합니다.

표준 Amazon EC2 지표는 온프레미스 서버에서 자동으로 캡처되지 않습니다. 이러한 지표는 온프레미스 인스턴스에서 사용하는 CloudWatch 에이전트 구성 파일에 정의되어야 합니다. CPU 사용률과 같은 지표를 사용하여 온프레미스 인스턴스에 대한 별도의 지표 구성 파일을 생성하고 이러한 지표를 표준 지표 구성 파일에 추가할 수 있습니다.

애플리케이션 수준 CloudWatch 구성

애플리케이션 로그 및 지표는 애플리케이션을 실행하여 생성되며 애플리케이션별로 다릅니다. 조직에서 정기적으로 사용하는 애플리케이션을 적절하게 모니터링하는 데 필요한 로그와 지표를 정의해야 합니다. 예를 들어 조직에서 웹 기반 애플리케이션을 위해 Microsoft Internet Information Server(IIS)를 표준화했을 수 있습니다. 조직 전체에서 사용할 수 있는 IIS에 대한 표준 로그 및 지표 CloudWatch 구성을 생성할 수 있습니다. 애플리케이션별 구성 파일은 중앙 위치(예: S3 버킷)에 저장할 수 있으며 워크로드 소유자 또는 자동 검색을 통해 액세스하고 CloudWatch 구성 디렉터리에 복사됩니다. CloudWatch 에이전트는 각 EC2 인스턴스 또는 서버의 구성 파일 디렉터리에 있는 CloudWatch 구성 파일을 복합 CloudWatch 구성으로 자동으로 결합합니다. 최종 결과는 조직의 표준 시스템 수준 구성과 모든 관련 애플리케이션 수준 CloudWatch 구성을 포함하는 CloudWatch 구성입니다.

워크로드 소유자는 모든 중요한 애플리케이션 및 구성 요소에 대한 로그 파일 및 지표를 식별하고 구성해야 합니다.

애플리케이션 수준 로그 구성

애플리케이션 수준 로깅은 애플리케이션이 상용 off-the-shelf(COTS)인지 아니면 사용자 지정 개발 애플리케이션인지에 따라 달라집니다. COTS 애플리케이션 및 해당 구성 요소는 로그 세부 정보 수준, 로그 파일 형식, 로그 파일 위치 등 로그 구성 및 출력에 대한 여러 옵션을 제공할 수 있습니다. 그러나 대부분의 COTS 또는 타사 애플리케이션에서는 로깅을 근본적으로 변경할 수 없습니다(예: 구성 불가능한 추가 로그 문 또는 형식을 포함하도록 애플리케이션 코드 업데이트). 최소한 COTS 또는 타사 애플리케이션에 대한 로깅 옵션을 구성하여 경고 및 오류 수준 정보를 가급적이면 JSON 형식으로 로깅해야 합니다.

CloudWatch 구성에 애플리케이션의 로그 파일을 포함하여 사용자 지정 개발 애플리케이션을 CloudWatch Logs와 통합할 수 있습니다. 사용자 지정 애플리케이션은 추가 필수 세부 정보를 포함하는 것 외에도 로그 출력 형식을 사용자 지정하고 구성 요소 출력을 별도의 로그 파일로 분류 및 분리할 수 있으므로 더 나은 로그 품질과 제어를 제공합니다. 분석 및 처리가 더 쉬워지도록 로깅 라이브러리와 조직에 필요한 데이터 및 형식을 검토하고 표준화해야 합니다.

CloudWatch Logs PutLogEvents API 호출을 사용하거나 AWS SDK를 사용하여 CloudWatch 로그 스트림에 쓸 수도 있습니다. 분산된 구성 요소 및 서버 집합에서 단일 로그 스트림에 대한 로깅 조정과 같은 사용자 지정 로깅 요구 사항에 API 또는 SDK를 사용할 수 있습니다. 그러나 유지 관리가 가장 쉽고 가장 광범위하게 적용할 수 있는 솔루션은 로그 파일에 쓰도록 애플리케이션을 구성한 다음 CloudWatch 에이전트를 사용하여 로그 파일을 읽고 CloudWatch로 스트리밍하는 것입니다.

또한 애플리케이션 로그 파일에서 측정하려는 지표의 종류를 고려해야 합니다. 지표 필터를 사용하여 CloudWatch 로그 그룹에서이 데이터를 측정, 그래프 및 경보할 수 있습니다. 예를 들어 지표 필터를 사용하여 로그에서 실패한 로그인 시도를 식별하여 실패한 로그인 시도 수를 계산할 수 있습니다.

애플리케이션 로그 파일에 CloudWatch 임베디드 지표 형식을 사용하여 사용자 지정 개발 애플리케이션에 대한 사용자 지정 지표를 생성할 수도 있습니다.

애플리케이션 수준 지표 구성

사용자 지정 지표는 AWS 서비스에서 CloudWatch에 직접 제공하지 않는 지표이며 CloudWatch 지표의 사용자 지정 네임스페이스에 게시됩니다. 모든 애플리케이션 지표는 사용자 지정 CloudWatch 지표로 간주됩니다. 애플리케이션 지표는 EC2 인스턴스, 애플리케이션 구성 요소, API 직접 호출 또는 심지어 비즈니스 함수에도 부합할 수 있습니다. 또한 지표에 대해 선택한 차원의 중요도와 카디널리티를 고려해야 합니다. 카디널리티가 높은 차원은 많은 수의 사용자 지정 지표를 생성하여 CloudWatch 비용을 증가시킬 수 있습니다.

CloudWatch를 사용하면 다음을 포함한 여러 가지 방법으로 애플리케이션 수준 지표를 캡처할 수 있습니다.

- procstat 플러그인에서 캡처하려는 개별 프로세스를 정의하여 프로세스 수준 지표를 캡처합니다.
- 애플리케이션은 Windows 성능 모니터에 지표를 게시하며이 지표는 CloudWatch 구성에 정의되어 있습니다.
- 지표 필터 및 패턴은 CloudWatch의 애플리케이션 로그에 적용됩니다.
- 애플리케이션은 CloudWatch 임베디드 지표 형식을 사용하여 CloudWatch 로그에 씁니다.
- 애플리케이션은 API 또는 AWS SDK를 통해 CloudWatch로 지표를 전송합니다.
- 애플리케이션은 CloudWatch 에이전트가 구성된 수집 또는 StatsD 데몬으로 지표를 전송합니다.

procstat를 사용하여 CloudWatch 에이전트로 중요한 애플리케이션 프로세스를 모니터링하고 측정할 수 있습니다. 이렇게 하면 애플리케이션에 대해 중요한 프로세스가 더 이상 실행되지 않는 경우 경보를 발생시키고 조치(예: 알림 또는 재시작 프로세스)를 취할 수 있습니다. 또한 애플리케이션 프로세스의 성능 특성을 측정하고 특정 프로세스가 비정상적으로 작동하는 경우 경보를 발생시킬 수 있습니다.

Procstat 모니터링은 추가 사용자 지정 지표로 COTS 애플리케이션을 업데이트할 수 없는 경우에도 유용합니다. 예를 들어를 측정하고 사용자 지정 `my_process` `application_version`차원을 `cpu_time` 포함하는 지표를 생성할 수 있습니다. 지표마다 차원이 다른 경우 애플리케이션에 여러 CloudWatch 에이전트 구성 파일을 사용할 수도 있습니다.

애플리케이션이 Windows에서 실행되는 경우 이미 Windows 성능 모니터에 지표를 게시하는지 평가해야 합니다. 많은 COTS 애플리케이션이 Windows 성능 모니터와 통합되어 애플리케이션 지표를 쉽게 모니터링할 수 있습니다. 또한 CloudWatch는 Windows 성능 모니터와 통합되며 이미 사용 가능한 모든 지표를 캡처할 수 있습니다.

애플리케이션에서 제공하는 로깅 형식과 로그 정보를 검토하여 지표 필터로 추출할 수 있는 지표를 결정해야 합니다. 애플리케이션의 기록 로그를 검토하여 오류 메시지와 비정상적인 종료가 어떻게 표시되는지 확인할 수 있습니다. 또한 이전에 보고된 문제를 검토하여 문제가 반복되지 않도록 지표를 캡처할 수 있는지 확인해야 합니다. 또한 애플리케이션의 설명서를 검토하고 애플리케이션 개발자에게 오류 메시지를 식별할 수 있는 방법을 확인하도록 요청해야 합니다.

사용자 지정 개발 애플리케이션의 경우 애플리케이션 개발자와 협력하여 CloudWatch 임베디드 지표 형식, AWS SDK 또는 AWS API를 사용하여 구현할 수 있는 중요한 지표를 정의합니다. 권장 접근 방식은 임베디드 지표 형식을 사용하는 것입니다. AWS 제공된 오픈 소스 임베디드 지표 형식 라이브러리를 사용하여 필요한 형식으로 문을 작성할 수 있습니다. 또한 임베디드 지표 형식 에이전트를 포함하도록 애플리케이션별 CloudWatch 구성을 업데이트해야 합니다. 이렇게 하면 EC2 인스턴스에서 실행되는 에이전트가 임베디드 지표 형식 지표를 CloudWatch로 전송하는 로컬 임베디드 지표 형식 엔드포인트 역할을 합니다.

애플리케이션이 이미 수집 또는 통계에 지표 게시를 지원하는 경우 이를 활용하여 지표를 CloudWatch에 수집할 수 있습니다.

Amazon EC2 및 온프레미스 서버에 대한 CloudWatch 에이전트 설치 접근 방식

CloudWatch 에이전트의 설치 프로세스를 자동화하면 빠르고 일관되게 배포하고 필요한 로그 및 지표를 캡처할 수 있습니다. 다중 계정 및 다중 리전 지원을 포함하여 CloudWatch 에이전트 설치를 자동화하는 몇 가지 접근 방식이 있습니다. 다음과 같은 자동 설치 접근 방식에 대해 설명합니다.

- Systems Manager Distributor 및 Systems Manager State Manager를 사용하여 CloudWatch 에이전트 설치 - EC2 인스턴스 및 온프레미스 서버가 Systems Manager 에이전트를 실행하는 경우 이 접근 방식을 사용하는 것이 좋습니다. 이렇게 하면 CloudWatch 에이전트가 계속 업데이트되고 CloudWatch 에이전트가 없는 서버를 보고하고 수정할 수 있습니다. 또한이 접근 방식은 여러 계정 및 리전을 지원하도록 확장됩니다.
- EC2 인스턴스 프로비저닝 중에 사용자 데이터 스크립트의 일부로 CloudWatch 에이전트 배포 - Amazon EC2를 사용하면 처음 부팅하거나 재부팅할 때 실행되는 시작 스크립트를 정의할 수 있습니다. 스크립트를 정의하여 에이전트의 다운로드 및 설치 프로세스를 자동화할 수 있습니다. 스크립트 CloudFormation 및 AWS Service Catalog 제품에도 포함될 수 있습니다. 이 접근 방식은 표준과 다른 특정 워크로드에 대한 사용자 지정 에이전트 설치 및 구성 접근 방식이 있는 경우 필요에 따라 적절할 수 있습니다.
- Amazon Machine Image(AMIs)에 CloudWatch 에이전트 포함 - Amazon EC2용 사용자 지정 AMIs에 CloudWatch 에이전트를 설치할 수 있습니다. AMI를 사용하는 EC2 인스턴스에는 에이전트가 자동으로 설치 및 시작됩니다. 그러나 에이전트와 해당 구성이 정기적으로 업데이트되도록 해야 합니다.

Systems Manager Distributor 및 State Manager를 사용하여 CloudWatch 에이전트 설치

Systems Manager Distributor와 함께 Systems Manager State Manager를 사용하여 서버 및 EC2 인스턴스에 CloudWatch 에이전트를 자동으로 설치하고 업데이트할 수 있습니다. Distributor에는 최신 CloudWatch 에이전트 버전을 설치하는 `AmazonCloudWatchAgent` AWS 관리형 패키지가 포함되어 있습니다.

이 설치 접근 방식에는 다음과 같은 사전 조건이 있습니다.

- Systems Manager 에이전트가 서버 또는 EC2 인스턴스에 설치되어 실행 중이어야 합니다. Systems Manager 에이전트는 Amazon Linux, Amazon Linux 2 및 일부 AMIs. 에이전트는 다른 이미지 또는 온프레미스 VMs 및 서버에도 설치 및 구성되어야 합니다.

> **Note**
>
> Amazon Linux 2는 곧 지원이 종료됩니다. 자세한 내용은 Amazon Linux 2 FAQ를 참조하세요.

- 필요한 CloudWatch 및 Systems Manager 권한이 있는 IAM 역할 또는 자격 증명은 EC2 인스턴스에 연결하거나 온프레미스 서버의 자격 증명 파일에 정의해야 합니다. 예를 들어 AWS 관리형 정책인 Systems Manager`AmazonSSMManagedInstanceCore`용 및 CloudWatch`CloudWatchAgentServerPolicy`용를 포함하는 IAM 역할을 생성할 수 있습니다. ssm-cloudwatch-instance-role.yaml CloudFormation 템플릿을 사용하여 이러한 두 정책을 모두 포함하는 IAM 역할 및 인스턴스 프로파일을 배포할 수 있습니다. 이 템플릿은 EC2 인스턴스에 대한 다른 표준 IAM 권한을 포함하도록 수정할 수도 있습니다. 온프레미스 서버 또는 VMs 경우는 온프레미스 서버에 대해 구성된 Systems Manager 서비스 역할을 사용하도록 CloudWatch 에이전트를 구성해야 합니다. 이에 대한 자세한 내용은 AWS 지식 센터의 Systems Manager 에이전트 및 통합 CloudWatch 에이전트를 사용하여 임시 자격 증명만 사용하도록 온프레미스 서버를 구성하려면 어떻게 해야 합니까?를 참조하세요.

다음 목록은 Systems Manager Distributor 및 State Manager 접근 방식을 사용하여 CloudWatch 에이전트를 설치하고 유지 관리할 때의 몇 가지 이점을 제공합니다.

- 여러 OSs의 자동 설치 - CloudWatch 에이전트를 다운로드하고 설치하기 위해 각 OS에 대한 스크립트를 작성하고 유지 관리할 필요가 없습니다.
- 자동 업데이트 확인 - 상태 관리자는 각 EC2 인스턴스에 최신 CloudWatch 버전이 있는지 자동으로 정기적으로 확인합니다.
- 규정 준수 보고 - Systems Manager 규정 준수 대시보드에는 Distributor 패키지를 성공적으로 설치하지 못한 EC2 인스턴스가 표시됩니다.
- 새로 시작된 EC2 인스턴스의 자동 설치 - 계정으로 시작된 새 EC2 인스턴스는 자동으로 CloudWatch 에이전트를 수신합니다.

그러나이 접근 방식을 선택하기 전에 다음 세 가지 영역도 고려해야 합니다.

- 기존 연결과의 충돌 - 다른 연결이 이미 CloudWatch 에이전트를 설치하거나 구성하는 경우 두 연결이 서로 간섭하여 잠재적으로 문제를 일으킬 수 있습니다. 이 접근 방식을 사용할 때는 CloudWatch 에이전트 및 구성을 설치하거나 업데이트하는 기존 연결을 제거해야 합니다.

- 사용자 지정 에이전트 구성 파일 업데이트 - Distributor는 기본 구성 파일을 사용하여 설치를 수행합니다. 사용자 지정 구성 파일 또는 여러 CloudWatch 구성 파일을 사용하는 경우 설치 후 구성을 업데이트해야 합니다.
- 다중 리전 또는 다중 계정 설정 - 각 계정 및 리전에서 State Manager 연결을 설정해야 합니다. 상태 관리자 연결을 포함하도록 다중 계정 환경의 새 계정을 업데이트해야 합니다. 여러 계정 및 리전이 필요한 표준을 검색하고 적용할 수 있도록 CloudWatch 구성을 중앙 집중화하거나 동기화해야 합니다.

CloudWatch 에이전트 배포 및 구성을 위한 State Manager 및 Distributor 설정

Systems Manager 빠른 설정을 사용하여 EC2 인스턴스에 CloudWatch 에이전트를 자동으로 설치 및 업데이트하는 등 Systems Manager 기능을 빠르게 구성할 수 있습니다. 빠른 설정은 선택에 따라 Systems Manager 리소스를 배포하고 구성하는 CloudFormation 스택을 배포합니다.

다음 목록은 자동 CloudWatch 에이전트 설치 및 업데이트를 위해 빠른 설정에서 수행하는 두 가지 중요한 작업을 제공합니다.

1. Systems Manager 사용자 지정 문서 생성 - 빠른 설정은 State Manager와 함께 사용할 다음과 같은 Systems Manager 문서를 생성합니다. 문서 이름은 다를 수 있지만 콘텐츠는 동일하게 유지됩니다.
   - `CreateAndAttachIAMToInstance` - `AmazonSSMRoleForInstancesQuickSetup` 역할 및 인스턴스 프로파일이 없는 경우 생성하고 `AmazonSSMManagedInstanceCore` 정책을 역할에 연결합니다. 여기에는 필수 `CloudWatchAgentServerPolicy` IAM 정책이 포함되지 않습니다. 다음 섹션에 설명된 대로이 정책을 업데이트하고이 정책을 포함하도록이 Systems Manager 문서를 업데이트해야 합니다.
   - `InstallAndManageCloudWatchDocument` - Distributor와 함께 CloudWatch 에이전트를 설치하고 `AWS-ConfigureAWSPackage` Systems Manager 문서를 사용하여 기본 CloudWatch 에이전트 구성으로 각 EC2 인스턴스를 한 번 구성합니다.
   - `UpdateCloudWatchDocument` - `AWS-ConfigureAWSPackage` Systems Manager 문서를 사용하여 최신 CloudWatch 에이전트를 설치하여 CloudWatch 에이전트를 업데이트합니다. 에이전트를 업데이트하거나 제거해도 EC2 인스턴스에서 기존 CloudWatch 구성 파일은 제거되지 않습니다.
2. 상태 관리자 연결 생성 - 상태 관리자 연결이 생성되고 사용자 지정으로 생성된 Systems Manager 문서를 사용하도록 구성됩니다. State Manager 연결 이름은 다를 수 있지만 구성은 동일하게 유지됩니다.

- `ManageCloudWatchAgent` - 각 EC2 인스턴스에 대해 `InstallAndManageCloudWatchDocument` Systems Manager 문서를 한 번 실행합니다.
- `UpdateCloudWatchAgent` - 각 EC2 인스턴스에 대해 30일마다 `UpdateCloudWatchDocument` Systems Manager 문서를 실행합니다.
- 각 EC2 인스턴스에 대해 `CreateAndAttachIAMToInstance` Systems Manager 문서를 한 번 실행합니다.

CloudWatch 권한을 포함하고 사용자 지정 CloudWatch 구성을 지원하려면 완료된 빠른 설정 구성을 보강하고 사용자 지정해야 합니다. 특히 `CreateAndAttachIAMToInstance` 및 `InstallAndManageCloudWatchDocument` 문서를 업데이트해야 합니다. 빠른 설정에서 생성한 Systems Manager 문서를 수동으로 업데이트할 수 있습니다. 또는 자체 CloudFormation 템플릿을 사용하여 필요한 업데이트로 동일한 리소스를 프로비저닝하고 다른 Systems Manager 리소스를 구성 및 배포할 수 있으며 빠른 설정을 사용하지 않을 수 있습니다.

> **Important**
>
> 빠른 설정은 스택을 생성 CloudFormation 하여 선택에 따라 Systems Manager 리소스를 배포하고 구성합니다. 빠른 설정 선택 사항을 업데이트하는 경우 Systems Manager 문서를 수동으로 다시 업데이트해야 할 수 있습니다.

다음 섹션에서는 빠른 설정에서 생성한 Systems Manager 리소스를 수동으로 업데이트하고 자체 CloudFormation 템플릿을 사용하여 업데이트된 빠른 설정을 수행하는 방법을 설명합니다. 빠른 설정 및에서 생성한 리소스를 수동으로 업데이트하지 않으려면 자체 CloudFormation 템플릿을 사용하는 것이 좋습니다 CloudFormation.

Systems Manager 빠른 설정을 사용하고 생성된 Systems Manager 리소스를 수동으로 업데이트합니다.

빠른 설정 접근 방식으로 생성된 Systems Manager 리소스는 필요한 CloudWatch 에이전트 권한을 포함하고 여러 CloudWatch 구성 파일을 지원하도록 업데이트해야 합니다. 이 섹션에서는 여러 계정에서 액세스할 수 있는 CloudWatch 구성이 포함된 중앙 집중식 S3 버킷을 사용하도록 IAM 역할 및 Systems Manager 문서를 업데이트하는 방법을 설명합니다. CloudWatch 구성 파일을 저장할 S3 버킷 생성은이 가이드의 CloudWatch 구성 관리 섹션에서 설명합니다.

`CreateAndAttachIAMToInstance` Systems Manager 문서 업데이트

빠른 설정에서 생성한이 Systems Manager 문서는 EC2 인스턴스에 기존 IAM 인스턴스 프로파일이 연결되어 있는지 확인합니다. 이 경우 기존 역할에 `AmazonSSMManagedInstanceCore` 정책을 연결합니다. 이렇게 하면 기존 EC2 인스턴스가 기존 인스턴스 프로파일을 통해 할당될 수 있는 AWS 권한을 상실하지 않도록 보호할 수 있습니다. 이미 인스턴스 프로파일이 연결되어 있는 EC2 인스턴스에 `CloudWatchAgentServerPolicy` IAM 정책을 연결하려면이 문서에 단계를 추가해야 합니다. Systems Manager 문서는 IAM 역할이 없고 EC2 인스턴스에 인스턴스 프로파일이 연결되어 있지 않은 경우에도 IAM 역할을 생성합니다. 문서의이 섹션을 업데이트하여 `CloudWatchAgentServerPolicy` IAM 정책도 포함해야 합니다.

완료된 CreateAndAttachIAMToInstance.yaml 샘플 문서를 검토하고 빠른 설정에서 생성한 문서와 비교합니다. 필요한 단계와 변경 사항을 포함하도록 기존 문서를 편집합니다. 빠른 설정 선택에 따라 빠른 설정에서 생성한 문서가 제공된 샘플 문서와 다를 수 있으므로 필요한 조정을 수행해야 합니다. 샘플 문서에는 매일 누락된 패치가 있는지 인스턴스를 스캔하는 빠른 설정 옵션 선택 사항이 포함되어 있으므로 Systems Manager Patch Manager에 대한 정책이 포함되어 있습니다.

`InstallAndManageCloudWatchDocument` Systems Manager 문서 업데이트

빠른 설정에서 생성한이 Systems Manager 문서는 CloudWatch 에이전트를 설치하고 기본 CloudWatch 에이전트 구성으로 구성합니다. 기본 CloudWatch 구성은 사전 정의된 기본 지표 세트에 맞춰집니다. 기본 구성 단계를 바꾸고 CloudWatch 구성 S3 버킷에서 CloudWatch 구성 파일을 다운로드하는 단계를 추가해야 합니다.

완료된 InstallAndManageCloudWatchDocument.yaml 업데이트 문서를 검토하고 Quick Setup에서 생성한 문서와 비교합니다. 빠른 설정에서 생성한 문서는 다를 수 있으므로 필요한 조정을 수행했는지 확인하세요. 필요한 단계와 변경 사항을 포함하도록 기존 문서를 편집합니다.

빠른 설정 CloudFormation 대신 사용

빠른 설정을 사용하는 대신를 사용하여 Systems Manager CloudFormation 를 구성할 수 있습니다. 이 접근 방식을 사용하면 특정 요구 사항에 따라 Systems Manager 구성을 사용자 지정할 수 있습니다. 또한이 접근 방식은 사용자 지정 CloudWatch 구성을 지원하기 위해 빠른 설정에서 생성한 구성된 Systems Manager 리소스에 대한 수동 업데이트를 방지합니다.

빠른 설정 기능은를 CloudFormation 사용하고 CloudFormation 스택 세트를 생성하여 선택에 따라 Systems Manager 리소스를 배포하고 구성합니다. CloudFormation 스택 세트를 사용하려면 먼저 CloudFormation StackSets에서 사용하는 IAM 역할을 생성하여 여러 계정 또는 리전에 대한 배포를 지원해야 합니다. 빠른 설정은 CloudFormation StackSets를 사용하여 다중 리전 또는 다중 계정 배포

를 지원하는 데 필요한 역할을 생성합니다. 여러 리전 또는 단일 계정 및 리전의 여러 계정에 Systems Manager 리소스를 구성하고 배포하려면 CloudFormation StackSets의 사전 조건을 완료해야 합니다. 이에 대한 자세한 내용은 CloudFormation 설명서의 스택 세트 작업을 위한 사전 조건을 참조하세요.

사용자 지정 빠른 설정에 대한 AWS-QuickSetup-SSMHostMgmt.yaml CloudFormation 템플릿을 검토합니다.

CloudFormation 템플릿의 리소스와 기능을 검토하고 요구 사항에 따라 조정해야 합니다. 사용하는 CloudFormation 템플릿을 버전 제어하고 변경 사항을 점진적으로 테스트하여 필요한 결과를 확인해야 합니다. 또한 클라우드 보안 검토를 수행하여 조직의 요구 사항에 따라 필요한 정책 조정이 있는지 확인해야 합니다.

CloudFormation 스택을 단일 테스트 계정 및 리전에 배포하고 필요한 테스트 사례를 수행하여 원하는 결과를 사용자 지정하고 확인해야 합니다. 그런 다음 단일 계정의 여러 리전으로 배포를 완료한 다음 여러 계정과 여러 리전으로 배포할 수 있습니다.

CloudFormation 스택이 있는 단일 계정 및 리전의 사용자 지정 빠른 설정

단일 계정 및 리전만 사용하는 경우 전체 예제를 CloudFormation 스택 세트 대신 CloudFormation 스택으로 배포할 수 있습니다. 그러나 가능하면 단일 계정 및 리전만 사용하더라도 다중 계정, 다중 리전 스택 세트 접근 방식을 사용하는 것이 좋습니다. CloudFormation StackSets를 사용하면 향후 추가 계정 및 리전으로 더 쉽게 확장할 수 있습니다.

AWSAWS-QuickSetup-SSMHostMgmt.yaml CloudFormation 템플릿을 단일 계정의 CloudFormation 스택으로 배포하려면 AWS 리전다음 단계를 사용합니다.

1. 템플릿을 다운로드하여 원하는 버전 관리 시스템(예: GitHub)에 확인합니다.
2. 조직의 요구 사항에 따라 기본 CloudFormation 파라미터 값을 사용자 지정합니다.
3. 상태 관리자 연결 일정을 사용자 지정합니다.
4. `InstallAndManageCloudWatchDocument` 논리적 ID로 Systems Manager 문서를 사용자 지정합니다. S3 버킷 접두사가 CloudWatch 구성이 포함된 S3 버킷의 접두사와 일치하는지 확인합니다.
5. CloudWatch 구성이 포함된 S3 버킷의 Amazon 리소스 이름(ARN)을 검색하고 기록합니다. 이에 대한 자세한 내용은이 가이드의 CloudWatch 구성 관리 섹션을 참조하세요. AWS Organizations 계정에 대한 읽기 액세스를 제공하는 버킷 정책이 포함된 샘플 cloudwatch-config-s3-bucket.yaml CloudFormation 템플릿을 사용할 수 있습니다.
6. 사용자 지정된 빠른 설정 CloudFormation 템플릿을 S3 버킷과 동일한 계정에 배포합니다.
   - `CloudWatchConfigBucketARN` 파라미터에 S3 버킷의 ARN을 입력합니다.

- Systems Manager에 대해 활성화하려는 기능에 따라 파라미터 옵션을 조정합니다.

7. IAM 역할이 있거나 없는 테스트 EC2 인스턴스를 배포하여 EC2 인스턴스가 CloudWatch에서 작동하는지 확인합니다.

- `AttachIAMToInstance` 상태 관리자 연결을 적용합니다. 일정에 따라 실행되도록 구성된 Systems Manager 실행서입니다. 실행서를 사용하는 상태 관리자 연결은 새 EC2 인스턴스에 자동으로 적용되지 않으며 예약된 시간에 실행되도록 구성할 수 있습니다. 자세한 내용은 Systems Manager 설명서의 State Manager를 사용하여 트리거로 자동화 실행을 참조하세요.
- EC2 인스턴스에 필수 IAM 역할이 연결되어 있는지 확인합니다.
- EC2 인스턴스가 Systems Manager에 표시되는지 확인하여 Systems Manager 에이전트가 올바르게 작동하는지 확인합니다.
- S3 버킷의 CloudWatch 구성을 기반으로 CloudWatch 로그 및 지표를 확인하여 CloudWatch 에이전트가 올바르게 작동하는지 확인합니다.

CloudFormation StackSets를 사용한 여러 리전 및 여러 계정의 사용자 지정 빠른 설정

여러 계정 및 리전을 사용하는 경우 AWS-QuickSetup-SSMHostMgmt.yaml CloudFormation 템플릿을 스택 세트로 배포할 수 있습니다. 스택 세트를 사용하기 전에 CloudFormation StackSet 사전 조건을 완료해야 합니다. 요구 사항은 자체 관리형 또는 서비스 관리형 권한으로 스택 세트를 배포하는지 여부에 따라 달라집니다.

새 계정이 사용자 지정된 빠른 설정을 자동으로 수신하도록 서비스 관리형 권한으로 스택 세트를 배포하는 것이 좋습니다. AWS Organizations 관리 계정 또는 위임된 관리자 계정에서 서비스 관리형 스택 세트를 배포해야 합니다. AWS Organizations 관리 계정이 아닌 위임된 관리자 권한이 있는 자동화에 사용되는 중앙 집중식 계정에서 스택 세트를 배포해야 합니다. 또한 한 리전에서 단일 또는 소수의 계정으로 테스트 조직 단위(OU)를 대상으로 스택 세트 배포를 테스트하는 것이 좋습니다.

1. 이 가이드의 CloudFormation 스택이 있는 단일 계정 및 리전의 사용자 지정 빠른 설정 섹션에서 1~5 단계를 완료합니다.
2. 에 로그인하고 CloudFormation 콘솔러를 AWS Management Console열고 StackSet 생성을 선택합니다.
   - 템플릿 준비 완료 및 템플릿 파일 업로드를 선택합니다. 요구 사항에 맞게 사용자 지정한 CloudFormation 템플릿을 업로드합니다.

- 스택 세트 세부 정보를 지정합니다.
  - 스택 세트 이름을 입력합니다. 예: `StackSet-SSM-QuickSetup`.
  - Systems Manager에 대해 활성화하려는 기능에 따라 파라미터 옵션을 조정합니다.
  - `CloudWatchConfigBucketARN` 파라미터에 CloudWatch 구성의 S3 버킷에 대한 ARN을 입력합니다.
  - 스택 세트 옵션을 지정하고 또는 AWS Organizations 자체 관리형 권한과 함께 서비스 관리형 권한을 사용할지 여부를 선택합니다.
    - 자체 관리형 권한을 선택하는 경우 AWSCloudFormationStackSetAdministrationRole 및 AWSCloudFormationStackSetExecutionRole IAM 역할 세부 정보를 입력합니다. 관리자 역할은 계정에 있어야 하며 실행 역할은 각 대상 계정에 있어야 합니다.
  - 를 사용한 서비스 관리형 권한의 경우 먼저 전체 조직 대신 테스트 OU에 배포하는 AWS Organizations것이 좋습니다.
    - 자동 배포를 활성화할지 여부를 선택합니다. 활성화됨을 선택하는 것이 좋습니다. 계정 제거 동작의 경우 권장 설정은 스택 삭제입니다.
  - 자체 관리형 권한의 경우 설정하려는 AWS 계정의 계정 IDs를 입력합니다. 자체 관리형 권한을 사용하는 경우 각 새 계정에 대해이 프로세스를 반복해야 합니다.
  - CloudWatch 및 Systems Manager를 사용할 리전을 입력합니다.
  - 스택 세트의 작업 및 스택 인스턴스 탭에서 상태를 확인하여 배포가 성공했는지 확인합니다.
  - 이 가이드 CloudFormation 스택이 있는 단일 계정 및 리전의 사용자 지정 빠른 설정 섹션의 7단계에 따라 Systems Manager와 CloudWatch가 배포된 계정에서 올바르게 작동하는지 테스트합니다.

온프레미스 서버 구성 시 고려 사항

온프레미스 서버 및 VMs용 CloudWatch 에이전트는 EC2 인스턴스와 유사한 접근 방식을 사용하여 설치 및 구성됩니다. 그러나 다음 표에는 온프레미스 서버 및 VM에 CloudWatch 에이전트를 설치하고 구성할 때 평가해야 하는 고려 사항이 나와 있습니다. VMs

CloudWatch 에이전트가 Systems Manager에 사용되는 것과 동일한 임시 자격 증명을 가리키도록 합니다.	온프레미스 서버가 포함된 하이브리드 환경에서 Systems Manager를 설정할 때 IAM 역할로 Systems Manager를 활성화할 수 있습니다. `CloudWatchAgentServerPolicy` 및 `AmazonSSMManagedInstanceCore` 정책

	이 포함된 EC2 인스턴스에 대해 생성된 역할을 사용해야 합니다. 그러면 Systems Manager 에이전트가 로컬 자격 증명 파일을 검색하여 임시 자격 증명을 작성합니다. CloudWatch 에이전트 구성이 동일한 파일을 가리키도록 할 수 있습니다. Systems Manager 에이전트와 통합 CloudWatch 에이전트를 사용하여 지식 센터의 임시 자격 증명만 사용하는 온프레미스 서버 구성의 프로세스를 사용할 수 있습니다. AWS 별도의 Systems Manager Automation 런북과 State Manager 연결을 정의하고 태그를 사용하여 온프레미스 인스턴스를 대상으로 지정하여이 프로세스를 자동화할 수도 있습니다. 온프레미스 인스턴스에 대해 Systems Manager 정품 인증을 생성할 때는 인스턴스를 온프레미스 인스턴스로 식별하는 태그를 포함해야 합니다.
VPN 또는 및 AWS PrivateLink에 Direct Connect 액세스할 수 있는 계정 및 리전을 사용하는 것이 좋습니다.	AWS Direct Connect 또는 AWS Virtual Private Network (Site-to-Site VPN)를 사용하여 온프레미스 네트워크와 Virtual Private Cloud(VPC) 간에 프라이빗 연결을 설정할 수 있습니다. AWS PrivateLink는 인터페이스 VPC 엔드포인트를 사용하여 CloudWatch Logs에 대한 프라이빗 연결을 설정합니다. 이 접근 방식은 퍼블릭 인터넷을 통해 퍼블릭 서비스 엔드포인트로 데이터가 전송되지 않도록 하는 제한이 있는 경우에 유용합니다.

모든 지표는 CloudWatch 구성 파일에 포함되어야 합니다.	Amazon EC2에는 표준 지표(예: CPU 사용률)가 포함되어 있지만 이러한 지표는 온프레미스 인스턴스에 대해 정의되어야 합니다. 별도의 플랫폼 구성 파일을 사용하여 온프레미스 서버에 대한 이러한 지표를 정의한 다음 플랫폼의 표준 CloudWatch 지표 구성에 구성을 추가할 수 있습니다.

임시 EC2 인스턴스에 대한 고려 사항

EC2 인스턴스는 Amazon EC2 Auto Scaling, Amazon EMR, Amazon EC2 스팟 인스턴스 또는에 의해 프로비저닝되는 경우 임시 또는 임시 인스턴스입니다 AWS Batch. Amazon EC2 임시 EC2 인스턴스는 런타임 오리진에 대한 추가 정보 없이도 공통 로그 그룹에서 매우 많은 수의 CloudWatch 스트림을 유발할 수 있습니다.

임시 EC2 인스턴스를 사용하는 경우 로그 그룹 및 로그 스트림 이름에 동적 컨텍스트 정보를 추가하는 것이 좋습니다. 예를 들어 스팟 인스턴스 요청 ID, Amazon EMR 클러스터 이름 또는 Auto Scaling 그룹 이름을 포함할 수 있습니다. 이 정보는 새로 시작된 EC2 인스턴스에 따라 다를 수 있으며 런타임 시 검색하고 구성해야 할 수 있습니다. 부팅 시 CloudWatch 에이전트 구성 파일을 작성하고 업데이트된 구성 파일을 포함하도록 에이전트를 다시 시작하여이 작업을 수행할 수 있습니다. 이를 통해 동적 런타임 정보를 사용하여 CloudWatch에 로그 및 지표를 전달할 수 있습니다.

또한 임시 EC2 인스턴스가 종료되기 전에 CloudWatch 에이전트가 지표와 로그를 전송해야 합니다. CloudWatch 에이전트에는 플러시 로그 및 지표 버퍼의 시간 간격을 정의하도록 구성할 수 있는 `flush_interval` 파라미터가 포함되어 있습니다. 워크로드를 기반으로이 값을 낮추고 CloudWatch 에이전트를 중지하고 EC2 인스턴스가 종료되기 전에 버퍼를 강제로 플러시할 수 있습니다.

자동화된 솔루션을 사용하여 CloudWatch 에이전트 배포

자동화 솔루션(예: Ansible 또는 Chef)을 사용하는 경우 이를 활용하여 CloudWatch 에이전트를 자동으로 설치하고 업데이트할 수 있습니다. 이 접근 방식을 사용하는 경우 다음 고려 사항을 평가해야 합니다.

- 자동화가 지원하는 OSs 및 OS 버전을 포함하는지 확인합니다. 자동화 스크립트가 조직의 모든 OSs를 지원하지 않는 경우 지원되지 않는 OSs에 대한 대체 솔루션을 정의해야 합니다.
- 자동화 솔루션이 CloudWatch 에이전트 업데이트 및 업그레이드를 정기적으로 확인하는지 확인합니다. 자동화 솔루션은 CloudWatch 에이전트에 대한 업데이트를 정기적으로 확인하거나 에이전트를

정기적으로 제거했다가 다시 설치해야 합니다. 스케줄러 또는 자동화 솔루션 기능을 사용하여 에이전트를 정기적으로 확인하고 업데이트할 수 있습니다.

- 에이전트 설치 및 구성 규정 준수를 확인할 수 있는지 확인합니다. 자동화 솔루션을 사용하면 시스템에 에이전트가 설치되어 있지 않거나 에이전트가 작동하지 않는 시기를 확인할 수 있습니다. 실패한 설치 및 구성을 추적하도록 자동화 솔루션에 알림 또는 경보를 구현할 수 있습니다.

사용자 데이터 스크립트를 사용하여 인스턴스 프로비저닝 중에 CloudWatch 에이전트 배포

Systems Manager를 사용할 계획이 없고 EC2 인스턴스에 CloudWatch를 선택적으로 사용하려는 경우이 접근 방식을 사용할 수 있습니다. 일반적으로이 접근 방식은 일회성으로 또는 특수 구성이 필요한 경우에 사용됩니다.는 시작 또는 사용자 데이터 스크립트에서 다운로드할 수 있는 CloudWatch 에이전트에 대한 직접 링크를 AWS 제공합니다. 에이전트 설치 패키지는 사용자 상호 작용 없이 자동으로 실행할 수 있으므로 자동 배포에서 사용할 수 있습니다. 이 접근 방식을 사용하는 경우 다음 고려 사항을 평가해야 합니다.

- 사용자가 에이전트를 설치하거나 표준 지표를 구성하지 않을 위험이 증가합니다. 사용자는 CloudWatch 에이전트를 설치하는 데 필요한 단계를 포함하지 않고 인스턴스를 프로비저닝할 수 있습니다. 또한 에이전트를 잘못 구성하여 로깅 및 모니터링 불일치가 발생할 수 있습니다.
- 설치 스크립트는 OS별로 달라야 하며 다양한 OS 버전에 적합해야 합니다. Windows와 Linux를 모두 사용하려는 경우 별도의 스크립트가 필요합니다. Linux 스크립트의 설치 단계도 배포에 따라 달라야 합니다.
- 사용 가능한 경우 CloudWatch 에이전트를 새 버전으로 정기적으로 업데이트해야 합니다. 이는 상태 관리자와 함께 Systems Manager를 사용하는 경우 자동화할 수 있지만 인스턴스 시작 시 다시 실행하도록 사용자 데이터 스크립트를 구성할 수도 있습니다. 그런 다음 재부팅할 때마다 CloudWatch 에이전트가 업데이트되고 다시 설치됩니다.
- 표준 CloudWatch 구성의 검색 및 적용을 자동화해야 합니다. 이는 상태 관리자와 함께 Systems Manager를 사용하는 경우 자동화할 수 있지만 부팅 시 구성 파일을 검색하고 CloudWatch 에이전트를 다시 시작하도록 사용자 데이터 스크립트를 구성할 수도 있습니다.

AMIs에 CloudWatch 에이전트 포함

이 접근 방식을 사용하면 CloudWatch 에이전트가 설치 및 구성될 때까지 기다릴 필요가 없으며 로깅 및 모니터링을 즉시 시작할 수 있다는 이점이 있습니다. 이렇게 하면 인스턴스가 시작되지 않는 경우 인스턴스 프로비저닝 및 시작 단계를 더 잘 모니터링할 수 있습니다. 이 접근 방식은 Systems Manager

에이전트를 사용할 계획이 없는 경우에도 적합합니다. 이 접근 방식을 사용하는 경우 다음 고려 사항을 평가해야 합니다.

- AMIs에 최신 CloudWatch 에이전트 버전이 포함되지 않을 수 있으므로 업데이트 프로세스가 있어야 합니다. AMI에 설치된 CloudWatch 에이전트는 AMI가 마지막으로 생성된 시점까지만 최신입니다. 에이전트를 정기적으로 업데이트하고 EC2 인스턴스가 프로비저닝될 때 추가 방법을 포함해야 합니다. Systems Manager를 사용하는 경우 이를 위해이 가이드에 제공된 Systems Manager Distributor 및 State Manager를 사용하여 CloudWatch 에이전트 설치 솔루션을 사용할 수 있습니다. Systems Manager를 사용하지 않는 경우 사용자 데이터 스크립트를 사용하여 인스턴스 시작 및 재부팅 시 에이전트를 업데이트할 수 있습니다.
- 인스턴스 시작 시 CloudWatch 에이전트 구성 파일을 검색해야 합니다. Systems Manager를 사용하지 않는 경우 부팅 시 구성 파일을 검색하도록 사용자 데이터 스크립트를 구성한 다음 CloudWatch 에이전트를 다시 시작할 수 있습니다.
- CloudWatch 구성이 업데이트된 후 CloudWatch 에이전트를 다시 시작해야 합니다.
- 자격 AWS 증명은 AMI에 저장해서는 안 됩니다. 로컬 AWS 자격 증명이 AMI에 저장되지 않았는지 확인합니다. Amazon EC2를 사용하는 경우 인스턴스에 필요한 IAM 역할을 적용하고 로컬 자격 증명을 피할 수 있습니다. 온프레미스 인스턴스를 사용하는 경우 CloudWatch 에이전트를 시작하기 전에 인스턴스 자격 증명을 자동화하거나 수동으로 업데이트해야 합니다.

Amazon ECS에서 로깅 및 모니터링

Amazon Elastic Container Service(Amazon ECS)는 컨테이너를 실행하기 위한 두 가지 시작 유형을 제공하며 작업 및 서비스를 호스팅하는 인프라 유형을 결정합니다. 이러한 시작 유형은 AWS Fargate 및 Amazon EC2입니다. 두 시작 유형 모두 CloudWatch와 통합되지만 구성 및 지원은 다릅니다.

다음 섹션에서는 Amazon ECS에서 로깅 및 모니터링에 CloudWatch를 사용하는 방법을 이해하는 데 도움이 됩니다.

주제

- EC2 시작 유형으로 CloudWatch 구성
- EC2 및 Fargate 시작 유형에 대한 Amazon ECS 컨테이너 로그
- FireLens for Amazon ECS에서 사용자 지정 로그 라우팅 사용
- Amazon ECS에 대한 지표

EC2 시작 유형으로 CloudWatch 구성

EC2 시작 유형을 사용하면 로깅 및 모니터링에 CloudWatch 에이전트를 사용하는 EC2 인스턴스의 Amazon ECS 클러스터를 프로비저닝합니다. Amazon ECS 최적화 AMI에는 Amazon ECS 컨테이너 에이전트가 사전 설치되어 있으며 Amazon ECS 클러스터에 대한 CloudWatch 지표를 제공합니다.

이러한 기본 지표는 Amazon ECS 비용에 포함되지만 Amazon ECS의 기본 구성은 로그 파일 또는 추가 지표(예: 여유 디스크 공간)를 모니터링하지 않습니다. 를 사용하여 EC2 시작 유형으로 Amazon ECS 클러스터를 프로비저닝 AWS Management Console 할 수 있습니다. 그러면 시작 구성으로 Amazon EC2 Auto Scaling 그룹을 배포하는 CloudFormation 스택이 생성됩니다. 그러나이 접근 방식은 사용자 지정 AMI를 선택하거나 다른 설정 또는 추가 부팅 스크립트로 시작 구성을 사용자 지정할 수 없음을 의미합니다.

추가 로그 및 지표를 모니터링하려면 Amazon ECS 컨테이너 인스턴스에 CloudWatch 에이전트를 설치해야 합니다. 이 가이드의 Systems Manager Distributor 및 State Manager를 사용하여 CloudWatch 에이전트 설치 섹션에서 EC2 인스턴스에 대한 설치 접근 방식을 사용할 수 있습니다. 그러나 Amazon ECS AMI에는 필수 Systems Manager 에이전트가 포함되지 않습니다. Amazon ECS 클러스터를 생성할 때 Systems Manager 에이전트를 설치하는 사용자 데이터 스크립트와 함께 사용자 지정 시작 구성을 사용해야 합니다. 이렇게 하면 컨테이너 인스턴스가 Systems Manager에 등록하고 상태 관리자 연결을 적용하여 CloudWatch 에이전트를 설치, 구성 및 업데이트할 수 있습니다. State Manager는 CloudWatch 에이전트 구성을 실행하고 업데이트할 때 Amazon EC2에 대해 표준화된 시스템 수준

CloudWatch 구성도 적용합니다. 또한 CloudWatch 구성의 S3 버킷에 Amazon ECS에 대한 표준화된 CloudWatch 구성을 저장하고 상태 관리자를 사용하여 자동으로 적용할 수 있습니다.

Amazon ECS 컨테이너 인스턴스에 적용된 IAM 역할 또는 인스턴스 프로파일에 필수 `CloudWatchAgentServerPolicy` 및 `AmazonSSMManagedInstanceCore` 정책이 포함되어 있는지 확인해야 합니다. ecs_cluster_with_cloudwatch_linux.yaml CloudFormation 템플릿을 사용하여 Linux 기반 Amazon ECS 클러스터를 프로비저닝할 수 있습니다. 이 템플릿은 Systems Manager를 설치하고 사용자 지정 CloudWatch 구성을 배포하여 Amazon ECS 관련 로그 파일을 모니터링하는 사용자 지정 시작 구성으로 Amazon ECS 클러스터를 생성합니다.

Amazon ECS 컨테이너 인스턴스에 대한 다음 로그와 표준 EC2 인스턴스 로그를 캡처해야 합니다.

- Amazon ECS 에이전트 시작 출력 - `/var/log/ecs/ecs-init.log`
- Amazon ECS 에이전트 출력 - `/var/log/ecs/ecs-agent.log`
- IAM 자격 증명 공급자 요청 로그 - `/var/log/ecs/audit.log`

출력 수준, 형식 지정 및 추가 구성 옵션에 대한 자세한 내용은 Amazon ECS 설명서의 Amazon ECS 로그 파일 위치를 참조하세요.

> **Important**
>
> EC2 컨테이너 인스턴스를 실행하거나 관리하지 않으므로 Fargate 시작 유형에는 에이전트 설치 또는 구성이 필요하지 않습니다.

Amazon ECS 컨테이너 인스턴스는 최신 Amazon ECS 최적화 AMIs 및 컨테이너 에이전트를 사용해야 합니다.는 AMI ID를 포함한 Amazon ECS 최적화 AMI 정보와 함께 퍼블릭 Systems Manager 파라미터 스토어 파라미터를 AWS 저장합니다. Amazon ECS 최적화 AMI에 대한 파라미터 스토어 파라미터 형식을 사용하여 파라미터 스토어에서 가장 최근에 최적화된 AMIs. 템플릿에서 최신 AMI 또는 특정 AMI 릴리스를 참조하는 퍼블릭 파라미터 스토어 파라미터를 참조할 수 있습니다 CloudFormation .

AWS 는 지원되는 각 리전에서 동일한 파라미터 스토어 파라미터를 제공합니다. 즉, 이러한 파라미터를 참조하는 CloudFormation 템플릿은 AMI를 업데이트하지 않고도 리전 및 계정 간에 재사용할 수 있습니다. 특정 릴리스를 참조하여 조직에 대한 최신 Amazon ECS AMIs의 배포를 제어할 수 있습니다. 이를 통해 테스트할 때까지 새 Amazon ECS 최적화 AMI의 사용을 방지할 수 있습니다.

EC2 및 Fargate 시작 유형에 대한 Amazon ECS 컨테이너 로그

Amazon ECS는 작업 정의를 사용하여 컨테이너를 작업 및 서비스로 배포하고 관리합니다. 작업 정의 내에서 Amazon ECS 클러스터로 시작하려는 컨테이너를 구성합니다. 로깅은 컨테이너 수준에서 로그 드라이버로 구성됩니다. 여러 로그 드라이버 옵션은 EC2 또는 Fargate 시작 유형을 사용하는지 여부에 따라 컨테이너에 다양한 로깅 시스템(예: ,,`syslog`,,`awslogs`fluentdgelfjson-filejournaldlogentriessplunk, 또는 `awsfirelens`)을 제공합니다. Fargate 시작 유형은 `awslogs`, `splunk`및 로그 드라이버 옵션의 하위 집합을 제공합니다`awsfirelens`.는 컨테이너 출력을 캡처하여 CloudWatch Logs로 전송할 수 있도록 `awslogs` 로그 드라이버를 AWS 제공합니다. 로그 드라이버 설정을 사용하면 다른 여러 옵션과 함께 로그 그룹, 리전 및 로그 스트림 접두사를 사용자 지정할 수 있습니다.

로그 그룹의 기본 이름 지정과의 CloudWatch Logs 자동 구성 옵션에서 AWS Management Console 사용하는 옵션은 입니다`/ecs/<task_name>`. Amazon ECS에서 사용하는 로그 스트림 이름의 `<awslogs-stream-prefix>/<container_name>/<task_id>` 형식은 다음과 같습니다. 조직의 요구 사항에 따라 로그를 그룹화하는 그룹 이름을 사용하는 것이 좋습니다. 다음 표에서 `image_name` 및 `image_tag`는 로그 스트림의 이름에 포함됩니다.

로그 그룹 이름	`/<Business unit>/<Project or application name>/<Environment>/<Cluster name>/<Task name>`
로그 스트림 이름 접두사	`/<image_name>/<image_tag>`

이 정보는 작업 정의에서도 사용할 수 있습니다. 그러나 작업은 새 개정으로 정기적으로 업데이트됩니다. 즉, 작업 정의에서 현재 사용 중인 `image_tag` 작업 정의`image_name`와 다른를 사용했을 수 있습니다. 자세한 내용과 이름 지정 제안은이 가이드의 CloudWatch 배포 계획 섹션을 참조하세요.

지속적 통합 및 지속적 전달(CI/CD) 파이프라인 또는 자동화된 프로세스를 사용하는 경우 각 새 Docker 이미지 빌드를 사용하여 애플리케이션에 대한 새 작업 정의 개정을 생성할 수 있습니다. 예를 들어 CI/CD 프로세스의 일부로 작업 정의 개정 및 로깅 구성에 Docker 이미지 이름, 이미지 태그, GitHub 개정 또는 기타 중요한 정보를 포함할 수 있습니다.

FireLens for Amazon ECS에서 사용자 지정 로그 라우팅 사용

Amazon ECS용 FireLens를 사용하면 로그를 Fluentd 또는 Fluent Bit로 라우팅할 수 있으므로 컨테이너 로그를 AWS 서비스 및 AWS 파트너 네트워크(APN) 대상으로 직접 전송하고 CloudWatch Logs로 로그 전송을 지원할 수 있습니다.

AWS 는 Amazon Kinesis Data Streams, Amazon Data Firehose 및 CloudWatch Logs용 플러그인이 사전 설치된 Fluent Bit용 Docker 이미지를 제공합니다. CloudWatch Logs로 전송된 로그를 더 사용자 지정하고 제어하려면 `awslogs` 로그 드라이버 대신 FireLens 로그 드라이버를 사용할 수 있습니다.

예를 들어 FireLens 로그 드라이버를 사용하여 로그 형식 출력을 제어할 수 있습니다. 즉, Amazon ECS 컨테이너의 CloudWatch 로그는 자동으로 JSON 객체로 형식이 지정되고 `ecs_cluster`, , `ecs_task_arn`, `container_name`, `container_idecs_task_definition`에 대한 JSON 형식의 속성이 포함됩니다`ec2_instance_id`. `awsfirelens` 드라이버를 지정하면 유창한 호스트가 `FLUENT_HOST` 및 `FLUENT_PORT` 환경 변수를 통해 컨테이너에 노출됩니다. 즉, 유창한 로거 라이브러리를 사용하여 코드에서 로그 라우터에 직접 로그인할 수 있습니다. 예를 들어 애플리케이션은 환경 변수에서 사용할 수 있는 값을 사용하여 Fluent Bit에 로그인하는 `fluent-logger-python` 라이브러리를 포함할 수 있습니다.

Amazon ECS에 FireLens를 사용하도록 선택한 경우 `awslogs` 로그 드라이버와 동일한 설정을 구성하고 다른 설정도 사용할 수 있습니다. 예를 들어, FireLens를 사용하여 CloudWatch에 로깅하도록 구성된 NGINX 서버를 시작하는 ecs-task-nginx-firelense.json Amazon ECS 태스크 정의를 사용할 수 있습니다. 또한 FireLens Fluent Bit 컨테이너를 로깅용 사이드카로 시작합니다.

Amazon ECS에 대한 지표

Amazon ECS는 Amazon ECS 컨테이너 에이전트를 사용하여 클러스터 및 서비스 수준에서 EC2 및 Fargate 시작 유형에 대한 표준 CloudWatch 지표(예: CPU 및 메모리 사용률)를 제공합니다. EC2 CloudWatch Container Insights를 사용하여 서비스, 작업 및 컨테이너에 대한 지표를 캡처하거나 임베디드 지표 형식을 사용하여 사용자 지정 컨테이너 지표를 캡처할 수도 있습니다.

Container Insights는 클러스터, 컨테이너 인스턴스, 서비스 및 작업 수준의 CPU 사용률, 메모리 사용률, 네트워크 트래픽 및 스토리지와 같은 지표를 제공하는 CloudWatch 기능입니다. 또한 Container Insights는 서비스 및 작업을 분석하고 컨테이너 수준에서 평균 메모리 또는 CPU 사용률을 확인하는 데 도움이 되는 자동 대시보드를 생성합니다. Container Insights는 그래프, 경보 및 대시보드에 사용할 수 있는 사용자 지정 네임스페이스에 `ECS/ContainerInsights` 사용자 지정 지표를 게시합니다.
https://docs.aws.amazon.com//AmazonECS/latest/developerguide/cloudwatch-metrics.html

각 개별 Amazon ECS 클러스터에 대해 Container Insights를 활성화하여 Container Insight 지표를 켤 수 있습니다. 컨테이너 인스턴스 수준에서 지표도 보려면 Amazon ECS 클러스터에서 CloudWatch 에이전트를 데몬 컨테이너로 시작할 수 있습니다. cwagent-ecs-instance-metric-cfn.yaml CloudFormation 템플릿을 사용하여 CloudWatch 에이전트를 Amazon ECS 서비스로 배포할 수 있습니다. 중요한 것은이 예제에서는 적절한 사용자 지정 CloudWatch 에이전트 구성을 생성하여 키와 함께 Parameter Store에 저장했다고 가정합니다`ecs-cwagent-daemon-service`.

CloudWatch Container Insights용 데몬 컨테이너로 배포된 CloudWatch 에이전트에는 `instance_cpu_reserved_capacity` 및 , `ClusterName`, `ContainerInstanceIdInstanceId`차원`instance_memory_reserved_capacity`과 같은 추가 디스크, 메모리 및 CPU 지표가 포함됩니다. CloudWatch 컨테이너 인스턴스 수준의 지표는 CloudWatch 임베디드 지표 형식을 사용하여 Container Insights에서 구현됩니다. 이 가이드의 CloudWatch 에이전트 배포 및 구성을 위한 State Manager 및 Distributor 설정 섹션에 있는 접근 방식을 사용하여 Amazon ECS 컨테이너 인스턴스에 대한 추가 시스템 수준 지표를 구성할 수 있습니다.

Amazon ECS에서 사용자 지정 애플리케이션 지표 생성

CloudWatch 임베디드 지표 형식을 사용하여 애플리케이션에 대한 사용자 지정 지표를 생성할 수 있습니다. `awslogs` 로그 드라이버는 CloudWatch 임베디드 지표 형식 문을 해석할 수 있습니다.

다음 예제의 `CW_CONFIG_CONTENT` 환경 변수는 `cwagentconfig` Systems Manager Parameter Store 파라미터의 내용으로 설정됩니다. 이 기본 구성으로 에이전트를 실행하여 임베디드 지표 형식 엔드포인트로 구성할 수 있습니다. 그러나 더 이상 필요하지 않습니다.

```
  {
  "logs": {
    "metrics_collected": {
      "emf": { }
    }
  }
}
```

여러 계정 및 리전에 Amazon ECS 배포가 있는 경우 보안 AWS Secrets Manager 암호를 사용하여 CloudWatch 구성을 저장하고 보안 암호 정책을 구성하여 조직과 공유할 수 있습니다. 작업 정의의 보안 암호 옵션을 사용하여 `CW_CONFIG_CONTENT` 변수를 설정할 수 있습니다.

애플리케이션에서 AWS 제공된 오픈 소스 임베디드 지표 형식 라이브러리를 사용하고 `AWS_EMF_AGENT_ENDPOINT` 환경 변수를 지정하여 임베디드 지표 형식 엔드포인트 역할을 하는

CloudWatch 에이전트 사이드카 컨테이너에 연결할 수 있습니다. 예를 들어 ecs_cw_emf_example 샘플 Python 애플리케이션을 사용하여 임베디드 지표 형식으로 지표를 임베디드 지표 형식 엔드포인트로 구성된 CloudWatch 에이전트 사이드카 컨테이너로 전송할 수 있습니다.

CloudWatch용 Fluent Bit 플러그인을 사용하여 임베디드 지표 형식 메시지를 보낼 수도 있습니다. 또한 ecs_firelense_emf_example 샘플 Python 애플리케이션을 사용하여 임베디드 지표 형식의 지표를 Firelens for Amazon ECS 사이드카 컨테이너로 전송할 수 있습니다.

임베디드 지표 형식을 사용하지 않으려면 AWS API 또는 AWS SDK를 통해 CloudWatch 지표를 생성하고 업데이트할 수 있습니다. 특정 사용 사례가 없으면 코드에 유지 관리 오버헤드가 추가되므로이 접근 방식을 사용하지 않는 것이 좋습니다.

Amazon EKS의 로깅 및 모니터링

Amazon Elastic Kubernetes Service(Amazon EKS)는 Kubernetes 컨트롤 플레인에 대한 CloudWatch Logs와 통합됩니다. 컨트롤 플레인은 Amazon EKS에서 관리형 서비스로 제공하며 CloudWatch 에이전트를 설치하지 않고도 로깅을 켤 수 있습니다. CloudWatch 에이전트를 배포하여 Amazon EKS 노드 및 컨테이너 로그를 캡처할 수도 있습니다. Fluent Bit 및 Fluentd는 컨테이너 로그를 CloudWatch Logs로 전송하는 데도 지원됩니다.

CloudWatch Container Insights는 클러스터, 노드, 포드, 작업 및 서비스 수준에서 Amazon EKS에 대한 포괄적인 지표 모니터링 솔루션을 제공합니다. Amazon EKS는 Prometheus를 사용한 지표 캡처를 위한 여러 옵션도 지원합니다. Amazon EKS 컨트롤 플레인은 Prometheus 형식으로 지표를 노출하는 지표 엔드포인트를 제공합니다. Amazon EKS 클러스터에 Prometheus를 배포하여 이러한 지표를 사용할 수 있습니다.

다른 Prometheus 엔드포인트를 사용하는 것 외에도 Prometheus 지표를 스크레이프하고 CloudWatch 지표를 생성하도록 CloudWatch 에이전트를 설정할 수도 있습니다. CloudWatch 또한 Prometheus에 대한 Container Insights 모니터링은 지원되는 컨테이너화된 워크로드 및 시스템에서 Prometheus 지표를 자동으로 검색하고 캡처할 수 있습니다.

Amazon EC2 with Distributor 및 State Manager에 사용되는 접근 방식과 유사한 방식으로 Amazon EKS 노드에 CloudWatch 에이전트를 설치하고 구성하여 Amazon EKS 노드를 표준 시스템 로깅 및 모니터링 구성에 맞게 조정할 수 있습니다.

Amazon EKS에 대한 로깅

Kubernetes 로깅은 컨트롤 플레인 로깅, 노드 로깅 및 애플리케이션 로깅으로 나눌 수 있습니다. Kubernetes 컨트롤 플레인은 Kubernetes 클러스터를 관리하고 감사 및 진단 목적으로 사용되는 로그를 생성하는 구성 요소 세트입니다. Amazon EKS를 사용하면 다양한 컨트롤 플레인 구성 요소에 대한 로그를 켜고 CloudWatch로 전송할 수 있습니다.

또한 Kubernetes는 포드를 실행하는 각 Kubernetes 노드`kube-proxy`에서 `kubelet` 및와 같은 시스템 구성 요소를 실행합니다. 이러한 구성 요소는 각 노드 내에 로그를 작성하며 각 Amazon EKS 노드에 대해 이러한 로그를 캡처하도록 CloudWatch 및 Container Insights를 구성할 수 있습니다.

컨테이너는 Kubernetes 클러스터 내에서 포드로 그룹화되며 Kubernetes 노드에서 실행되도록 예약됩니다. 대부분의 컨테이너화된 애플리케이션은 표준 출력 및 표준 오류에 쓰고 컨테이너 엔진은 출력을 로깅 드라이버로 리디렉션합니다. Kubernetes에서 컨테이너 로그는 노드의 `/var/log/pods` 디렉

터리에 있습니다. 각 Amazon EKS 포드에 대해 이러한 로그를 캡처하도록 CloudWatch 및 Container Insights를 구성할 수 있습니다.

Amazon EKS 컨트롤 플레인 로깅

Amazon EKS 클러스터는 Kubernetes 클러스터와 컨테이너를 실행하는 Amazon EKS 노드를 위한 고가용성 단일 테넌트 컨트롤 플레인으로 구성됩니다. 컨트롤 플레인 노드는에서 관리하는 계정에서 실행됩니다 AWS. Amazon EKS 클러스터 컨트롤 플레인 노드는 CloudWatch와 통합되며 특정 컨트롤 플레인 구성 요소에 대한 로깅을 켤 수 있습니다.

로그는 각 Kubernetes 컨트롤 플레인 구성 요소 인스턴스에 대해 제공됩니다.는 컨트롤 플레인 노드의 상태를 AWS 관리하고 Kubernetes 엔드포인트에 대한 서비스 수준 계약(SLA)을 제공합니다.

Amazon EKS 노드 및 애플리케이션 로깅

CloudWatch Container Insights를 사용하여 Amazon EKS에 대한 로그 및 지표를 캡처하는 것이 좋습니다. Container Insights는 CloudWatch 에이전트와 CloudWatch에 대한 로그 캡처를 위한 Fluent Bit 또는 Fluentd를 사용하여 클러스터, 노드 및 포드 수준 지표를 구현합니다 CloudWatch. Container Insights는 캡처된 CloudWatch 지표의 계층화된 보기와 함께 자동 대시보드도 제공합니다. Container Insights는 모든 Amazon EKS 노드에서 실행되는 CloudWatch DaemonSet 및 Fluent Bit DaemonSet로 배포됩니다. Fargate 노드는에서 관리 AWS 하며 DaemonSets를 지원하지 않기 때문에 Container Insights에서 지원되지 않습니다. Amazon EKS에 대한 Fargate 로깅은이 안내서에서 별도로 다룹니다.

다음 표에는 Amazon EKS에 대한 기본 Fluentd 또는 Fluent Bit 로그 캡처 구성으로 캡처된 CloudWatch 로그 그룹 및 로그가 나와 있습니다. https://docs.aws.amazon.com//AmazonCloudWatch/latest/monitoring/Container-Insights-setup-logs-FluentBit.html

`/aws/containerinsights/Cluster_Name/application`	의 모든 로그 파일`/var/log/containers` . 이 디렉터리는 `/var/log/pods` 디렉터리 구조의 모든 Kubernetes 컨테이너 로그에 대한 심볼릭 링크를 제공합니다. 이렇게 하면 애플리케이션 컨테이너 로그가 `stdout` 또는 에 기록됩니다`stderr`. 또한 `aws-vpc-cni-init` , `kube-proxy` 및와 같은 Kubernetes 시스템 컨테이너에 대한 로그도 포함됩니다`coreDNS`.

/aws/containerinsights/Cluster_Name/host	/var/log/dmesg , /var/log/secure 및에서 로그합니다/var/log/messages .
/aws/containerinsights/Cluster_Name/dataplane	kubelet.service , kubeproxy.service 및 docker.service 에 대한 /var/log/journal 에서의 로그.

로깅에 Container Insights를 Fluent Bit 또는 Fluentd와 함께 사용하지 않으려면 Amazon EKS 노드에 설치된 CloudWatch 에이전트를 사용하여 노드 및 컨테이너 로그를 캡처할 수 있습니다. Amazon EKS 노드는 EC2 인스턴스이므로 Amazon EC2의 표준 시스템 수준 로깅 접근 방식에 포함해야 합니다. Distributor 및 State Manager를 사용하여 CloudWatch 에이전트를 설치하는 경우 Amazon EKS 노드도 CloudWatch 에이전트 설치, 구성 및 업데이트에 포함됩니다.

다음 표에는 Kubernetes와 관련된 로그와 로깅에 Container Insights를 Fluent Bit 또는 Fluentd와 함께 사용하지 않는 경우 캡처해야 하는 로그가 나와 있습니다.

/var/log/containers	이 디렉터리는 /var/log/pods 디렉터리 구조 아래의 모든 Kubernetes 컨테이너 로그에 대한 심볼 링크를 제공합니다. 이렇게 하면 stdout 또는에 쓰는 애플리케이션 컨테이너 로그가 효과적으로 캡처됩니다stderr. 여기에는 aws-vpc-cni-init , kube-proxy 및와 같은 Kubernetes 시스템 컨테이너에 대한 로그가 포함됩니다coreDNS. 중요: Container Insights를 사용하는 경우에는 필요하지 않습니다.
var/log/aws-routed-eni/ipamd.log /var/log/aws-routed-eni/plugin.log	L-IPAM 데몬에 대한 로그는 여기에서 찾을 수 있습니다.

Amazon EKS 노드가 적절한 시스템 수준 로그 및 지표를 전송하도록 CloudWatch 에이전트를 설치하고 구성해야 합니다. 그러나 Amazon EKS 최적화 AMI에는 Systems Manager 에이전트가 포함되지 않습니다. 시작 템플릿을 사용하면 Systems Manager 에이전트 설치와 사용자 데이터 섹션을 통해 구현된 시작 스크립트를 사용하여 중요한 Amazon EKS별 로그를 캡처하는 기본 CloudWatch 구성을 자

동화할 수 있습니다. Amazon EKS 노드는 Auto Scaling 그룹을 관리형 노드 그룹 또는 자체 관리형 노드로 사용하여 배포됩니다.

관리형 노드 그룹을 사용하면 사용자 데이터 섹션이 포함된 시작 템플릿을 제공하여 Systems Manager 에이전트 설치 및 CloudWatch 구성을 자동화할 수 있습니다. amazon_eks_managed_node_group_launch_config.yaml CloudFormation 템플릿을 사용자 지정하고 사용하여 Systems Manager 에이전트, CloudWatch 에이전트를 설치하고 CloudWatch 구성 디렉터리에 Amazon EKS별 로깅 구성을 추가하는 시작 템플릿을 생성할 수 있습니다. 이 템플릿을 사용하여 infrastructure-as-code(IaC) 접근 방식으로 Amazon EKS 관리형 노드 그룹 시작 템플릿을 업데이트할 수 있습니다. CloudFormation 템플릿을 업데이트할 때마다 시작 템플릿의 새 버전이 프로비저닝됩니다. 그런 다음 새 템플릿 버전을 사용하도록 노드 그룹을 업데이트하고 관리형 수명 주기 프로세스가 가동 중지 없이 노드를 업데이트하도록 할 수 있습니다. 관리형 노드 그룹에 적용된 IAM 역할 및 인스턴스 프로파일에 `CloudWatchAgentServerPolicy` 및 `AmazonSSMManagedInstanceCore` AWS 관리형 정책이 포함되어 있는지 확인합니다.

자체 관리형 노드를 사용하면 Amazon EKS 노드의 수명 주기 및 업데이트 전략을 직접 프로비저닝하고 관리할 수 있습니다. 자체 관리형 노드를 사용하면 다른 옵션과 함께 Amazon EKS 클러스터 및 Bottlerocket에서 Windows 노드를 실행할 수 있습니다. CloudFormation 를 사용하여 자체 관리형 노드를 Amazon EKS 클러스터에 배포할 수 있습니다. 즉, Amazon EKS 클러스터에 IaC 및 관리형 변경 접근 방식을 사용할 수 있습니다. AWS 는 있는 그대로 사용하거나 사용자 지정할 수 있는 amazon-eks-nodegroup.yaml CloudFormation 템플릿을 제공합니다. 템플릿은 클러스터의 Amazon EKS 노드에 필요한 모든 리소스(예: 별도의 IAM 역할, 보안 그룹, Amazon EC2 Auto Scaling 그룹 및 시작 템플릿)를 프로비저닝합니다. amazon-eks-nodegroup.yaml CloudFormation 템플릿은 필요한 Systems Manager 에이전트인 CloudWatch 에이전트를 설치하고 CloudWatch 구성 디렉터리에 Amazon EKS별 로깅 구성을 추가하는 업데이트된 버전입니다.

Fargate의 Amazon EKS에 대한 로깅

Fargate의 Amazon EKS를 사용하면 Kubernetes 노드를 할당하거나 관리하지 않고도 포드를 배포할 수 있습니다. 이렇게 하면 Kubernetes 노드에 대한 시스템 수준 로그를 캡처할 필요가 없습니다. Fargate 포드에서 로그를 캡처하려면 Fluent Bit를 사용하여 로그를 CloudWatch에 직접 전달할 수 있습니다. 이를 통해 추가 구성이나 Fargate의 Amazon EKS 포드용 사이드카 컨테이너 없이 CloudWatch로 로그를 자동으로 라우팅할 수 있습니다. 이에 대한 자세한 내용은 Amazon EKS 설명서의 Fargate 로깅 및 AWS 블로그의 Amazon EKS용 Fluent Bit를 참조하세요. 이 솔루션은 컨테이너에서 `STDOUT` 및 `STDERR` 입출력(I/O) 스트림을 캡처하여 Fargate의 Amazon EKS 클러스터에 대해 설정된 Fluent Bit 구성을 기반으로 Fluent Bit를 통해 CloudWatch로 전송합니다.

Amazon EKS 및 Kubernetes에 대한 지표

Kubernetes는 리소스 사용량 지표(예: 노드 및 포드의 CPU 및 메모리 사용량)에 액세스할 수 있는 지표 API를 제공하지만 API는 과거 지표가 아닌 point-in-time 정보만 제공합니다. Kubernetes 지표 서버는 일반적으로 Amazon EKS 및 Kubernetes 배포에 사용되어 지표를 집계하고, 지표에 대한 단기 기록 정보를 제공하고, Horizontal Pod Autoscaler와 같은 기능을 지원합니다.

Amazon EKS는 Kubernetes API 서버를 통해 Prometheus 형식으로 컨트롤 플레인 지표를 노출하며 CloudWatch는 이러한 지표를 캡처하고 수집할 수 있습니다. CloudWatch 및 Container Insights는 Amazon EKS 노드 및 포드에 대한 포괄적인 지표 캡처, 분석 및 경보를 제공하도록 구성할 수도 있습니다.

Kubernetes 컨트롤 플레인 지표

Kubernetes는 `/metrics` HTTP API 엔드포인트를 사용하여 Prometheus 형식의 컨트롤 플레인 지표를 노출합니다. 웹 브라우저를 사용하여 이러한 지표를 그래프로 표시하고 보려면 Kubernetes 클러스터에 Prometheus를 설치해야 합니다. Kubernetes API 서버에서 노출한 지표를 CloudWatch로 수집할 수도 있습니다. CloudWatch

Kubernetes에 대한 노드 및 시스템 지표

Kubernetes는 클러스터, 노드, 포드 수준 CPU 및 메모리 통계를 위해 Kubernetes 클러스터에 배포하고 실행할 수 있는 Prometheus 지표 서버 포드를 제공합니다. 이러한 지표는 Horizontal Pod Autoscaler 및 Vertical Pod Autoscaler와 함께 사용됩니다. CloudWatch는 이러한 지표를 제공할 수도 있습니다.

Kubernetes 대시보드 또는 수평 및 수직 포드 오토스케일러를 사용하는 경우 Kubernetes 지표 서버를 설치해야 합니다. Kubernetes 대시보드를 사용하면 Kubernetes 클러스터, 노드, 포드 및 관련 구성을 검색 및 구성하고 Kubernetes 지표 서버에서 CPU 및 메모리 지표를 볼 수 있습니다.

Kubernetes Metrics Server에서 제공하는 지표는 비자동 조정(예: 모니터링)을 위해 사용할 수 없습니다. 지표는 과거 분석이 아닌 point-in-time 분석을 위한 것입니다. Kubernetes 대시보드는를 배포`dashboard-metrics-scraper`하여 Kubernetes 지표 서버의 지표를 짧은 기간 동안 저장합니다.

Container Insights는 Kubernetes DaemonSet에서 실행되는 컨테이너화된 버전의 CloudWatch 에이전트를 사용하여 클러스터에서 실행 중인 모든 컨테이너를 검색하고 노드 수준 지표를 제공합니다. 성능 스택의 모든 계층에서 성능 데이터를 수집합니다. 퀵 스타트에서 AWS 퀵 스타트를 사용하거나 Container Insights를 별도로 구성할 수 있습니다. 퀵 스타트는 CloudWatch 에이전트를 사용한 지표 모니터링과 Fluent Bit를 사용한 로깅을 설정하므로 로깅 및 모니터링을 위해 한 번만 배포하면 됩니다.

Amazon EKS 노드는 EC2 인스턴스이므로 Amazon EC2에 대해 정의한 표준을 사용하여 Container Insights에서 캡처한 지표 외에도 시스템 수준 지표를 캡처해야 합니다. 이 가이드의 CloudWatch 에이전트 배포 및 구성을 위한 State Manager 및 Distributor 설정 섹션에서 동일한 접근 방식을 사용하여 Amazon EKS 클러스터에 대한 CloudWatch 에이전트를 설치하고 구성할 수 있습니다. 지표와 Amazon EKS별 로그 구성을 포함하도록 Amazon EKS별 CloudWatch 구성 파일을 업데이트할 수 있습니다.

Prometheus가 지원되는 CloudWatch 에이전트는 지원되는 컨테이너화된 워크로드 및 시스템에서 Prometheus 지표를 자동으로 검색하고 스크레이프할 수 있습니다. CloudWatch Logs Insights를 사용하여 분석하기 위해 임베디드 지표 형식의 CloudWatch 로그로 수집하고 CloudWatch 지표를 자동으로 생성합니다.

> **Important**
>
> Prometheus 지표를 수집하려면 특수 버전의 CloudWatch 에이전트를 배포해야 합니다. CloudWatch 이는 Container Insights용으로 배포된 CloudWatch 에이전트와는 별도의 에이전트입니다. CloudWatch 에이전트 및 Amazon EKS 포드 배포에 대한 배포 및 구성 파일이 포함된 prometheus_jmx 샘플 Java 애플리케이션을 사용하여 Prometheus 지표 검색을 시연할 수 있습니다. 자세한 내용은 CloudWatch 설명서의 Amazon EKS 및 Kubernetes에서 Java/JMX 샘플 워크로드 설정을 참조하세요. Amazon EKS 클러스터에서 실행되는 다른 Prometheus 대상의 지표를 캡처하도록 CloudWatch 에이전트를 구성할 수도 있습니다.

애플리케이션 지표

CloudWatch 임베디드 지표 형식을 사용하여 사용자 지정 지표를 생성할 수 있습니다. 임베디드 지표 형식 문을 수집하려면 임베디드 지표 형식 항목을 임베디드 지표 형식 엔드포인트로 전송해야 합니다. CloudWatch 에이전트는 Amazon EKS 포드에서 사이드카 컨테이너로 구성할 수 있습니다. CloudWatch 에이전트 구성은 Kubernetes ConfigMap으로 저장되며 CloudWatch 에이전트 사이드카 컨테이너에서 읽어 임베디드 지표 형식 엔드포인트를 시작합니다.

또한 애플리케이션을 Prometheus 대상으로 설정하고 Prometheus 지원을 통해 CloudWatch 에이전트를 구성하여 지표를 검색, 스크레이프 및 CloudWatch로 수집할 수 있습니다. 예를 들어 오픈 소스 JMX Exporter를 Java 애플리케이션과 함께 사용하여 CloudWatch 에이전트의 Prometheus 소비를 위해 JMX Beans를 노출할 수 있습니다.

임베디드 지표 형식을 사용하지 않으려면 AWS API 또는 AWS SDK를 사용하여 CloudWatch 지표를 생성하고 업데이트할 수도 있습니다. 그러나 모니터링과 애플리케이션 로직을 혼합하므로이 접근 방식은 권장하지 않습니다.

Fargate의 Amazon EKS 지표

Fargate는 Kubernetes 포드를 실행하도록 Amazon EKS 노드를 자동으로 프로비저닝하므로 노드 수준 지표를 모니터링하고 수집할 필요가 없습니다. 그러나 Fargate의 Amazon EKS 노드에서 실행되는 포드에 대한 지표를 모니터링해야 합니다. Container Insights는 현재 지원되지 않는 다음 기능이 필요하므로 Fargate의 Amazon EKS에서는 현재 사용할 수 없습니다.

- DaemonSets는 현재 지원되지 않습니다. Container Insights는 각 클러스터 노드에서 CloudWatch 에이전트를 DaemonSet로 실행하여 배포됩니다.
- HostPath 영구 볼륨은 지원되지 않습니다. CloudWatch 에이전트 컨테이너는 컨테이너 지표 데이터를 수집하기 위한 사전 조건으로 hostPath 영구 볼륨을 사용합니다.
- Fargate는 권한이 있는 컨테이너와 호스트 정보에 대한 액세스를 방지합니다.

Fargate용 내장 로그 라우터를 사용하여 임베디드 지표 형식 문을 CloudWatch로 전송할 수 있습니다. 로그 라우터는 임베디드 지표 형식 문을 지원하도록 구성할 수 있는 CloudWatch 플러그인이 있는 Fluent Bit를 사용합니다.

Amazon EKS 클러스터에 Prometheus 서버를 배포하여 Fargate 노드에서 지표를 수집하여 Fargate 노드에 대한 포드 수준 지표를 검색하고 캡처할 수 있습니다. Prometheus에는 영구 스토리지가 필요하므로 영구 스토리지에 Amazon Elastic File System(Amazon EFS)을 사용하는 경우 Fargate에 Prometheus를 배포할 수 있습니다. Amazon EC2 지원 노드에 Prometheus를 배포할 수도 있습니다. 자세한 내용은 블로그의 Monitoring Amazon EKS on AWS Fargate using Prometheus and Grafana를 AWS 참조하세요.

Amazon EKS에서 Prometheus 모니터링

Amazon Managed Service for Prometheus는 오픈 소스 Prometheus를 위한 확장 가능하고 안전한 AWS 관리형 서비스를 제공합니다. Prometheus 쿼리 언어(PromQL)를 사용하면 운영 지표를 수집, 저장 및 쿼리하기 위한 기본 인프라를 관리하지 않고도 컨테이너화된 워크로드의 성능을 모니터링할 수 있습니다. AWS Distro for OpenTelemetry(ADOT) 또는 Prometheus 서버를 수집 에이전트로 사용하여 Amazon EKS 및 Amazon ECS에서 Prometheus 지표를 수집할 수 있습니다.

Prometheus에 대한 CloudWatch Container Insights 모니터링을 사용하면 CloudWatch 에이전트를 구성하고 사용하여 Amazon ECS, Amazon EKS 및 Kubernetes 워크로드에서 Prometheus 지표를 검색하고 이를 CloudWatch 지표로 수집할 수 있습니다. 이 솔루션은 CloudWatch가 기본 관찰성 및 모니터링 솔루션인 경우에 적합합니다. 그러나 다음 목록은 Amazon Managed Service for Prometheus가 Prometheus 지표를 수집, 저장 및 쿼리하는 데 더 많은 유연성을 제공하는 사용 사례를 간략하게 설명합니다.

- Amazon Managed Service for Prometheus를 사용하면 Amazon EKS 또는 자체 관리형 Kubernetes에 배포된 기존 Prometheus 서버를 사용하고 로컬로 구성된 데이터 스토어 대신 Amazon Managed Service for Prometheus에 쓰도록 구성할 수 있습니다. 이렇게 하면 Prometheus 서버 및 해당 인프라를 위한 고가용성 데이터 스토어를 관리하기 위한 차별화되지 않은 과도한 부담이 제거됩니다. Amazon Managed Service for Prometheus는 AWS 클라우드에서 활용하려는 성숙한 Prometheus 배포가 있는 경우에 적합한 선택입니다.
- Grafana는 Prometheus를 시각화를 위한 데이터 소스로 직접 지원합니다. 컨테이너 모니터링에 CloudWatch Dashboards 대신 Grafana를 Prometheus와 함께 사용하려는 경우 Amazon Managed Service for Prometheus가 요구 사항을 충족할 수 있습니다. Amazon Managed Service for Prometheus는 Amazon Managed Grafana와 통합되어 관리형 오픈 소스 모니터링 및 시각화 솔루션을 제공합니다.
- Prometheus를 사용하면 PromQL 쿼리를 사용하여 운영 지표에 대한 분석을 수행할 수 있습니다. 반대로 CloudWatch 에이전트는 임베디드 지표 형식의 Prometheus 지표를 CloudWatch Logs로 수집하여 CloudWatch 지표를 생성합니다. CloudWatch CloudWatch Logs Insights를 사용하여 임베디드 지표 형식 로그를 쿼리할 수 있습니다.
- 모니터링 및 지표 캡처에 CloudWatch를 사용할 계획이 없는 경우 Prometheus 서버 및 Grafana와 같은 시각화 솔루션과 함께 Amazon Managed Service for Prometheus를 사용해야 합니다. Prometheus 대상에서 지표를 스크레이프하도록 Prometheus 서버를 구성하고 Amazon Managed Service for Prometheus 워크스페이스에 원격으로 쓰도록 서버를 구성해야 합니다. Amazon Managed Grafana를 사용하는 경우 포함된 플러그인을 사용하여 Amazon Managed Grafana를 Amazon Managed Service for Prometheus 데이터 소스와 직접 통합할 수 있습니다. 지표 데이터

는 Amazon Managed Service for Prometheus에 저장되므로 CloudWatch 에이전트를 배포하기 위한 종속성이나 CloudWatch에 데이터를 수집하기 위한 요구 사항은 없습니다. Prometheus에 대한 Container Insights 모니터링에는 CloudWatch 에이전트가 필요합니다.

ADOT Collector를 사용하여 Prometheus 계측 애플리케이션에서 스크레이프하고 지표를 Amazon Managed Service for Prometheus로 전송할 수도 있습니다. ADOT Collector에 대한 자세한 내용은 AWS Distro for OpenTelemetry 설명서를 참조하세요.

에 대한 로깅 및 지표 AWS Lambda

Lambda를 사용하면 워크로드에 대한 서버를 관리하고 모니터링할 필요가 없으며 애플리케이션 코드를 추가로 구성하거나 계측하지 않고도 CloudWatch 지표 및 CloudWatch Logs에서 자동으로 작동합니다. 이 섹션에서는 Lambda에서 사용하는 시스템의 성능 특성과 구성 선택이 성능에 미치는 영향을 이해하는 데 도움이 됩니다. 또한 성능 최적화 및 애플리케이션 수준 문제 진단을 위해 Lambda 함수를 로깅하고 모니터링하는 데 도움이 됩니다.

Lambda 함수 로깅

Lambda는 로깅 드라이버 없이 Lambda 함수의 표준 출력 및 표준 오류 메시지를 CloudWatch Logs로 자동으로 스트리밍합니다. 또한 Lambda는 Lambda 함수를 실행하는 컨테이너를 자동으로 프로비저닝하고 별도의 로그 스트림에서 로그 메시지를 출력하도록 구성합니다.

Lambda 함수의 후속 호출은 동일한 컨테이너와 출력을 동일한 로그 스트림에 재사용할 수 있습니다. Lambda는 새 컨테이너를 프로비저닝하고 호출을 새 로그 스트림에 출력할 수도 있습니다.

Lambda 함수가 처음 호출되면 Lambda가 로그 그룹을 자동으로 생성합니다. Lambda 함수에는 여러 버전이 있을 수 있으며 실행할 버전을 선택할 수 있습니다. Lambda 함수의 호출에 대한 모든 로그는 동일한 로그 그룹에 저장됩니다. 이름은 변경할 수 없으며 `/aws/lambda/<YourLambdaFunctionName>` 형식입니다. 각 Lambda 함수 인스턴스의 로그 그룹에 별도의 로그 스트림이 생성됩니다. Lambda에는 `YYYY/MM/DD/[<FunctionVersion>]<InstanceId>` 형식을 사용하는 로그 스트림에 대한 표준 이름 지정 규칙이 있습니다. `InstanceId`는 Lambda 함수 인스턴스를 식별하기 AWS 위해에서 생성됩니다.

CloudWatch Logs Insights를 사용하여 로그 메시지를 더 쉽게 쿼리할 수 있으므로 로그 메시지를 JSON 형식으로 포맷하는 것이 좋습니다. 또한 더 쉽게 필터링하고 내보낼 수 있습니다. 로깅 라이브러리를 사용하여이 프로세스를 단순화하거나 자체 로그 처리 함수를 작성할 수 있습니다. 로깅 라이브러리를 사용하여 로그 메시지의 형식을 지정하고 분류하는 것이 좋습니다. 예를 들어 Lambda 함수가 Python으로 작성된 경우 Python 로깅 모듈을 사용하여 메시지를 로깅하고 출력 형식을 제어할 수 있습니다. Lambda는 기본적으로 Python으로 작성된 Lambda 함수용 Python 로깅 라이브러리를 사용하며 Lambda 함수 내에서 로거를 검색하고 사용자 지정할 수 있습니다. AWS Labs는 콜드 스타트와 같은 주요 데이터로 로그 메시지를 더 쉽게 보강할 수 있도록 AWS Lambda Python 개발자용 Powertools 툴킷을 만들었습니다. 도구 키트는 Python, Java, Typescript 및 .NET에 사용할 수 있습니다.

또 다른 모범 사례는 변수를 사용하여 로그 출력 수준을 설정하고 환경 및 요구 사항에 따라 조정하는 것입니다. Lambda 함수의 코드는 사용된 라이브러리 외에도 로그 출력 수준에 따라 대량의 로그 데이터를 출력할 수 있습니다. 이는 로깅 비용에 영향을 미치고 성능에 영향을 미칠 수 있습니다.

Lambda를 사용하면 코드를 업데이트하지 않고도 Lambda 함수 런타임 환경에 대한 환경 변수를 설정할 수 있습니다. 예를 들어 코드에서 검색할 수 있는 로그 출력 수준을 정의하는 LAMBDA_LOG_LEVEL 환경 변수를 생성할 수 있습니다. 다음 예제에서는 LAMBDA_LOG_LEVEL 환경 변수를 검색하고 값을 사용하여 로깅 출력을 정의하려고 시도합니다. 환경 변수가 설정되지 않은 경우 기본적으로 INFO 레벨로 설정됩니다.

```
import logging
from os import getenv

logger = logging.getLogger()
log_level = getenv("LAMBDA_LOG_LEVEL", "INFO")
level = logging.getLevelName(log_level)
logger.setLevel(level)
```

CloudWatch에서 다른 대상으로 로그 전송

구독 필터를 사용하여 다른 대상(예: Amazon OpenSearch Service 또는 Lambda 함수)에 로그를 전송할 수 있습니다. Amazon OpenSearch Service를 사용하지 않는 경우 Lambda 함수를 사용하여 로그를 처리하고 SDK를 사용하여 원하는 AWS 서비스로 전송할 수 있습니다. AWS SDKs

Lambda 함수에서 AWS 클라우드 외부의 로그 대상에 SDKs를 사용하여 원하는 대상으로 로그 문을 직접 보낼 수도 있습니다. 이 옵션을 선택하는 경우 지연 시간, 추가 처리 시간, 오류 및 재시도 처리, Lambda 함수에 대한 운영 로직 결합의 영향을 고려하는 것이 좋습니다.

Lambda 함수 지표

Lambda를 사용하면 서버를 관리하거나 확장하지 않고도 코드를 실행할 수 있으므로 시스템 수준 감사 및 진단의 부담이 거의 없습니다. 그러나 Lambda 함수의 시스템 수준에서 성능 및 호출 지표를 이해하는 것이 여전히 중요합니다. 이렇게 하면 리소스 구성을 최적화하고 코드 성능을 개선할 수 있습니다. 성능을 효과적으로 모니터링하고 측정하면 Lambda 함수의 크기를 적절하게 조정하여 사용자 경험을 개선하고 비용을 절감할 수 있습니다. 일반적으로 Lambda 함수로 실행되는 워크로드에는 캡처하고 분석해야 하는 애플리케이션 수준 지표도 있습니다. Lambda는 임베디드 지표 형식을 직접 지원하여 애플리케이션 수준 CloudWatch 지표를 더 쉽게 캡처할 수 있습니다.

시스템 수준 지표

Lambda는 CloudWatch 지표와 자동으로 통합되며 Lambda 함수에 대한 표준 지표 세트를 제공합니다. 또한 Lambda는 이러한 지표와 함께 각 Lambda 함수에 대한 별도의 모니터링 대시보드를 제공합니다. 모니터링해야 하는 두 가지 중요한 지표는 오류와 호출 오류입니다. 호출 오류와 기타 오류 유형의 차이점을 이해하면 Lambda 배포를 진단하고 지원하는 데 도움이 됩니다.

호출 오류로 인해 Lambda 함수가 실행되지 않습니다. 이러한 오류는 코드가 실행되기 전에 발생하므로 코드 내에서 오류 처리를 구현하여 식별할 수 없습니다. 대신 이러한 오류를 감지하고 작업 및 워크로드 소유자에게 알리는 Lambda 함수에 대한 경보를 구성해야 합니다. 이러한 오류는 종종 구성 또는 권한 오류와 관련이 있으며 구성 또는 권한의 변경으로 인해 발생할 수 있습니다. 호출 오류로 인해 재시도가 시작되어 함수가 여러 번 호출될 수 있습니다.

성공적으로 호출된 Lambda 함수는 함수에서 예외가 발생하더라도 HTTP 200 응답을 반환합니다. Lambda 함수는 오류 처리를 구현하고 `Errors` 지표가 Lambda 함수의 실패한 실행을 캡처하고 식별하도록 예외를 발생시켜야 합니다. Lambda 함수 호출에서 형식이 지정된 응답을 반환해야 하며, 여기에는 실행이 완전히 실패했는지, 부분적으로 실패했는지 또는 성공했는지 확인하는 정보가 포함됩니다.

CloudWatch는 개별 Lambda 함수에 대해 활성화할 수 있는 CloudWatch Lambda Insights를 제공합니다. Lambda Insights는 시스템 수준 지표(예: CPU 시간, 메모리, 디스크 및 네트워크 사용량)를 수집, 집계 및 요약합니다. 또한 Lambda Insights는 진단 정보(예: 콜드 스타트 및 Lambda 작업자 종료)를 수집, 집계 및 요약하여 문제를 격리하고 신속하게 해결하는 데 도움이 됩니다.

Lambda Insights는 임베디드 지표 형식을 사용하여 Lambda 함수의 이름을 기반으로 `/aws/lambda-insights/` 로그 스트림 이름 접두사가 있는 로그 그룹에 성능 정보를 자동으로 내보냅니다. 이러한 성능 로그 이벤트는 자동 CloudWatch 대시보드의 기반인 CloudWatch 지표를 생성합니다. 성능 테스트 및 프로덕션 환경에 대해 Lambda Insights를 활성화하는 것이 좋습니다. Lambda Insights에서 생성한 추가 지표에는 필요하지 않은 용량에 대한 비용을 지불하지 않도록 Lambda 함수의 크기를 올바르게 조정`memory_utilization`하는 데 도움이 되는가 포함됩니다.

애플리케이션 지표

임베디드 지표 형식을 사용하여 CloudWatch에서 자체 애플리케이션 지표를 생성하고 캡처할 수도 있습니다. AWS 임베디드 지표 형식에 제공된 라이브러리를 활용하여 임베디드 지표 형식 문을 생성하고 CloudWatch로 내보낼 수 있습니다. 통합 Lambda CloudWatch 로깅 기능은 적절한 형식의 임베디드 지표 형식 문을 처리하고 추출하도록 구성되어 있습니다.

Searching and analyzing logs in CloudWatch

로그와 지표를 일관된 형식과 위치로 캡처한 후 이를 검색하고 분석하여 문제를 식별하고 해결하는 것 외에도 운영 효율성을 개선할 수 있습니다. 로그를 더 쉽게 검색하고 분석할 수 있도록 로그를 올바른 형식(예: JSON)으로 캡처하는 것이 좋습니다. 대부분의 워크로드는 네트워크, 컴퓨팅, 스토리지 및 데이터베이스와 같은 AWS 리소스 모음을 사용합니다. 가능하면 이러한 리소스의 지표와 로그를 집합적으로 분석하고 상호 연관시켜 모든 AWS 워크로드를 효과적으로 모니터링하고 관리해야 합니다.

CloudWatch는 로그 및 지표를 분석하는 데 도움이 되는 몇 가지 기능을 제공합니다. 예를 들어 CloudWatch Application Insights는 다양한 AWS 리소스의 애플리케이션에 대한 지표와 로그를 총체적으로 정의하고 모니터링하며, CloudWatch Anomaly Detection은 지표에 대한 이상을 표시하고, CloudWatch Logs는 CloudWatch Logs에서 로그 데이터를 대화식으로 검색하고 분석합니다.

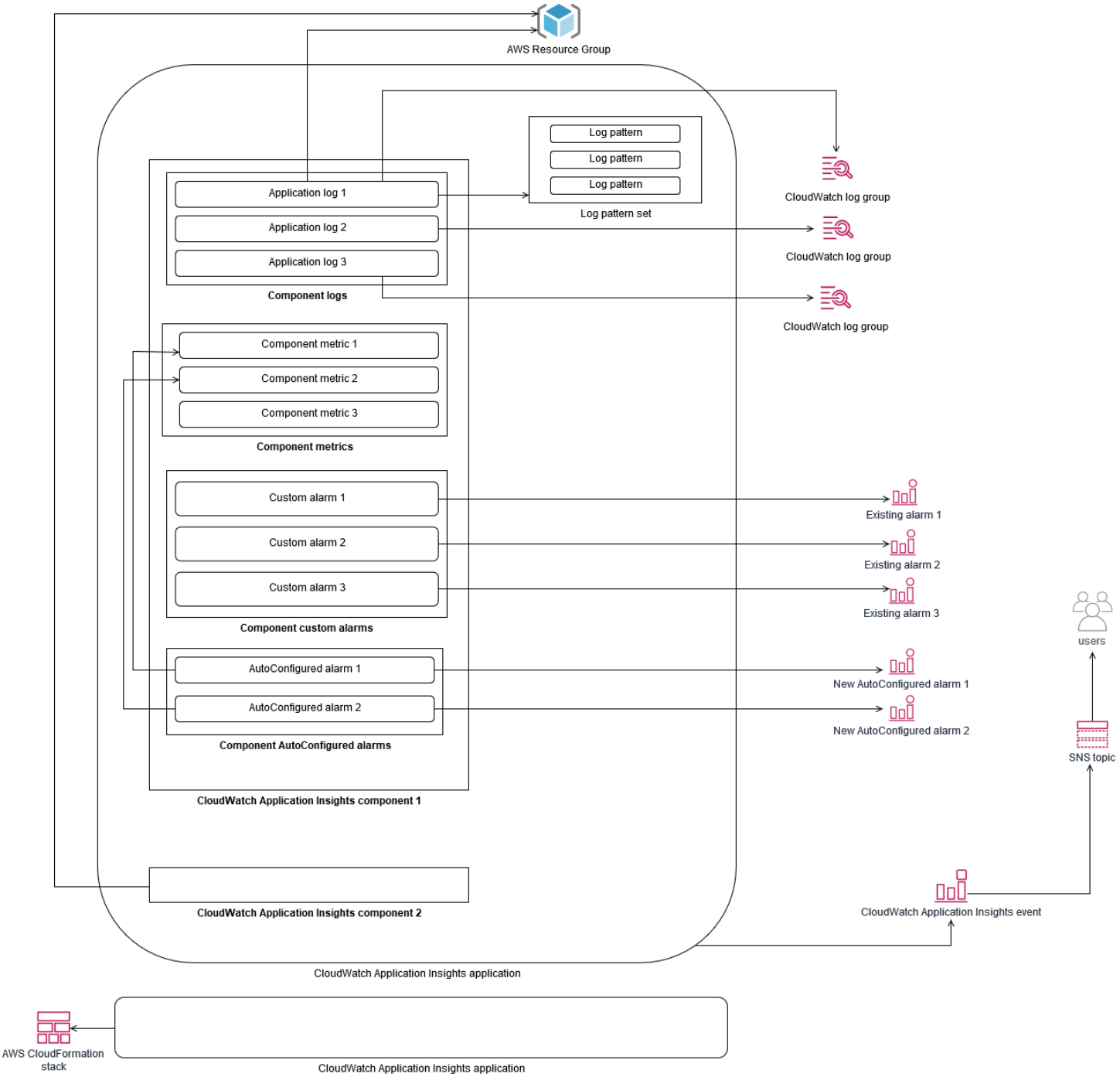
CloudWatch Application Insights를 사용하여 애플리케이션을 종합적으로 모니터링 및 분석

애플리케이션 소유자는 Amazon CloudWatch Application Insights를 사용하여 워크로드에 대한 자동 모니터링 및 분석을 설정할 수 있습니다. 이는 계정의 모든 워크로드에 대해 구성된 표준 시스템 수준 모니터링 외에도 구성할 수 있습니다. 또한 CloudWatch Application Insights를 통해 모니터링을 설정하면 애플리케이션 팀이 사전에 운영에 맞게 조정하고 평균 복구 시간(MTTR)을 줄이는 데 도움이 될 수 있습니다. CloudWatch Application Insights는 애플리케이션 수준 로깅 및 모니터링을 설정하는 데 필요한 노력을 줄이는 데 도움이 될 수 있습니다. 또한 팀이 로깅 및 모니터링 책임을 분할하는 데 도움이 되는 구성 요소 기반 프레임워크를 제공합니다.

CloudWatch Application Insights는 리소스 그룹을 사용하여 애플리케이션으로 집합적으로 모니터링해야 하는 리소스를 식별합니다. 리소스 그룹에서 지원되는 리소스는 CloudWatch Application Insights 애플리케이션의 개별적으로 정의된 구성 요소가 됩니다. CloudWatch Application Insights 애플리케이션의 각 구성 요소에는 자체 로그, 지표 및 경보가 있습니다.

로그의 경우 CloudWatch Application Insights 애플리케이션 내에서 구성 요소에 사용해야 하는 로그 패턴 세트를 정의합니다. 로그 패턴 세트는 패턴이 감지될 때 심각도가 낮음, 중간 또는 높음과 함께 정규식을 기반으로 검색할 로그 패턴 모음입니다. 지표의 경우 서비스별 및 지원되는 지표 목록에서 각 구성 요소에 대해 모니터링할 지표를 선택합니다. 경보의 경우 CloudWatch Application Insights는 모니터링 중인 지표에 대한 표준 또는 이상 탐지 경보를 자동으로 생성하고 구성합니다. CloudWatch Application Insights에는 CloudWatch 설명서의 CloudWatch Application Insights에서 지원하는 로그 및 지표에 설명된 기술에 대한 지표 및 로그 캡처에 대한 자동 구성이 있습니다. 다음 다이어그램은

CloudWatch Application Insights 구성 요소와 로깅 및 모니터링 구성 간의 관계를 보여줍니다. 각 구성 요소는 CloudWatch 로그 및 지표를 사용하여 모니터링할 자체 로그 및 지표를 정의했습니다.



CloudWatch Application Insights에서 모니터링하는 EC2 인스턴스에는 Systems Manager 및 CloudWatch 에이전트와 권한이 필요합니다. 이에 대한 자세한 내용은 CloudWatch 설명서의 CloudWatch Application Insights로 애플리케이션을 구성하기 위한 사전 조건을 참조하세요. CloudWatch CloudWatch Application Insights는 Systems Manager를 사용하여 CloudWatch 에이전트를 설치하고 업데이트합니다. CloudWatch Application Insights에 구성된 지표 및 로그는 각

CloudWatch Application Insights 구성 요소의 `AmazonCloudWatch-ApplicationInsights-SSMParameter` 접두사가 있는 Systems Manager 파라미터에 저장되는 CloudWatch 에이전트 구성 파일을 생성합니다. 이로 인해 EC2 인스턴스의 CloudWatch 에이전트 구성 디렉터리에 별도의 CloudWatch 에이전트 구성 파일이 추가됩니다. Systems Manager 명령이 실행되어이 구성을 EC2 인스턴스의 활성 구성에 추가합니다. CloudWatch Application Insights 사용은 기존 CloudWatch 에이전트 구성 설정에 영향을 주지 않습니다. 자체 시스템 및 애플리케이션 수준 CloudWatch 에이전트 구성 외에도 CloudWatch Application Insights를 사용할 수 있습니다. 그러나 구성이 겹치지 않도록 해야 합니다.

CloudWatch Logs Insights를 사용하여 로그 분석 수행

CloudWatch Logs Insights를 사용하면 간단한 쿼리 언어를 사용하여 여러 로그 그룹을 쉽게 검색할 수 있습니다. 애플리케이션 로그가 JSON 형식으로 구성된 경우 CloudWatch Logs Insights는 여러 로그 그룹의 로그 스트림에서 JSON 필드를 자동으로 검색합니다. CloudWatch Logs Insights를 사용하여 애플리케이션 및 시스템 로그를 분석하여 나중에 사용할 수 있도록 쿼리를 저장할 수 있습니다. CloudWatch Logs Insights의 쿼리 구문은 애플리케이션 문제 해결 또는 성능 분석에 도움이 될 수 있는 sum(), avg(), count(), min() 및 max()와 같은 함수를 사용한 집계와 같은 함수를 지원합니다.

임베디드 지표 형식을 사용하여 CloudWatch 지표를 생성하는 경우 임베디드 지표 형식 로그를 쿼리하여 지원되는 집계 함수를 사용하여 일회성 지표를 생성할 수 있습니다. 이를 통해 사용자 지정 지표로 적극적으로 캡처하는 대신 필요에 따라 특정 지표를 생성하는 데 필요한 데이터 포인트를 캡처하여 CloudWatch 모니터링 비용을 줄일 수 있습니다. 이는 카디널리티가 높아 지표 수가 많은 차원에 특히 효과적입니다. 또한 CloudWatch Container Insights는이 접근 방식을 취하여 자세한 성능 데이터를 캡처하지만이 데이터의 하위 집합에 대한 CloudWatch 지표만 생성합니다.

예를 들어 다음 임베디드 지표 항목은 임베디드 지표 형식 문에 캡처된 지표 데이터에서 제한된 CloudWatch 지표 집합만 생성합니다.

```
{
 "AutoScalingGroupName": "eks-e0bab7f4-fa6c-64ba-dbd9-094aee6cf9ba",
 "CloudWatchMetrics": [
 {
 "Metrics": [
 {
 "Unit": "Count",
 "Name": "pod_number_of_container_restarts"
 }
 ],
 "Dimensions": [
 [
```

```
"PodName",
"Namespace",
"ClusterName"
]
],
"Namespace": "ContainerInsights"
}
],
"ClusterName": "eksdemo",
"InstanceId": "i-03b21a16b854aa4ca",
"InstanceType": "t3.medium",
"Namespace": "amazon-cloudwatch",
"NodeName": "ip-172-31-10-211.ec2.internal",
"PodName": "cloudwatch-agent",
"Sources": [
"cadvisor",
"pod",
"calculated"
],
"Timestamp": "1605111338968",
"Type": "Pod",
"Version": "0",
"pod_cpu_limit": 200,
"pod_cpu_request": 200,
"pod_cpu_reserved_capacity": 10,
"pod_cpu_usage_system": 3.268605094109382,
"pod_cpu_usage_total": 8.899539221131045,
"pod_cpu_usage_user": 4.160042847048305,
"pod_cpu_utilization": 0.44497696105655227,
"pod_cpu_utilization_over_pod_limit": 4.4497696105655224,
"pod_memory_cache": 4096,
"pod_memory_failcnt": 0,
"pod_memory_hierarchical_pgfault": 0,
"pod_memory_hierarchical_pgmajfault": 0,
"pod_memory_limit": 209715200,
"pod_memory_mapped_file": 0,
"pod_memory_max_usage": 43024384,
"pod_memory_pgfault": 0,
"pod_memory_pgmajfault": 0,
"pod_memory_request": 209715200,
"pod_memory_reserved_capacity": 5.148439982463127,
"pod_memory_rss": 38481920,
"pod_memory_swap": 0,
"pod_memory_usage": 42803200,
```

```
 "pod_memory_utilization": 0.6172094650851303,
 "pod_memory_utilization_over_pod_limit": 11.98828125,
 "pod_memory_working_set": 25141248,
 "pod_network_rx_bytes": 3566.4174629544723,
 "pod_network_rx_dropped": 0,
 "pod_network_rx_errors": 0,
 "pod_network_rx_packets": 3.3495665260575094,
 "pod_network_total_bytes": 4283.442421354973,
 "pod_network_tx_bytes": 717.0249584005006,
 "pod_network_tx_dropped": 0,
 "pod_network_tx_errors": 0,
 "pod_network_tx_packets": 2.6964010534762948,
 "pod_number_of_container_restarts": 0,
 "pod_number_of_containers": 1,
 "pod_number_of_running_containers": 1,
 "pod_status": "Running"
}
```


그러나 캡처된 지표를 쿼리하여 추가 인사이트를 얻을 수 있습니다. 예를 들어 다음 쿼리를 실행하여 메모리 페이지 장애가 있는 최신 포드 20개를 확인할 수 있습니다.

```
fields @timestamp, @message
| filter (pod_memory_pgfault > 0)
| sort @timestamp desc
| limit 20
```

Amazon OpenSearch Service를 사용하여 로그 분석 수행

CloudWatch는 구독 필터를 사용하여 CloudWatch 로그 그룹에서 원하는 Amazon OpenSearch Service 클러스터로 로그 데이터를 스트리밍할 수 있도록 하여 Amazon OpenSearch Service와 통합됩니다. 기본 로그 및 지표 캡처 및 분석에 CloudWatch를 사용한 다음 다음 다음 사용 사례에 대해 Amazon OpenSearch Service로 보강할 수 있습니다.

- 세분화된 데이터 액세스 제어 - Amazon OpenSearch Service를 사용하면 데이터에 대한 액세스를 필드 수준까지 제한하고 사용자 권한에 따라 필드의 데이터를 익명화할 수 있습니다. 이는 민감한 데이터를 노출하지 않고 문제 해결을 지원하려는 경우에 유용합니다.
- 여러 계정, 리전 및 인프라에서 로그 집계 및 검색 - 여러 계정 및 리전의 로그를 공통 Amazon OpenSearch Service 클러스터로 스트리밍할 수 있습니다. 중앙 집중식 운영 팀은 추세, 문제를 분석하고 계정 및 리전 간에 분석을 수행할 수 있습니다. 또한 CloudWatch 로그를 Amazon OpenSearch Service로 스트리밍하면 중앙 위치에서 다중 리전 애플리케이션을 검색하고 분석할 수 있습니다.

- ElasticSearch 에이전트를 사용하여 로그를 Amazon OpenSearch Service로 직접 발송하고 보강 ElasticSearch - 애플리케이션 및 기술 스택 구성 요소는 CloudWatch 에이전트에서 지원하지 않는 OSs를 사용할 수 있습니다. 로깅 솔루션으로 전송되기 전에 로그 데이터를 보강하고 변환할 수도 있습니다. Amazon OpenSearch Service는 로그 데이터를 Amazon OpenSearch Service로 보내기 전에 로그 보강 및 변환을 지원하는 Elastic Beats 패밀리 데이터 전송자 및 Logstash와 같은 표준 Elasticsearch OpenSearch 클라이언트를 지원합니다. https://docs.aws.amazon.com//opensearch-service/latest/developerguide/managedomains-logstash.html
- 기존 운영 관리 솔루션은 로깅 및 모니터링을 위해 ElasticSearch, Logstash, Kibana(ELK) 스택을 사용합니다. 이미 많은 워크로드가 구성된 Amazon OpenSearch Service 또는 오픈 소스 Elasticsearch에 상당한 투자를 하고 있을 수 있습니다. Kibana에서 생성되어 계속 사용하려는 운영 대시보드가 있을 수도 있습니다.

CloudWatch 로그를 사용할 계획이 없는 경우 Amazon OpenSearch Service 지원 에이전트, 로그 드라이버 및 라이브러리(예: Fluent Bit, Fluentd, logstash 및 Open Distro for ElasticSearch API)를 사용하여 로그를 Amazon OpenSearch Service로 직접 전송하고 CloudWatch를 우회할 수 있습니다. 그러나 AWS 서비스에서 생성된 로그를 캡처하는 솔루션도 구현해야 합니다. CloudWatch Logs는 많은 AWS 서비스를 위한 기본 로그 캡처 솔루션이며 여러 서비스가 CloudWatch에서 새 로그 그룹을 자동으로 생성합니다. 예를 들어 Lambda는 모든 Lambda 함수에 대해 새 로그 그룹을 생성합니다. 로그 그룹이 Amazon OpenSearch Service로 로그를 스트리밍하도록 구독 필터를 설정할 수 있습니다. Amazon OpenSearch Service로 스트리밍하려는 각 개별 로그 그룹에 대해 구독 필터를 수동으로 구성할 수 있습니다. 또는 새 로그 그룹을 ElasticSearch 클러스터에 자동으로 구독하는 솔루션을 배포할 수 있습니다. 동일한 계정 또는 중앙 집중식 계정의 ElasticSearch 클러스터로 로그를 스트리밍할 수 있습니다. 동일한 계정의 ElasticSearch 클러스터로 로그를 스트리밍하면 워크로드 소유자가 워크로드를 더 잘 분석하고 지원할 수 있습니다.

계정, 리전 및 애플리케이션 간에 로그를 집계하기 위해 중앙 집중식 또는 공유 계정에서 ElasticSearch 클러스터를 설정하는 것을 고려해야 합니다. 예를 들어, 중앙 집중식 로깅에 사용되는 로그 아카이브 계정을 AWS Control Tower 설정합니다. 에서 새 계정이 생성되면 AWS Control Tower AWS CloudTrail 및 AWS Config 로그가이 중앙 집중식 계정의 S3 버킷으로 전송됩니다. 에서 계측한 로깅 AWS Control Tower 은 구성, 변경 및 감사 로깅을 위한 것입니다.

Amazon OpenSearch Service를 사용하여 중앙 집중식 애플리케이션 로그 분석 솔루션을 설정하려면 중앙 집중식 로깅 계정에 하나 이상의 중앙 집중식 Amazon OpenSearch Service 클러스터를 배포하고 다른 계정의 로그 그룹을 구성하여 중앙 집중식 Amazon OpenSearch Service 클러스터로 로그를 스트리밍할 수 있습니다.

별도의 Amazon OpenSearch Service 클러스터를 생성하여 계정 전체에 분산될 수 있는 클라우드 아키텍처의 다양한 애플리케이션 또는 계층을 처리할 수 있습니다. 별도의 Amazon OpenSearch Service 클러스터를 사용하면 보안 및 가용성 위험을 줄일 수 있으며 일반적인 Amazon OpenSearch Service 클러스터를 사용하면 동일한 클러스터 내에서 데이터를 더 쉽게 검색하고 연결할 수 있습니다.

CloudWatch를 사용한 경보 옵션

중요한 지표에 대한 일회성 자동 분석을 수행하면 워크로드에 영향을 미치기 전에 문제를 감지하고 해결하는 데 도움이 됩니다. CloudWatch를 사용하면 특정 기간 동안 여러 통계를 사용하여 여러 지표를 쉽게 그래프로 표시하고 비교할 수 있습니다. CloudWatch를 사용하여 필요한 차원 값으로 모든 지표를 검색하여 분석에 필요한 지표를 찾을 수 있습니다.

워크로드 모니터링의 기준으로 사용할 초기 지표 및 차원 세트를 포함하여 지표 캡처 접근 방식을 시작하는 것이 좋습니다. 시간이 지남에 따라 워크로드가 성숙해지고 추가 지표와 차원을 추가하여 이를 추가로 분석하고 지원할 수 있습니다. 애플리케이션 또는 워크로드는 여러 AWS 리소스를 사용하고 자체 사용자 지정 지표가 있을 수 있으므로 이러한 리소스를 쉽게 식별할 수 있도록 네임스페이스로 그룹화해야 합니다.

또한 관련 로깅 및 모니터링 데이터를 신속하게 식별하여 특정 문제를 진단할 수 있도록 데이터 로깅 및 모니터링의 상관 관계를 고려해야 합니다. AWS X-Ray 추적 맵을 사용하여 문제 진단에 대한 추적, 지표, 로그 및 경보를 상호 연관시킬 수 있습니다. 또한 시스템 및 서비스 전반의 문제를 빠르게 검색하고 식별하는 데 도움이 되도록 워크로드의 로그에 지표 및 식별자에 추가 차원을 포함하는 것도 고려해야 합니다.

CloudWatch 경보를 사용하여 모니터링 및 경보

CloudWatch 경보를 사용하여 워크로드 또는 애플리케이션의 수동 모니터링을 줄일 수 있습니다. 먼저 각 워크로드 구성 요소에 대해 캡처하는 지표를 검토하고 각 지표에 적합한 임계값을 결정해야 합니다. 임계값 위반 시 알림을 받아야 하는 팀원을 식별해야 합니다. 개별 팀원이 아닌 배포 그룹을 설정하고 대상으로 지정해야 합니다.

CloudWatch 경보는 서비스 관리 솔루션과 통합되어 새 티켓을 자동으로 생성하고 운영 워크플로를 실행할 수 있습니다. 예를 들어,는 통합을 빠르게 설정하는 AWS Service Management Connector 데 도움이 되도록 ServiceNow 및 용 AWS 서비스 관리 커넥터를 AWS 제공합니다. 이 접근 방식은 발생한 경보를 확인하고 이러한 제품에 이미 정의되어 있을 수 있는 기존 운영 워크플로에 맞게 조정하는 데 매우 중요합니다.

임계값과 평가 기간이 다른 동일한 지표에 대해 여러 경보를 생성할 수도 있으므로 에스컬레이션 프로세스를 설정하는 데 도움이 됩니다. 예를 들어 고객 주문을 추적하는 `OrderQueueDepth` 지표가 있는 경우 애플리케이션 팀원에게 이메일 또는 Slack으로 알리는 짧은 1분 평균 기간 동안 더 낮은 임계값을 정의할 수 있습니다. 또한 15분 이상 동일한 임계값에서 동일한 지표에 대해 다른 경보를 정의하고 해당 페이지, 이메일을 전송하고 애플리케이션 팀과 애플리케이션 팀의 리드에게 알릴 수 있습니다. 마지

막으로 상위 관리를 알리고 이전에 알림을 받은 모든 팀원에게 알리는 30분 동안의 하드 평균 임계값에 대한 세 번째 경보를 정의할 수 있습니다. 여러 경보를 생성하면 다양한 조건에 대해 다양한 작업을 수행하는 데 도움이 됩니다. 간단한 알림 프로세스로 시작한 다음 필요에 따라 조정하고 개선할 수 있습니다.

CloudWatch 이상 탐지를 사용하여 모니터링 및 경보

특정 지표에 적용할 임계값에 대해 확실하지 않거나 경보가 관찰된 기록 값을 기반으로 임계값을 자동으로 조정하도록 하려는 경우 CloudWatch 이상 탐지를 사용할 수 있습니다. CloudWatch 이상 탐지는 마감 시간 전에 증가하는 당일 배송에 대한 일일 구매 주문과 같이 활동에 정기적이고 예측 가능한 변화가 있을 수 있는 지표에 특히 유용합니다. 이상 탐지는 자동으로 조정되는 임계값을 활성화하고 거짓 경보를 줄이는 데 도움이 될 수 있습니다. 각 지표 및 통계에 대해 이상 탐지를 활성화하고 이상치를 기반으로 경보를 울리도록 CloudWatch를 구성할 수 있습니다.

예를 들어 EC2 인스턴스에서 `CPUUtilization` 지표 및 `AVG` 통계에 대한 이상 탐지를 활성화할 수 있습니다. 그런 다음 이상 탐지는 최대 14일의 기록 데이터를 사용하여 기계 학습(ML) 모델을 생성합니다. 서로 다른 이상 탐지 밴드로 여러 경보를 생성하여 임계값이 서로 다른 여러 표준 경보를 생성하는 것과 마찬가지로 경보 에스컬레이션 프로세스를 설정할 수 있습니다.

이 섹션에 대한 자세한 내용은 CloudWatch 설명서의 이상 탐지를 기반으로 CloudWatch 경보 생성을 참조하세요. CloudWatch

여러 리전 및 계정에서 경보 발생

애플리케이션 및 워크로드 소유자는 여러 리전에 걸쳐 있는 워크로드에 대한 애플리케이션 수준 경보를 생성해야 합니다. 워크로드가 배포된 각 계정 및 리전 내에 별도의 경보를 생성하는 것이 좋습니다. 계정 및 리전에 구애받지 않는 CloudFormation StackSets 및 템플릿을 사용하여 필요한 경보와 함께 애플리케이션 리소스를 배포하여이 프로세스를 간소화하고 자동화할 수 있습니다. templateYou일반적인 Amazon Simple Notification Service(Amazon SNS) 주제를 대상으로 하는 경보 작업을 구성할 수 있습니다. 즉, 계정 또는 리전에 관계없이 동일한 알림 또는 수정 작업이 사용됩니다.

다중 계정 및 다중 리전 환경에서는 StackSets를 사용하고 모든 EC2 인스턴스`CPUUtilization`의 평균과 같은 집계 지표를 사용하여 CloudFormation 계정 및 리전 문제를 모니터링하기 위해 계정 및 리전에 대한 집계 경보를 생성하는 것이 좋습니다.

또한 캡처하는 표준 CloudWatch 지표 및 로그에 대해 구성된 각 워크로드에 대해 표준 경보를 생성하는 것도 고려해야 합니다. 예를 들어 CPU 사용률 지표를 모니터링하고 평균 CPU 사용률이 매일 80%를 초과하면 중앙 운영 팀에 알리는 각 EC2 인스턴스에 대해 별도의 경보를 생성할 수 있습니다. 또한

평균 CPU 사용률을 매일 10% 미만으로 모니터링하는 표준 경보를 생성할 수 있습니다. 이러한 경보는 중앙 운영 팀이 특정 워크로드 소유자와 협력하여 필요한 경우 EC2 인스턴스의 크기를 변경하는 데 도움이 됩니다.

EC2 인스턴스 태그를 사용하여 경보 생성 자동화

EC2 인스턴스에 대한 표준 경보 세트를 생성하면 시간이 많이 걸리고 일관되지 않으며 오류가 발생하기 쉽습니다. amazon-cloudwatch-auto-alarms 솔루션을 사용하여 EC2 인스턴스에 대한 표준 CloudWatch 경보 세트를 자동으로 생성하고 EC2 인스턴스 태그를 기반으로 사용자 지정 경보를 생성하여 경보 생성 프로세스를 가속화할 수 있습니다. 이 솔루션은 표준 경보를 수동으로 생성할 필요가 없으며 CloudEndure와 같은 도구를 사용하는 EC2 인스턴스를 대규모로 마이그레이션하는 동안 유용할 수 있습니다. 또한 CloudFormation StackSets를 사용하여이 솔루션을 배포하여 여러 리전 및 계정을 지원할 수 있습니다. 자세한 내용은 블로그에서 태그를 사용하여 Amazon EC2 인스턴스에 대한 Amazon CloudWatch 경보 생성 및 유지 Amazon EC2 관리를 참조하세요 AWS .

애플리케이션 및 서비스 가용성 모니터링

CloudWatch를 사용하면 애플리케이션 및 워크로드의 성능 및 런타임 측면을 모니터링하고 분석할 수 있습니다. 또한 애플리케이션 및 워크로드의 가용성 및 연결성 측면을 모니터링해야 합니다. Amazon Route 53 상태 확인 및 CloudWatch Synthetics와 함께 활성 모니터링 접근 방식을 사용하여 이를 달성할 수 있습니다.

HTTP 또는 HTTPS를 통한 웹 페이지 연결 또는 TCP를 통한 퍼블릭 도메인 이름 시스템(DNS) 이름 또는 IP 주소에 대한 네트워크 연결을 모니터링하려는 경우 Route 53 상태 확인을 사용할 수 있습니다. Route 53 상태 확인은 10초 또는 30초 간격으로 지정한 리전에서 연결을 시작합니다. 상태 확인을 실행할 여러 리전을 선택할 수 있으며, 각 상태 확인은 독립적으로 실행되며, 최소 3개의 리전을 선택해야 합니다. HTTP 또는 HTTPS 요청의 응답 본문이 상태 확인 평가를 위해 반환된 처음 5,120바이트의 데이터에 나타나는 경우 특정 하위 문자열을 검색할 수 있습니다. HTTP 또는 HTTPS 요청은 2xx 또는 3xx 응답을 반환하는 경우 정상으로 간주됩니다. Route 53 상태 확인을 사용하여 다른 상태 확인의 상태를 확인하여 복합 상태 확인을 생성할 수 있습니다. 서비스 엔드포인트가 여러 개 있고 둘 중 하나가 비정상이 될 때 동일한 알림을 수행하려는 경우이 작업을 수행할 수 있습니다. DNS에 Route 53를 사용하는 경우 상태 확인이 비정상이 되면 다른 DNS 항목으로 장애 조치하도록 Route 53을 구성할 수 있습니다. 각 중요 워크로드에 대해 정상 운영에 중요한 외부 엔드포인트에 대한 Route 53 상태 확인을 설정하는 것을 고려해야 합니다. Route 53 상태 확인을 사용하면 애플리케이션에 장애 조치 로직을 쓰지 않아도 됩니다.

CloudWatch 합성을 사용하면 카나리아를 스크립트로 정의하여 워크로드의 상태와 가용성을 평가할 수 있습니다. 카나리아는 Node.js 또는 Python으로 작성된 스크립트이며 HTTP 또는 HTTPS 프로토콜을 통해 작동합니다. Node.js 또는 Python을 프레임워크로 사용하는 Lambda 함수를 계정에 생성합니다. 정의한 각 canary는 여러 엔드포인트에 대해 여러 HTTP 또는 HTTPS 호출을 수행할 수 있습니다. 즉, 사용 사례 또는 다운스트림 종속성이 있는 엔드포인트와 같은 일련의 단계의 상태를 모니터링할 수 있습니다. 카나리아는 실행된 각 단계를 포함하는 CloudWatch 지표를 생성하므로 서로 다른 단계를 독립적으로 경보하고 측정할 수 있습니다. 카나리아는 Route 53 상태 확인보다 더 많은 계획과 노력이 필요하지만 고도로 사용자 지정 가능한 모니터링 및 평가 접근 방식을 제공합니다. Canary는 Virtual Private Cloud(VPC) 내에서 실행되는 프라이빗 리소스도 지원하므로 엔드포인트에 대한 퍼블릭 IP 주소가 없는 경우 가용성 모니터링에 적합합니다. VPC 내에서 엔드포인트로 연결되어 있는 한 canary를 사용하여 온프레미스 워크로드를 모니터링할 수도 있습니다. 이는 온프레미스에 존재하는 엔드포인트가 포함된 워크로드가 있는 경우 특히 중요합니다.

를 사용하여 애플리케이션 추적 AWS X-Ray

애플리케이션을 통한 요청은 온프레미스 서버, Amazon EC2, 컨테이너 또는 Lambda에서 실행되는 데이터베이스, 애플리케이션 및 웹 서비스에 대한 호출로 구성될 수 있습니다. 애플리케이션 추적을 구현하면 분산 구성 요소 및 서비스를 사용하는 애플리케이션에서 문제의 근본 원인을 빠르게 식별할 수 있습니다. AWS X-Ray를 사용하여 여러 구성 요소에서 애플리케이션 요청을 추적할 수 있습니다. X-Ray는 요청이 애플리케이션 구성 요소를 통과하고 각 구성 요소가 세그먼트로 표시될 때 서비스 그래프에서 요청을 샘플링하고 시각화합니다. X-Ray는 추적 식별자를 생성하므로 요청이 여러 구성 요소를 통과할 때 요청을 상호 연관시킬 수 있으므로 요청을 처음부터 끝까지 볼 수 있습니다. 요청의 특성을 고유하게 검색하고 식별하는 데 도움이 되는 주석과 메타데이터를 포함하여 이를 더욱 개선할 수 있습니다.

X-Ray를 사용하여 애플리케이션의 각 서버 또는 엔드포인트를 구성하고 계측하는 것이 좋습니다. X-Ray는 X-Ray 서비스를 호출하여 애플리케이션 코드에서 구현됩니다. 또한 X-Ray는 X-Ray로 데이터를 자동으로 전송하는 계측 클라이언트를 포함하여 여러 언어에 AWS SDKs를 제공합니다. X-Ray SDK는 다른 서비스(예: HTTP, MySQL, PostgreSQL 또는 MongoDB)를 직접적으로 호출하는 데 사용되는 공통 라이브러리에 대한 패치를 제공합니다.

X-Ray는 Amazon EC2 및 Amazon ECS에서 설치 및 실행하여 데이터를 X-Ray로 릴레이할 수 있는 X-Ray 데몬을 제공합니다. X-Ray는 요청에 서비스를 제공한 X-Ray 데몬을 실행하는 서버 및 컨테이너에서 성능 데이터를 캡처하는 애플리케이션에 대한 추적을 생성합니다. X-Ray는 AWS SDK 패치를 통해 Amazon DynamoDB와 같은 AWS 서비스에 대한 호출을 하위 세그먼트로 자동으로 계측합니다. X-Ray는 Lambda 함수와 자동으로 통합할 수도 있습니다.

애플리케이션 구성 요소가 X-Ray 데몬을 구성 및 설치하거나 코드를 계측할 수 없는 외부 서비스를 호출하는 경우 하위 세그먼트를 생성하여 외부 서비스에 대한 호출을 래핑할 수 있습니다. X-Ray는를 사용하는 경우 CloudWatch 로그 및 지표를 애플리케이션 추적과 연관시킵니다. 즉 AWS X-Ray SDK for Java, 요청에 대한 관련 지표 및 로그를 신속하게 분석할 수 있습니다.

Amazon EC2에서 애플리케이션 및 서비스를 추적하기 위한 X-Ray 데몬 배포

애플리케이션 구성 요소 또는 마이크로서비스가 실행되는 EC2 인스턴스에 X-Ray 데몬을 설치하고 실행해야 합니다. EC2 인스턴스가 프로비저닝될 때 사용자 데이터 스크립트를 사용하여 X-Ray 데몬을 배포하거나 자체 AMI를 생성하는 경우 AMIs 빌드 프로세스에 포함할 수 있습니다. 이는 EC2 인스턴스가 휘발성일 때 특히 유용할 수 있습니다.

상태 관리자를 사용하여 X-Ray 데몬이 EC2 인스턴스에 일관되게 설치되도록 해야 합니다. Amazon EC2 Windows 인스턴스의 경우 Systems Manager AWS-RunPowerShellScript 문서를 사용하여 X-Ray 에이전트를 다운로드하고 설치하는 Windows 스크립트를 실행할 수 있습니다. Linux의 EC2 인스턴스의 경우 AWS-RunShellScript 문서를 사용하여 에이전트를 서비스로 다운로드하고 설치하는 Linux 스크립트를 실행할 수 있습니다.

Systems Manager AWS-RunRemoteScript 문서를 사용하여 다중 계정 환경에서 스크립트를 실행할 수 있습니다. 모든 계정에서 액세스할 수 있는 S3 버킷을 생성해야 하며를 사용하는 경우 조직 기반 버킷 정책을 사용하여 S3 버킷을 생성하는 것이 좋습니다 AWS Organizations. 그런 다음 스크립트를 S3 버킷에 업로드하되 EC2 인스턴스의 IAM 역할에 버킷 및 스크립트에 액세스할 수 있는 권한이 있는지 확인합니다.

스크립트를 X-Ray 에이전트가 설치된 EC2 인스턴스에 연결하도록 State Manager를 구성할 수도 있습니다. 모든 EC2 인스턴스가 X-Ray를 요구하거나 사용하지 않을 수 있으므로 인스턴스 태그와의 연결을 대상으로 지정할 수 있습니다. 예를 들어 또는 `InstallAWSXRayDaemonWindows` `InstallAWSXRayDaemonLinux` 태그의 존재 여부에 따라 상태 관리자 연결을 생성할 수 있습니다.

Amazon ECS 또는 Amazon EKS에서 애플리케이션 및 서비스를 추적하기 위한 X-Ray 데몬 배포

X-Ray 데몬을 Amazon ECS 또는 Amazon EKS와 같은 컨테이너 기반 워크로드의 사이드카 컨테이너로 배포할 수 있습니다. 그런 다음 Amazon ECS를 사용하는 경우 애플리케이션 컨테이너를 컨테이너 연결로 사이드카 컨테이너에 연결하거나 awsvpc 네트워크 모드를 사용하는 경우 컨테이너를 localhost의 사이드카 컨테이너에 직접 연결할 수 있습니다.

Amazon EKS의 경우 애플리케이션의 포드 정의에서 X-Ray 데몬을 정의하면 애플리케이션이 지정한 컨테이너 포트의 localhost를 통해 데몬에 연결할 수 있습니다.

요청을 X-Ray로 추적하도록 Lambda 구성

애플리케이션에 Lambda 함수에 대한 호출이 포함될 수 있습니다. 데몬 프로세스는 Lambda에서 완전히 관리되며 사용자가 구성할 수 없으므로 Lambda용 X-Ray 데몬을 설치할 필요가 없습니다. 를 사용하고 X-Ray 콘솔에서 활성 추적 옵션을 AWS Management Console 확인하여 Lambda 함수에 대해 이를 활성화할 수 있습니다.

추가 계측을 위해 X-Ray SDK를 Lambda 함수와 번들링하여 발신 호출을 기록하고 주석 또는 메타데이터를 추가할 수 있습니다.

X-Ray용 애플리케이션 계측

애플리케이션의 프로그래밍 언어에 맞는 X-Ray SDK를 평가하고 애플리케이션이 다른 시스템에 수행하는 모든 호출을 분류해야 합니다. 선택한 라이브러리에서 제공하는 클라이언트를 검토하고 SDK가 애플리케이션의 요청 또는 응답에 대한 추적을 자동으로 계측할 수 있는지 확인합니다. SDK에서 제공하는 클라이언트를 다른 다운스트림 시스템에 사용할 수 있는지 확인합니다. 애플리케이션이 호출하고 X-Ray로 계측할 수 없는 외부 시스템의 경우 추적 정보에서 이를 캡처하고 식별할 사용자 지정 하위 세그먼트를 생성해야 합니다.

애플리케이션을 계측할 때 요청을 식별하고 검색하는 데 도움이 되는 주석을 생성해야 합니다. 예를 들어 애플리케이션은와 같은 고객의 식별자를 사용하거나 애플리케이션의 역할에 따라 다른 사용자를 `customer id`세그먼트화할 수 있습니다.

각 트레이스에 대해 최대 50개의 주석을 생성할 수 있지만 세그먼트 문서가 64KB를 초과하지 않는 한 하나 이상의 필드가 포함된 메타데이터 객체를 생성할 수 있습니다. 주석을 선택적으로 사용하여 정보를 찾고 메타데이터 객체를 사용하여 요청을 찾은 후 문제를 해결하는 데 도움이 되는 추가 컨텍스트를 제공해야 합니다.

X-Ray 샘플링 규칙 구성

샘플링 규칙을 사용자 지정하면 코드를 수정하거나 재배포하지 않고도 기록하는 데이터의 양을 제어하고 샘플링 동작을 수정할 수 있습니다. 샘플링 규칙은 X-Ray SDK에 기준 세트에 대해 얼마나 많은 요청을 기록할지 알려줍니다. 기본적으로 X-Ray SDK는 초당 첫 번째 요청과 추가 요청의 5%를 기록합니다. 초당 하나의 요청은 리저버입니다. 이는 서비스가 요청을 처리 중인 동안 하나 이상의 트레이스가 매초 기록되도록 합니다. 5%는 리저버 크기를 초과하여 추가 요청이 샘플링되는 비율입니다.

기본 구성을 검토하고 업데이트하여 계정에 적합한 값을 결정해야 합니다. 요구 사항은 개발, 테스트, 성능 테스트 및 프로덕션 환경에 따라 다를 수 있습니다. 수신하는 트래픽 양 또는 중요도 수준에 따라 자체 샘플링 규칙이 필요한 애플리케이션이 있을 수 있습니다. 기준선으로 시작하여 기준선이 요구 사항을 충족하는지 정기적으로 재평가해야 합니다.

Dashboards and visualizations with CloudWatch

대시보드를 사용하면 애플리케이션 및 워크로드에 대한 관심 영역에 빠르게 집중할 수 있습니다. CloudWatch는 자동 대시보드를 제공하며 CloudWatch 지표를 사용하는 대시보드를 쉽게 생성할 수도 있습니다. CloudWatch 대시보드는 여러 지표를 상호 연관시키고 추세를 식별하는 데 도움이 되므로 지표를 개별적으로 보는 것보다 더 많은 인사이트를 제공합니다. 예를 들어 수신된 주문, 메모리, CPU 사용률 및 데이터베이스 연결이 포함된 대시보드를 사용하면 주문 수가 증가하거나 감소하는 동안 여러 AWS 리소스에서 워크로드 지표의 변화를 상호 연관시킬 수 있습니다.

계정 및 애플리케이션 수준에서 대시보드를 생성하여 워크로드 및 애플리케이션을 모니터링해야 합니다. 서비스별 지표로 미리 구성된 서비스 수준 대시보드인 AWS CloudWatch 자동 대시보드를 사용하여 시작할 수 있습니다. 자동 서비스 대시보드에는 서비스에 대한 모든 표준 CloudWatch 지표가 표시됩니다. 자동 대시보드는 각 서비스 지표에 사용되는 모든 리소스를 그래프로 표시하고 계정 전체에서 이상치 리소스를 빠르게 식별하는 데 도움이 됩니다. 이렇게 하면 사용률이 높거나 낮은 리소스를 식별할 수 있으므로 비용을 최적화하는 데 도움이 될 수 있습니다.

교차 서비스 대시보드 생성

서비스에 대한 자동 서비스 수준 대시보드를 보고 작업 메뉴에서 대시보드에 추가 옵션을 사용하여 교차 AWS 서비스 대시보드를 생성할 수 있습니다. 그런 다음 다른 자동 대시보드의 지표를 새 대시보드에 추가하고 지표를 제거하여 대시보드의 초점을 좁힐 수 있습니다. 또한 사용자 지정 지표를 추가하여 주요 관측치(예: 수신된 주문 또는 초당 트랜잭션)를 추적해야 합니다. 자체 사용자 지정 교차 서비스 대시보드를 생성하면 워크로드와 가장 관련성이 높은 지표에 집중할 수 있습니다. 주요 지표를 다루고 계정의 모든 워크로드를 표시하는 계정 수준의 교차 서비스 대시보드를 생성하는 것이 좋습니다.

클라우드 운영 팀을 위한 중앙 사무실 공간 또는 공용 공간이 있는 경우 자동 새로 고침을 사용하여 대형 TV 모니터에 전체 화면 모드로 CloudWatch 대시보드를 표시할 수 있습니다.

애플리케이션 또는 워크로드별 대시보드 생성

프로덕션 환경의 모든 중요한 애플리케이션 또는 워크로드에 대한 주요 지표 및 리소스에 초점을 맞춘 애플리케이션 및 워크로드별 대시보드를 생성하는 것이 좋습니다. 애플리케이션 및 워크로드별 대시보드는 사용자 지정 애플리케이션 또는 워크로드 지표와 성능에 영향을 미치는 중요한 AWS 리소스 지표에 중점을 둡니다.

인시던트가 발생한 후 주요 지표를 추적하려면 CloudWatch 애플리케이션 또는 워크로드 대시보드를 정기적으로 평가하고 사용자 지정해야 합니다. 또한 기능이 도입되거나 사용 중지될 때 애플리케이션

또는 워크로드별 대시보드를 업데이트해야 합니다. 워크로드 및 애플리케이션별 대시보드 업데이트는 로깅 및 모니터링 외에도 품질을 지속적으로 개선하는 데 필요한 활동이어야 합니다.

교차 계정 또는 교차 리전 대시보드 생성

AWS 리소스는 주로 리전별이며 지표, 경보 및 대시보드는 리소스가 배포된 리전에 따라 다릅니다. 이렇게 하려면 리전 간 워크로드 및 애플리케이션에 대한 지표, 대시보드 및 경보를 보도록 리전을 변경해야 할 수 있습니다. 애플리케이션과 워크로드를 여러 계정으로 분리하는 경우 각 계정에 다시 인증하고 로그인해야 할 수도 있습니다. 그러나 CloudWatch는 단일 계정에서 교차 계정 및 교차 리전 데이터 보기를 지원하므로 단일 계정 및 리전에서 지표, 경보, 대시보드 및 로그 위젯을 볼 수 있습니다. 이는 중앙 집중식 로깅 및 모니터링 계정이 있는 경우에 매우 유용합니다.

계정 소유자와 애플리케이션 팀 소유자는 중앙 위치에서 주요 지표를 효과적으로 모니터링하기 위해 계정별 교차 리전 애플리케이션에 대한 대시보드를 생성해야 합니다. CloudWatch 대시보드는 교차 리전 위젯을 자동으로 지원하므로 추가 구성 없이 여러 리전의 지표가 포함된 대시보드를 생성할 수 있습니다.

로그 데이터는 현재 로그인한 계정 및 리전에 대해서만 표시할 수 있으므로 CloudWatch Logs Insights 위젯은 중요한 예외입니다. 지표 필터를 사용하여 로그에서 리전별 지표를 생성할 수 있으며 이러한 지표는 리전 간 대시보드에 표시될 수 있습니다. 그런 다음 해당 로그를 추가로 분석해야 할 때 특정 리전으로 전환할 수 있습니다.

운영 팀은 중요한 교차 계정 및 교차 리전 지표를 모니터링하는 중앙 집중식 대시보드를 생성해야 합니다. 예를 들어 각 계정 및 리전의 집계 CPU 사용률을 포함하는 교차 계정 대시보드를 생성할 수 있습니다. 지표 수학을 사용하여 여러 계정 및 리전에서 및 대시보드 데이터를 집계할 수도 있습니다.

지표 수학을 사용하여 관찰성 및 경보 미세 조정

지표 수학을 사용하여 워크로드와 관련된 형식 및 표현식으로 지표를 계산할 수 있습니다. 계산된 지표는 추적을 위해 대시보드에 저장하고 볼 수 있습니다. 예를 들어 표준 Amazon EBS 볼륨 지표는 특정 기간 동안 수행된 읽기(`VolumeReadOps`) 및 쓰기(`VolumeWriteOps`) 작업 수를 제공합니다.

그러나는 IOPS로 Amazon EBS 볼륨 성능에 대한 지침을 AWS 제공합니다. `VolumeReadOps` 및를 추가`VolumeWriteOps`한 다음 이러한 지표에 대해 선택한 기간으로 나누어 지표 수학에서 Amazon EBS 볼륨의 IOPS를 그래프로 표시하고 계산할 수 있습니다.

이 예제에서는 기간의 IOPS를 합산한 다음 기간 길이로 나누어 IOPS를 가져옵니다. 그런 다음이 지표 수학 표현식에 대해 경보를 설정하여 볼륨의 IOPS가 볼륨 유형의 최대 용량에 가까워지면 알

림을 보낼 수 있습니다. 지표 수학을 사용하여 CloudWatch 지표를 사용하여 Amazon Elastic File System(Amazon EFS) 파일 시스템을 모니터링하는 방법에 대한 자세한 내용과 예제는 AWS 블로그EFS에서 Amazon CloudWatch 지표 수학을 참조하세요.

CloudWatchContainer Insights 및 CloudWatch Lambda Insights와 함께 Amazon ECS, Amazon EKS 및 Lambda에 자동 대시보드 사용

CloudWatch Container Insights는 Amazon ECS 및 Amazon EKS에서 실행되는 컨테이너 워크로드에 대한 동적 자동 대시보드를 생성합니다. Container Insights를 활성화하여 CPU, 메모리, 디스크, 네트워크 및 컨테이너 재시작 실패와 같은 진단 정보를 관찰할 수 있어야 합니다. Container Insights는 클러스터, 컨테이너 인스턴스 또는 노드, 서비스, 작업, 포드 및 개별 컨테이너 수준에서 빠르게 필터링할 수 있는 동적 대시보드를 생성합니다. Container Insights는 서비스에 따라 클러스터 및 노드 또는 컨테이너 인스턴스 수준에서 구성됩니다. AWS

Container Insights와 마찬가지로 CloudWatch Lambda Insights는 Lambda 함수에 대한 동적 자동 대시보드를 생성합니다. 이 솔루션은 CPU 시간, 메모리, 디스크 및 네트워크를 포함한 시스템 수준 지표를 수집, 집계 및 요약합니다. 또한 콜드 스타트 및 Lambda 작업자 종료와 같은 진단 정보를 수집, 집계 및 요약하여 Lambda 함수 문제를 격리하고 신속하게 해결하는 데 도움이 됩니다. Lambda는 함수 수준에서 활성화되며 에이전트가 필요하지 않습니다.

또한 Container Insights 및 Lambda Insights를 사용하면 애플리케이션 또는 성능 로그, X-Ray 추적 및 서비스 맵으로 빠르게 전환하여 컨테이너 워크로드를 시각화할 수 있습니다. 둘 다 CloudWatch 임베디드 지표 형식을 사용하여 CloudWatch 지표와 성능 로그를 캡처합니다.

Container Insights 및 Lambda Insights에서 캡처한 지표를 사용하는 워크로드에 대한 공유 CloudWatch 대시보드를 생성할 수 있습니다. CloudWatch Container Insights를 통해 자동 대시보드를 필터링하고 확인한 다음 대시보드에 추가 옵션을 선택하여 표시되는 지표를 표준 CloudWatch 대시보드에 추가할 수 있습니다. 그런 다음 지표를 제거하거나 사용자 지정하고 다른 지표를 추가하여 워크로드를 올바르게 나타낼 수 있습니다.

AWS 서비스와 CloudWatch 통합

AWS 는 로깅 및 지표에 대한 추가 구성 옵션을 포함하는 많은 서비스를 제공합니다. 이러한 서비스를 사용하면 로그 출력에 대해 CloudWatch Logs를 구성하고 지표 출력에 대해 CloudWatch 지표를 구성할 수 있습니다. 이러한 서비스를 제공하는 데 사용되는 기본 인프라는에서 AWS 관리하며 액세스할 수 없지만 프로비저닝된 서비스에 대한 로깅 및 지표 옵션을 사용하여 추가 인사이트를 얻고 문제를 해결할 수 있습니다. 예를 들어 VPC 흐름 로그를 CloudWatch에 게시하거나 로그를 CloudWatch에 게시하도록 Amazon Relational Database Service(Amazon RDS) 인스턴스를 구성할 수도 있습니다.

대부분의 AWS 서비스는 통합을 통해 API 호출을 AWS CloudTrail에 로깅합니다. 또한 CloudTrail은 CloudWatch Logs와의 통합을 지원하므로 AWS 서비스에서 활동을 검색하고 분석할 수 있습니다. 또는 Amazon EventBridge를 사용하여 AWS 서비스에서 수행되는 특정 작업에 대한 이벤트 규칙을 사용하여 자동화 및 알림을 생성하고 구성할 수도 있습니다. 특정 서비스는 EventBridge와 직접 통합됩니다. CloudTrail을 통해 전달되는 이벤트를 생성할 수도 있습니다.

대시보드 및 시각화를 위한 Amazon Managed Grafana

Amazon Managed Grafana를 사용하여 AWS 워크로드를 관찰하고 시각화할 수 있습니다. Amazon Managed Grafana를 사용하면 운영 데이터를 대규모로 시각화하고 분석할 수 있습니다. Grafana는 저장된 지표를 쿼리, 시각화, 알림 및 이해하는 데 도움이 되는 오픈 소스 분석 플랫폼입니다. Amazon Managed Grafana는 조직이 기존 워크로드의 시각화에 이미 Grafana를 사용하고 워크로드에 대한 AWS 적용 범위를 확장하려는 경우에 특히 유용합니다. Amazon Managed Grafana를 데이터 소스로 추가하여 CloudWatch와 함께 사용할 수 있습니다. 즉, CloudWatch 지표를 사용하여 시각화를 생성할 수 있습니다. Amazon Managed Grafana는를 지원 AWS Organizations 하며 여러 계정 및 리전의 CloudWatch 지표를 사용하여 대시보드를 중앙 집중화할 수 있습니다.

다음 표에는 대시보드에 CloudWatch 대신 Amazon Managed Grafana를 사용할 때의 이점과 고려 사항이 나와 있습니다. 하이브리드 접근 방식은 최종 사용자, 워크로드 및 애플리케이션의 다양한 요구 사항에 따라 적합할 수 있습니다.

Amazon Managed Grafana 및 오픈 소스 Grafana에서 지원하는 데이터 소스와 통합되는 시각화 및 대시보드 생성	Amazon Managed Grafana를 사용하면 CloudWatch 지표를 비롯한 다양한 데이터 소스에서 시각화 및 대시보드를 생성할 수 있습니다. Amazon Managed Grafana에는 AWS 서비스, 오픈 소스 소프트웨어 및 COTS 소프트웨어에 걸친 여러 기본 제공 데이터 소스가 포함되어 있습니다. 이에 대한 자세한 내용은 Amazon Managed Grafana 설명서의 기본 제공 데이터 소스를 참조하세요. 워크스페이스를 Grafana Enterprise로 업그레이드하여 더 많은 데이터 소스에 대한 지원을 추가할 수도 있습니다. 또한 Grafana는 다양한 외부 시스템과 통신할 수 있는 데이터 소스 플러그인을 지원합니다. CloudWatch 대시보드에서 데이터를 표시하려면 CloudWatch 지표 또는 CloudWatch Logs Insights 쿼리가 필요합니다. CloudWatch
AWS 계정 액세스와 별도로 대시보드 솔루션에 대한 액세스 관리	Amazon Managed Grafana는 인증 및 권한 부여를 AWS Organizations 위해 AWS IAM Identity Center (IAM Identity Center) 및를 사용해야 합니다. 이렇게 하면 IAM Identity Center 또는에

	서 이미 사용할 수 있는 자격 증명 페더레이션을 사용하여 사용자를 Grafana에 인증할 수 있습니다 AWS Organizations. 그러나 IAM Identity Center 또는를 사용하지 않는 경우 Amazon Managed Grafana 설정 프로세스의 일부로 AWS Organizations설정됩니다. 조직에서 IAM Identity Center 또는 사용을 제한한 경우 문제가 될 수 있습니다 AWS Organizations.
AWS Organizations 통합을 통해 여러 계정 및 리전에서 데이터 수집 및 액세스	Amazon Managed Grafana는와 통합되어 모든 계정에서 CloudWatch 및 Amazon OpenSearch Service와 같은 AWS 소스의 데이터를 AWS Organizations 읽을 수 있습니다. 이렇게 하면 계정 전반의 데이터를 사용하여 시각화를 표시하는 대시보드를 생성할 수 있습니다. 간 데이터 액세스를 자동으로 활성화하려면 AWS Organizations 관리 계정에서 Amazon Managed Grafana 워크스페이스를 설정해야 AWS Organizations합니다. AWS Organizations 관리 계정의 모범 사례에 따라 권장되지 않습니다. 반대로 CloudWatch는 CloudWatch 지표에 대한 교차 계정, 교차 리전 대시보드도 지원합니다.
오픈 소스 커뮤니티에서 사용할 수 있는 고급 시각화 위젯 및 Grafana 정의 사용	Grafana는 대시보드를 생성할 때 사용할 수 있는 대규모 시각화 모음을 제공합니다. 또한 요구 사항에 따라 편집하고 재사용할 수 있는 커뮤니티 기여 대시보드의 대규모 라이브러리가 있습니다.
신규 및 기존 Grafana 배포와 함께 대시보드 사용	이미 Grafana를 사용하는 경우 Grafana 배포에서 대시보드를 가져오고 내보내고 Amazon Managed Grafana에서 사용하도록 사용자 지정할 수 있습니다. Amazon Managed Grafana를 사용하면 Grafana를 대시보드 솔루션으로 표준화할 수 있습니다.

워크스페이스, 권한 및 데이터 소스에 대한 고급 설정 및 구성	Amazon Managed Grafana를 사용하면 구성된 자체 데이터 소스, 사용자 및 정책 세트가 있는 여러 Grafana 워크스페이스를 생성할 수 있습니다. 이를 통해 고급 사용 사례 요구 사항과 고급 보안 구성을 충족할 수 있습니다. 고급 기능을 사용하려면 팀이 아직 Grafana에 대한 경험을 높여야 할 수 있습니다.

CloudWatch FAQ를 사용한 로깅 및 모니터링 설계 및 구현

이 섹션에서는 CloudWatch를 사용한 로깅 및 모니터링 솔루션 설계 및 구현에 대해 자주 묻는 질문에 대한 답변을 제공합니다.

CloudWatch 구성 파일은 어디에 저장하나요?

Amazon EC2용 CloudWatch 에이전트는 CloudWatch 구성 디렉터리에 저장된 여러 구성 파일을 적용할 수 있습니다. CloudWatch 구성을 파일 세트로 저장하는 것이 가장 좋습니다. 여러 계정 및 환경에서 버전 관리를 하고 다시 사용할 수 있기 때문입니다. 이에 대한 자세한 내용은이 가이드의 CloudWatch 구성 관리 섹션을 참조하세요. 또는 GitHub의 리포지토리에 구성 파일을 저장하고 새 EC2 인스턴스가 프로비저닝될 때 구성 파일 검색을 자동화할 수 있습니다.

경보가 발생할 때 서비스 관리 솔루션에서 티켓을 생성하려면 어떻게 해야 합니까?

서비스 관리 시스템을 Amazon Simple Notification Service(Amazon SNS) 주제와 통합하고 경보가 발생하면 SNS 주제에 알리도록 CloudWatch 경보를 구성합니다. 통합 시스템은 SNS 메시지를 수신하고 서비스 관리 시스템 APIs 또는 SDKs를 사용하여 티켓을 생성할 수 있습니다.

CloudWatch를 사용하여 컨테이너의 로그 파일을 캡처하려면 어떻게 해야 하나요?

Amazon ECS 작업 및 Amazon EKS 포드는 STDOUT 및 STDERR 출력을 CloudWatch로 자동으로 전송하도록 구성할 수 있습니다. 컨테이너화된 애플리케이션을 로깅하는 데 권장되는 접근 방식은 컨테이너가 출력을 STDOUT 및 STDERR로 전송하도록 하는 것입니다. 이는 Twelve-Factor 앱 매니페스토에서도 다룹니다.

그러나 특정 로그 파일을 CloudWatch로 전송하려는 경우 Amazon EKS 포드 또는 Amazon ECS 작업 정의에 볼륨을 탑재하여 애플리케이션이 해당 로트 파일을 작성하고 Fluentd 또는 Fluent Bit용 사이드카 컨테이너를 사용하여 로그를 CloudWatch로 전송할 수 있습니다. 컨테이너의 특정 로그 파일을 `/dev/stdout` 및에 연결하는 심볼을 고려해야 합니다`/dev/stderr`. 이에 대한 자세한 내용은 Docker 설명서의 컨테이너 또는 서비스에 대한 로그 보기를 참조하세요.

AWS 서비스의 상태 문제를 모니터링하려면 어떻게 해야 합니까?

AWS Health Dashboard를 사용하여 AWS 상태 이벤트를 모니터링할 수 있습니다. 상태 이벤트와 관련된 샘플 자동화 솔루션은 aws-health-tools GitHub 리포지토리를 참조할 AWS 수도 있습니다.

에이전트 지원이 없는 경우 사용자 지정 CloudWatch 지표를 생성하려면 어떻게 해야 합니까?

임베디드 지표 형식을 사용하여 CloudWatch에 지표를 수집할 수 있습니다. AWS SDK(예: put_metric_data), AWS CLI (예: put-metric-data) 또는 AWS API(예: PutMetricData)를 사용하여 사용자 지정 지표를 생성할 수도 있습니다. 사용자 지정 로직을 장기간 유지하는 방법을 고려해야 합니다. 한 가지 접근 방식은 Lambda를 내장된 지표 형식 지원과 함께 사용하여 지표를 생성하고 CloudWatch Events 이벤트 일정 규칙을 사용하여 지표 기간을 설정하는 것입니다.

기존 로깅 및 모니터링 도구를와 통합하려면 어떻게 해야 하나요 AWS?

와 통합하려면 소프트웨어 또는 서비스 공급업체가 제공하는 지침을 참조해야 합니다 AWS. 에이전트 소프트웨어, SDK 또는 제공된 API를 사용하여 로그와 지표를 솔루션에 전송할 수 있습니다. 공급업체 사양에 맞게 구성된 Fluentd 또는 Fluent Bit와 같은 오픈 소스 솔루션을 사용할 수도 있습니다. Lambda 및 Kinesis Data Streams와 AWS 함께 SDK 및 CloudWatch Logs 구독 필터를 사용하여 사용자 지정 로그 프로세서 및 송하인을 생성할 수도 있습니다. 마지막으로 여러 계정 및 리전을 사용하는 경우 소프트웨어를 통합하는 방법도 고려해야 합니다.

리소스

소개

- AWS Well-Architected

목표 비즈니스 성과

- logging-monitoring-apg-guide-examples
- 클라우드 컴퓨팅의 6가지 이점

CloudWatch 배포 계획

- AWS Organizations 용어 및 개념
- AWS Systems Manager 빠른 설정
- CloudWatch 에이전트를 사용하여 Amazon EC2 인스턴스 및 온프레미스 서버에서 지표 및 로그 수집
- cloudwatch-config-s3-bucket.yaml
- 마법사를 사용하여 CloudWatch 에이전트 구성 파일 생성
- Enterprise DevOps: 빌드를 실행해야 하는 이유
- Exporting log data to Amazon S3
- Amazon OpenSearch Service의 세분화된 액세스 제어
- Lambda 할당량
- 수동으로 CloudWatch 에이전트 구성 파일 생성 또는 편집
- 구독을 통한 로그 데이터 실시간 처리
- 빌드할 도구 AWS

EC2 인스턴스 및 온프레미스 서버에 대한 CloudWatch 에이전트 구성

- Amazon EC2 지표 차원

- 성능 버스트 가능 인스턴스
- CloudWatch 에이전트 사전 정의 지표 세트
- procstat 플러그인을 사용하여 프로세스 지표 수집
- procstat용 CloudWatch 에이전트 구성
- EC2 인스턴스에 대한 세부 모니터링 관리
- CloudWatch 임베디드 지표 형식을 사용하여 높은 카디널리티 로그 수집 및 지표 생성
- 로그 그룹 및 로그 스트림 작업
- 인스턴스에 사용 가능한 CloudWatch 지표 나열
- PutLogEvents
- collectd를 사용하여 사용자 지정 지표 검색
- StatsD를 사용하여 사용자 지정 지표 검색

Amazon EC2 및 온프레미스 서버에 대한 CloudWatch 에이전트 설치 접근 방식

- 하이브리드 및 멀티클라우드 환경에서 Systems Manager에 필요한 IAM 서비스 역할 생성
- 하이브리드 환경에 대한 관리형 인스턴스 활성화 생성
- CloudWatch 에이전트와 함께 사용할 IAM 역할 및 사용자 생성
- 명령줄을 사용하여 CloudWatch 에이전트 다운로드 및 구성
- Systems Manager 에이전트와 통합 CloudWatch 에이전트를 사용하여 임시 자격 증명만 사용하도록 온프레미스 서버를 구성하려면 어떻게 해야 하나요?
- 스택 세트 작업을 위한 사전 조건
- 스팟 인스턴스 사용

Amazon ECS에서 로깅 및 모니터링

- amazon-cloudwatch-logs-for-fluent-bit
- Amazon ECS CloudWatch 지표
- Amazon ECS Container Insights 지표
- Amazon ECS 컨테이너 에이전트
- Amazon ECS 시작 유형

- CloudWatch 에이전트를 배포하여 Amazon ECS에서 EC2 인스턴스 수준 지표 수집
- ecs_cluster_with_cloudwatch_linux.yaml
- ecs_cw_emf_example
- ecs_firelense_emf_example
- ecs-task-nginx-firelense.json
- Amazon ECS 최적화 AMI 메타데이터 검색
- awslogs 로그 드라이버 사용
- 클라이언트 라이브러리를 사용하여 임베디드 지표 형식 로그 생성

Amazon EKS의 로깅 및 모니터링

- Amazon EKS 컨트롤 영역 로깅
- amazon_eks_managed_node_group_launch_config.yaml
- Amazon EKS 노드
- amazon-eks-nodegroup.yaml
- Amazon EKS 서비스 수준 계약
- Container Insights Prometheus 지표 모니터링
- Prometheus를 사용한 컨트롤 플레인 지표
- Fargate 로깅
- Fargate의 Amazon EKS용 Fluent Bit
- Fargate에서 Amazon EKS를 사용할 때 애플리케이션 로그를 캡처하는 방법
- Prometheus 지표를 수집하기 위한 CloudWatch 에이전트 설치
- Kubernetes 지표 서버 설치
- kubernetes/대시보드
- Kubernetes Horizontal Pod Autoscaler
- Kubernetes 컨트롤 플레인 구성 요소
- Kubernetes 포드
- 시작 템플릿 지원
- 관리형 노드 그룹
- 관리형 노드 업데이트 동작
- 지표-서버

- Prometheus 및 Grafana를 사용하여 Fargate에서 Amazon EKS 모니터링
- prometheus_jmx
- prometheus/jmx_exporter
- 추가 Prometheus 소스 스크레이핑 및 해당 지표 가져오기
- 자체 관리형 노드
- CloudWatch Logs로 로그 전송
- FluentD를 DaemonSet로 설정하여 CloudWatch Logs로 로그 전송
- Amazon EKS 및 Kubernetes에서 Java/JMX 샘플 워크로드 설정
- 새 Prometheus 스크레이프 대상을 추가하기 위한 자습서: Prometheus API Server 지표
- Vertical Pod Autoscaler

에 대한 로깅 및 지표 AWS Lambda

- Lambda 호출 오류
- 로깅 - Python의 로깅 기능
- 클라이언트 라이브러리를 사용하여 임베디드 지표 형식 로그 생성
- Lambda 함수 지표 작업

Searching and analyzing logs in CloudWatch

- Beats 패밀리
- Elastic Logstash
- Elastic 스택
- CloudWatch Logs 데이터를 Amazon OpenSearch Service로 스트리밍

CloudWatch를 사용한 경보 옵션

- amazon-cloudwatch-auto-alarms
- AWS Jira Service Management Cloud용 서비스 관리 커넥터
- AWS Jira 서비스 관리 데이터 센터용 서비스 관리 커넥터
- AWS ServiceNow 관리 커넥터

애플리케이션 및 서비스 가용성 모니터링

- DNS 장애 조치 구성

를 사용하여 애플리케이션 추적 AWS X-Ray

- Amazon ECS 작업 네트워킹
- X-Ray 콘솔에서 샘플링 규칙 구성
- Windows PowerShell 명령 또는 스크립트 실행
- Amazon EC2에서 X-Ray 데몬 실행
- 추적 데이터를 X-Ray로 전송
- X-Ray의 서비스 그래프

Dashboards and visualizations with CloudWatch

- Amazon CloudWatch 지표 수학은 Amazon EFS 파일 시스템의 실시간에 가까운 모니터링을 간소화합니다.
- CloudWatch Container Insights 설정
- 지표 수학 사용

CloudWatch와 AWS 서비스 통합

- AWS CloudTrail 지원되는 서비스 및 통합
- Amazon EventBridge AWS 서비스 의 이벤트
- 를 통해 전달되는 AWS 서비스 이벤트 AWS CloudTrail
- CloudWatch Logs를 사용하여 CloudTrail 로그 파일 모니터링 CloudWatch
- CloudWatch Logs에 데이터베이스 로그 게시
- CloudWatch Logs에 흐름 로그 게시

대시보드 및 시각화를 위한 Amazon Managed Grafana

- 의 관리 계정 모범 사례 AWS Organizations

- Amazon Managed Grafana용 기본 제공 데이터 소스
- CloudWatch의 교차 계정 및 교차 리전 대시보드
- Grafana 플러그인

문서 기록

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

변경 사항	설명	날짜
업데이트된 로깅 정보	로깅에 대한 섹션을 업데이트했습니다 AWS Lambda.	2023년 4월 17일
업데이트된 구성 정보	CloudWatch 구성 생성 및 저장에 대한 섹션을 업데이트하고 이름을 변경했습니다.	2023년 2월 9일
지표 정보 업데이트	Amazon ECS에 대한 지표 섹션에서 사용자 지정 애플리케이션 지표 정보를 업데이트했습니다.	2023년 1월 31일
미리 보기 알림 제거	Amazon Managed Grafana를 정식 버전으로 사용할 수 있습니다.	2022년 5월 25일
삭제된 섹션	CloudWatch SDK 지표가 더 이상 지원되지 않습니다.	2022년 1월 7일
최초 게시	—	2021년 4월 30일

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 취약성을 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 LLM에 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 프롬프팅도 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.