



Running Containers on Amazon EKS

모듈 7: Amazon EKS 클러스터에서 관찰 기능 구성



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.



| 수강생용 노트

이 모듈은 Amazon EKS 클러스터에서 관찰 기능 구성입니다.



Running Containers on Amazon EKS

모듈 7 개요

- Amazon EKS 클러스터에서 관찰 가능성 구성
- 지표 수집
- 로그 관리
- Amazon EKS에서 애플리케이션 추적
- 실습 4 준비



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Running Containers on Amazon EKS

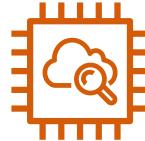
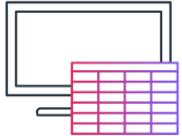
Amazon EKS

클러스터에서 관찰 가능성 구성



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

관찰 가능성



애플리케이션

Pod 및 컨테이너

컴퓨팅 리소스

관찰 가능성은 시스템이 수행하는 작업을 통해 시스템을 얼마나 잘 이해할 수 있는지 나타내는 척도이며 시스템을 개선하는 기반이 됩니다.



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

| 수강생용 노트

관찰 가능성이란 데이터 또는 프로세스를 분석하고 볼 수 있는 기능입니다. 관찰 가능성은 모니터링 데이터(예: 지표)가 컴파일된 후에만 달성됩니다. 관찰 가능성은 종종 모니터링과 같은 의미로 사용되지만 두 개념은 서로 다릅니다.

쉽지 않은 인사이트 확보

- 마이크로서비스 아키텍처
 - 다양한 지표
 - 복잡한 상호 작용
- 컨테이너
 - 일시적 - 중요한 증거가 컨테이너와 함께 사라집니다.
 - 운영 체제는 하나의 요소일 뿐입니다.



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

| 수강생용 노트

모니터링도 중요하지만 클러스터 상태에 대한 실행 가능한 인사이트를 얻는 것이 진정한 과제입니다. 어떤 환경에서든 인사이트를 얻기는 어렵습니다. 하지만 선택할 수 있는 많은 잠재적 지표가 있는 마이크로서비스 환경에서는 더욱 어렵습니다. 올바른 지표를 선택하는 것은 어려울 수 있습니다.

특히 컨테이너는 일반적으로 일반 워크로드보다 일시적이기 때문에 모니터링하기가 어렵습니다. 실행 중인 컨테이너에서 로그 및 기타 유용한 아티팩트를 캡처할 수 있는 프로세스가 있어야 합니다. 그런 다음 이러한 로그와 아티팩트를 내구성 있고 검색 가능한 위치에 저장해야 합니다.

인사이트의 가치



고객 경험

성능 및 비용

추세

문제 해결 및 수정

학습 및 개선



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

| 수강생용 노트

클러스터에 대한 좋은 인사이트가 있다면 다음과 같은 중요한 질문에 답할 수 있습니다.

- 고객 경험
 - 애플리케이션이 최고의 경험을 제공합니까?
- 성능 및 비용
 - 변경 사항이 전반적인 성능과 비용에 어떠한 영향을 미칩니다?
- 추세
 - 크기 조정이 필요합니까?
- 문제 해결 및 수정
 - 문제가 어디에서 발생했으며 신속하게 해결하려면 어떻게 해야 합니까?
- 학습 및 개선
 - 향후 문제를 감지하고 예방하려면 어떻게 해야 합니까?

3개 핵심 요소



지표



로그



추적



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

| 수강생용 노트

관찰 가능성은 애플리케이션, 컨테이너/오케스트레이션, 물리적 호스트 등 스택의 3가지 계층 모두에서 실행 가능한 정보를 수집할 수 있는 기능입니다.

완전한 관찰 가능성은 3가지 주요 소스에서 비롯됩니다.

- 지표 - 시간 간격을 두고 측정되는 리소스의 상태와 성능에 관한 데이터를 수집하고 시각화하는 기능입니다. 또한 핵심 성과 지표가 범위를 벗어날 때 경고하도록 경고를 구성하는 기능도 바람직합니다.
- 로그 - 리소스에서 로그 파일을 수집하고 집계하고 배경 노이즈에서 실행 가능한 인사이트를 필터링하는 기능입니다.
- 추적 - 요청이 여러 서비스를 통과할 때 요청의 경로를 추적하는 기능입니다. 추적은 개발자가 애플리케이션과 그 기본 서비스의 성능을 이해하는데 도움이 됩니다. 이러한 추적은 성능 문제 및 오류의 근본 원인을 파악하여 문제를 해결하는데 도움이 됩니다.

관찰 가능성 도구

지표

로깅

추적



Amazon Managed Service for Prometheus



Amazon CloudWatch



Amazon OpenSearch Service



Amazon CloudWatch



AWS X-Ray



Amazon OpenSearch Service



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

| 수강생용 노트

관찰 가능성을 구현하는 방법을 선택하는 데는 관찰 가능성 요구 사항을 충족하는 적절한 도구를 선택하기 위한 신중한 고려가 필요합니다. 지표, 로깅, 추적에 매우 다양한 도구를 사용할 수 있습니다. 적절한 도구를 선택하는 기준에는 현재의 도구 전문 지식, 조직 요구 사항 및 팀 문화가 포함됩니다. 다음은 일반적으로 사용되는 몇 가지 AWS 서비스입니다.

- **지표:** Amazon Managed Service for Prometheus, Amazon CloudWatch
- **로깅:** Amazon OpenSearch Service, Amazon CloudWatch
- **추적:** AWS X-ray, Amazon OpenSearch Service

이 모듈에서는 이러한 관리형 서비스 각각에 대해 설명합니다. **Prometheus** 및 **OpenSearch**의 오픈 소스 대안은 셀프 매니지드 옵션을 위한 관찰 가능성 도구입니다.



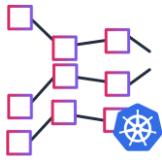
Running Containers on Amazon EKS

자료 수집



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Prometheus



Prometheus 지표
(지표 형식)

Prometheus
(오픈 소스 도구)

PromQL
(쿼리 언어)



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

| 수강생용 노트

Prometheus라는 용어는 일반적으로 제어 영역 지표와 관련되어 있지만 사람들은 종종 이 용어를 다른 것을 지칭하는 데 사용합니다.

지표 형식

Prometheus 지표는 시계열로 모니터링하고 분석하는데 유용합니다. **Prometheus** 지표 형식은 일부 다른 컨테이너화된 워크로드 및 시스템에서도 사용됩니다.

Prometheus 형식은 Internet Engineering 과제 Force(IETF)로 형식을 표준화하려는 별도의 Cloud Native Computing Foundation(CNCF) 인큐베이팅 프로젝트인 **OpenMetrics**의 기반입니다. 자세한 내용은 GitHub 웹

사이트(<https://github.com/OpenObservability/OpenMetrics/blob/main/specification/OpenMetrics.md>)의 'OpenMetrics'를 참조하십시오.

오픈 소스 도구

Prometheus는 다양한 소스에서 지표를 수집하고 저장하는 데 사용되는 오픈 소스 시계열 데이터베이스의 이름이기도 합니다. **Prometheus** 프로젝트는 **Kubernetes**처럼 CNCF의 출입 단계 프로젝트이며 인기 있는 솔루션입니다. **Prometheus** 프로젝트에 대한 자세한 내용은 <https://prometheus.io/>에서 온라인으로 확인할 수 있습니다.

Prometheus 쿼리 언어

Prometheus 도구 사용자는 **Prometheus** 쿼리 언어(종종 **PromQL**로 약칭)를 사용하여 제어 영역 지표를 집계하고 시각화하는 식을 생성합니다.

예: 제어 영역 지표

Prometheus 형식: metric_name{"tag"="value"}[,...] value

```
$ kubectl get --raw /metrics
...
# HELP rest_client_requests_total Number of HTTP requests, partitioned by status
# code, method, and host.
# TYPE rest_client_requests_total counter
rest_client_requests_total{code="200",host="127.0.0.1:21362",method="POST"} 4994
rest_client_requests_total{code="200",host="127.0.0.1:443",method="DELETE"} 1
rest_client_requests_total{code="200",host="127.0.0.1:443",method="PUT"} 862173
rest_client_requests_total{code="404",host="127.0.0.1:443",method="GET"} 2
rest_client_requests_total{code="409",host="127.0.0.1:443",method="POST"} 3
rest_client_requests_total{code="409",host="127.0.0.1:443",method="PUT"} 8
# HELP ssh_tunnel_open_fail_count Counter of ssh tunnel failed open attempts
# TYPE ssh_tunnel_open_fail_count counter
ssh_tunnel_open_fail_count 0
...
```



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

| 수강생용 노트

Kubernetes API 서버는 모니터링 및 분석에 유용한 여러 지표를 노출합니다. 이러한 지표는 `/metrics` HTTP API를 참조하는 지표 엔드포인트를 통해 내부적으로 노출됩니다. 다른 엔드포인트와 마찬가지로 이 엔드포인트는 Amazon EKS 제어 영역에 노출됩니다. 기본 Kubernetes 도구를 사용하거나 Prometheus 또는 Amazon CloudWatch와 같은 다른 도구를 사용하여 지표를 수집할 수 있습니다.

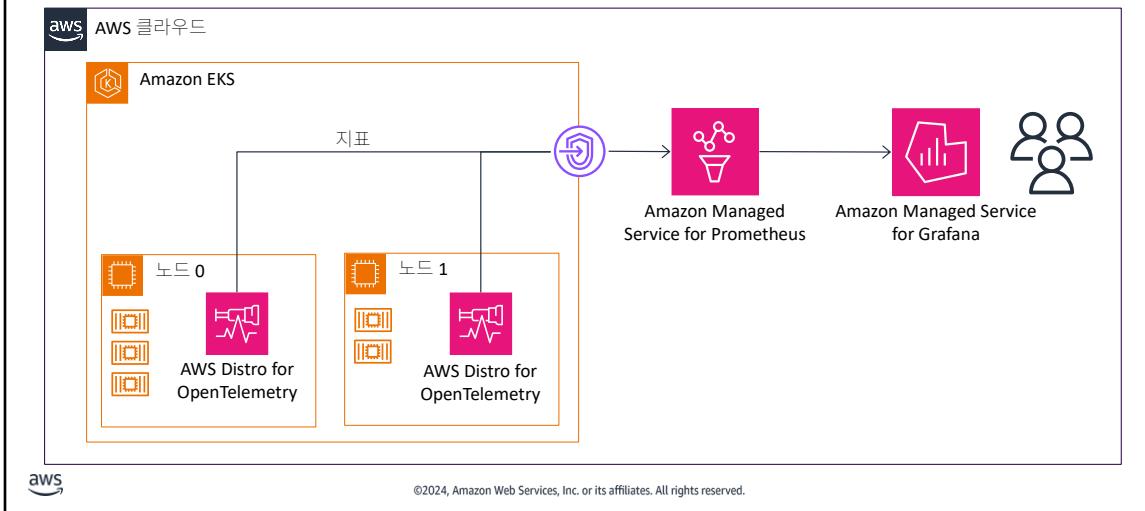
이것은 `kubectl`을 사용하여 원시 지표 출력을 수집하는 예입니다. 이 명령을 사용하여 HTTP 경로를 전달하고 API 서버가 Prometheus 형식으로 노출하는 그대로의 원시 응답을 반환합니다. 이 형식을 사용하면 API 서버에서 각 출력 블록을 해석하는데 도움이 되는 주석과 함께 다양한 지표를 줄에 따라 세분화하여 표시할 수 있습니다. 각 줄에는 지표 이름, 태그(선택 사항), 값이 포함됩니다.

이 예에서는 두 개의 서로 다른 지표가 표시됩니다. 첫 번째로 **HTTP** 요청 수에는 코드, 호스트, 메서드의 3가지 태그가 각 줄에 있습니다. 두 번째 지표인 **ssh** 터널 실패한 열기 시도 횟수 카운터에는 지표 이름과 값만 표시됩니다.

원시 지표의 더 많은 예는 GitHub 웹

사이트(https://github.com/prometheus/docs/blob/master/content/docs/instrumenting/exposition_formats.md)의 'Exposition formats'를 참조하십시오.

Prometheus 및 Grafana를 사용하여 지표 모니터링 및 시각화



~ALT text

~클러스터: Amazon Managed Service for Grafana에 연결되는 여러 노드가 있는 Amazon EKS 클러스터입니다. 자세한 내용은 노트를 참조하십시오.

~화살표: AWS Distro for OpenTelemetry가 있는 각 노드에서 Amazon Managed Service for Prometheus로 가는 화살표입니다.

|수강생용 노트

이 슬라이드에는 2개의 노드가 있는 Amazon EKS 클러스터를 보여주는 아키텍처 그림이 포함되어 있습니다. 각 노드에는 ADOT Collector가 설치되어 있습니다.

데이터는 ADOT 수집기에서 Amazon Managed Service for Prometheus로, 거기에서 Amazon Managed Grafana로 흐릅니다.

오픈 소스 도구인 Grafana와 함께 Prometheus를 사용하여 지표를 시각화하고 쿼리할 수 있습니다. Grafana에 대한 자세한 내용은 <https://grafana.com/>에서 온라인으로 확인할 수 있습니다.

이 예에서는 AWS Distro for OpenTelemetry, Amazon Managed Service for Prometheus, Amazon Managed Service for Grafana를 사용하여 EKS 클러스터 지표를 시각화하는 방법을 보여줍니다.

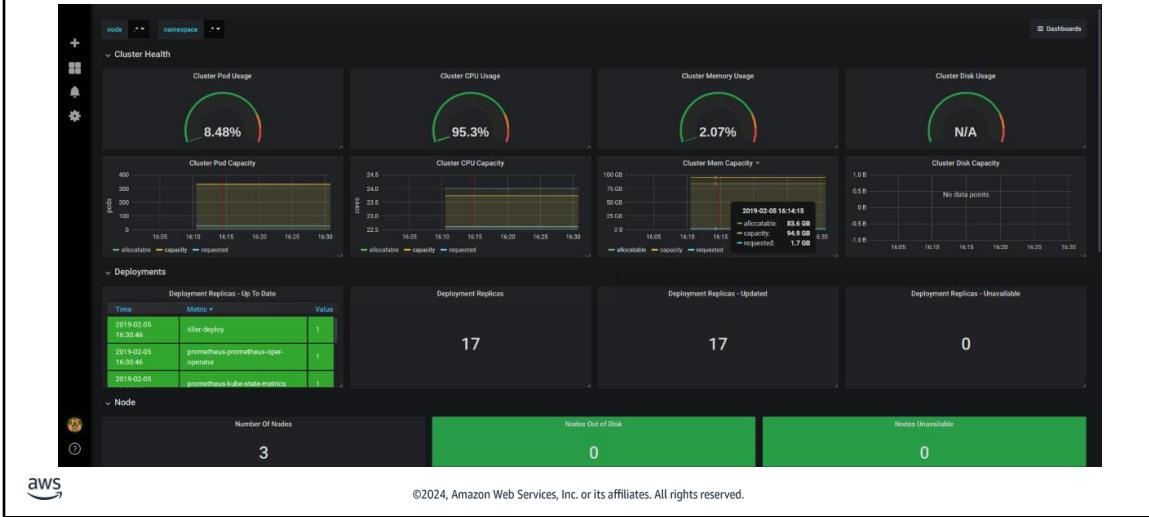
AWS Distro for OpenTelemetry는 CNCF OpenTelemetry 프로젝트의 안전한 프로덕션 준비 AWS 지원 배포입니다. OpenTelemetry는 오픈 소스 API, 라이브러리, 에이전트를 제공하여 애플리케이션 모니터링을 위한 분산 추적 및 지표를 수집합니다. AWS Distro for OpenTelemetry에 대한 자세한 내용은 <https://aws.amazon.com/otel>에서 온라인으로 확인할 수 있습니다.

AWS Distro for OpenTelemetry, Amazon Managed Service for Prometheus, Amazon Managed Service for Grafana 가 이제 정식 출시되었습니다. 대안으로 자체 Prometheus 및 Grafana 서버를 배포하여 셀프 매니지드 솔루션을 구축할 수 있습니다.

AWS Distro for OpenTelemetry를 사용하여 지표를 수집하는 예는 다음 블로그 게시물:
(<https://aws.amazon.com/blogs/opensource/aws-distro-for-opentelemetry-is-now-generally-available-for-metrics/>)을 참조하십시오.

또는, Prometheus는 Amazon Managed Service for Prometheus를 통해 에이전트 없는 수집을 사용할 수 있습니다. 자세한 내용은 <https://aws.amazon.com/blogs/aws/amazon-managed-service-for-prometheus-collector-provides-agentless-metric-collection-for-amazon-eks/>를 참조하십시오.

Grafana 대시보드



~ALT text

~Grafana 대시보드의 스크린샷입니다.

| 수강생용 노트

Grafana는 대시보드를 사용하여 데이터를 시각화할 수 있습니다. 대시보드에는 데이터의 다양한 측면을 보여주는 다양한 패널이 포함될 수 있으며 데이터는 여러 소스로부터 올 수 있습니다.

대시보드는 클러스터 Pod 용량과 같은 Pod 관련 데이터를 모니터링합니다.

Amazon CloudWatch Container Insights



Amazon
CloudWatch



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

- 지표 및 로그 수집
- 클러스터, 노드, Pod, 태스크, 서비스 수준에서 집계
- 대시보드에서 지표 시각화
 - CPU, 메모리, 디스크, 네트워크
 - 컨테이너 데이터
- 지표에 대한 CloudWatch 경보 설정

| 수강생용 노트

원시 지표에서는 실행 가능한 인사이트를 얻기가 어렵습니다. Prometheus 프로젝트 외에도 지표를 수집하고 시각화하는데 사용할 수 있는 솔루션은 Grafana와 같은 오픈 소스 도구부터 Datadog, Splunk, SumoLogic과 같은 APN 파트너에 이르기까지 다양합니다. 이 예는 Amazon CloudWatch Container Insights를 사용합니다.

CloudWatch Container Insights

CloudWatch Container Insights는 컨테이너화된 애플리케이션과 마이크로서비스에서 지표와 로그를 수집하고 집계하며 요약합니다. 이 지표에는 CPU, 메모리, 디스크, 네트워크 같은 리소스 사용률이 포함되어 있습니다. 또한 Container Insights는 컨테이너 재시작 오류 같은 진단 정보를 제공하여 문제를 격리하고 신속하게 해결할 수 있도록 도와줍니다. Container Insights가 수집하는 지표는 CloudWatch 자동 대시보드에서 확인할 수 있습니다. Container Insights가 수집하는 지표에 CloudWatch 경보를 설정할 수도 있습니다.

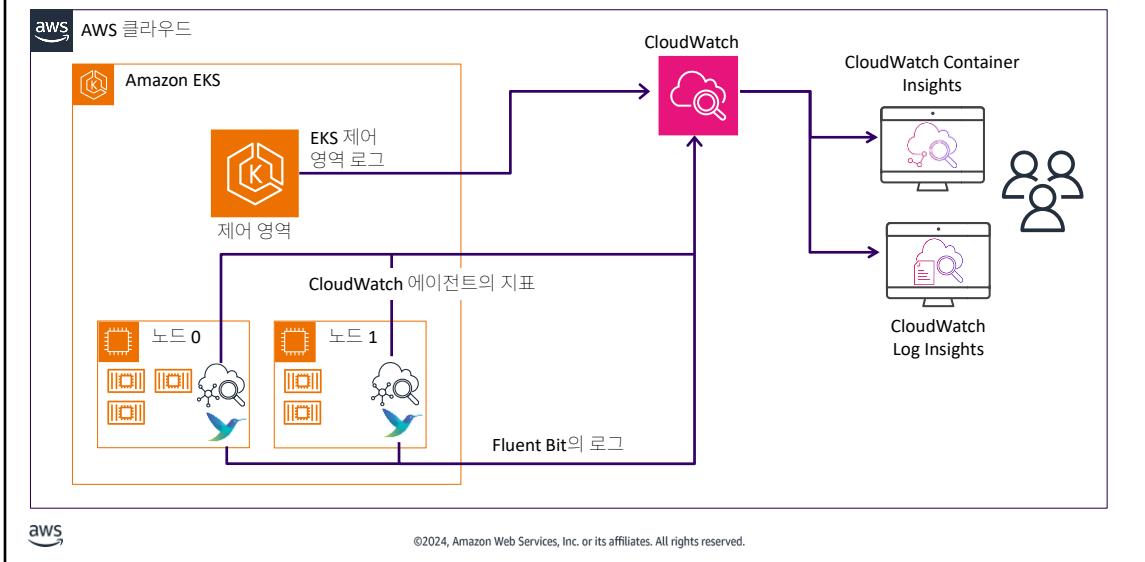
Amazon EKS와 Kubernetes에서 Container Insights는 컨테이너화된 버전의 CloudWatch 에이전트를 사용하여 클러스터에서 실행 중인 모든 컨테이너를 검색합니다. 그런 다음 성능 스택의 모든 계층에서 성능 데이터를 수집합니다.

CloudWatch는 성능 로그 데이터를 수집하여 집계된 지표를 생성합니다. CloudWatch Logs Insights를 사용하여 컨테이너 성능을 분석하고 문제를 해결하고 데이터를 기록할 수 있습니다.

로그 데이터 암호화

Container Insights는 수집하는 로그 및 지표에 대해 고객 마스터 키(CMK)를 사용한 암호화를 지원합니다. 이 암호화를 사용하도록 설정하려면 **Container Insights** 데이터를 수신하는 로그 그룹에 대해 **AWS Key Management Service(AWS KMS)** 암호화를 수동으로 사용하도록 설정해야 합니다. 그 결과 **Container Insights**는 제공된 CMK를 사용하여 이 데이터를 암호화합니다. 대칭 CMK만 지원됩니다. 로그 그룹을 암호화할 때 비대칭 CMK를 사용하지 마십시오.

CloudWatch 지표 및 로그 수집



~ALT text:

~CloudWatch에 대한 화살표: Fluent Bit에서 로그를 수집하는 과정과 Amazon EKS 클러스터 내의 각 노드에 대한 CloudWatch 에이전트에서 지표를 수집하는 과정을 보여줍니다.

~클러스터: 제어 영역 로그가 있는 Amazon EKS 클러스터와 Fluent Bit 및 CloudWatch 에이전트 지표가 있는 여러 노드입니다.

~인사이트에 대한 화살표: CloudWatch의 정보는 CloudWatch Container Insights 및 CloudWatch Log Insights에서 볼 수 있습니다.

| 강사용 노트 및 애니메이션

|<1> 화살표는 CloudWatch 에이전트의 지표와 CloudWatch가 수집하는 Fluent Bit의 로그 프로세스를 보여줍니다.

|<2> 수집된 정보는 CloudWatch Container Insights 및 CloudWatch Log Insights에서 볼 수 있습니다.

| 수강생용 노트

Amazon EKS에서 지표를 수집, 집계, 시각화하도록 CloudWatch를 구성하는 방법의 예입니다. CloudWatch 에이전트는 각 노드에서 DaemonSet로 실행됩니다.

CloudWatch 플러그인이 있는 로그 수집기는 모든 노드에서 DaemonSet로 실행됩니다. 이 예에서는 오픈 소스 로그 수집 및 집계 도구인 Fluent Bit를 사용합니다. 이 도구는 이 모듈 뒷부분에서 자세히 설명합니다.

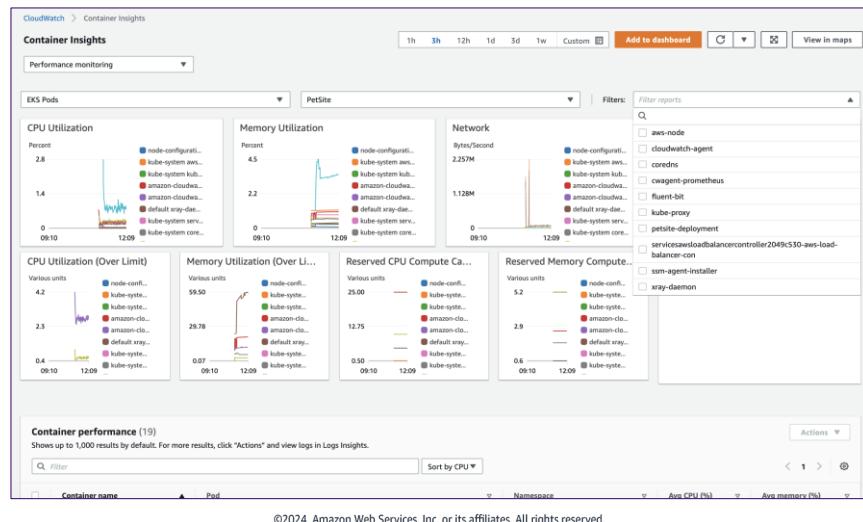
Amazon EKS 클러스터에 대한 제어 영역 로깅을 사용하여 제어 영역 지표를 수집할 수 있습니다. CloudWatch는 CloudWatch Log Insights를 사용하여 볼 수 있는 이러한 로그에서 지표 정보를 수집합니다.

클러스터의 노드에서 지표를 수집할 수도 있습니다.

- **Container Insights**는 컨테이너식 버전의 **CloudWatch** 에이전트를 사용하여 클러스터에서 실행 중인 모든 컨테이너를 검색합니다. 그런 다음 성능 스택의 모든 계층에서 성능 데이터를 수집합니다. 이 지표에는 **CPU**, 메모리, 디스크, 네트워크 같은 리소스 사용률이 포함되어 있습니다. 또한 **Container Insights**는 컨테이너 재시작 오류 같은 진단 정보를 제공하여 문제를 격리하고 신속하게 해결할 수 있도록 도와줍니다. **Container Insights**가 수집하는 지표에 **CloudWatch** 경보를 설정할 수도 있습니다.
- **Container Insights**는 노드에서 로그 파일을 수집할 수도 있습니다. 빠른 시작 방법을 사용하여 **CloudWatch Logs**로 로그를 보내도록 **Container Insights**를 설정하면 **Fluent Bit** 및 **CloudWatch** 에이전트가 클러스터의 모든 노드에 배포됩니다.

****접근성을 위한 설명:** 이 슬라이드에는 2개의 노드가 있는 **Amazon EKS** 클러스터를 보여주는 아키텍처 그림이 포함되어 있습니다. 각 노드에는 **CloudWatch** 에이전트와 **Fluent Bit**가 설치되어 있습니다. 지표 데이터는 **CloudWatch** 에이전트에서 **CloudWatch**로 흐르고 여기서 **Container Insights**를 사용하여 볼 수 있습니다. 애플리케이션 로그는 **Fluent Bit**에서 **CloudWatch**로 흐르고 **CloudWatch Log Insights**를 사용하여 볼 수 있습니다. 설명 끝.

Container Insights에서 클러스터 전체 지표 보기



~Alt text

~CloudWatch Container Insights 콘솔입니다. 자세한 내용은 노트에 설명되어 있습니다.

~

| 수강생용 노트

Container Insights를 설정하고 지표를 수집한 후 CloudWatch 콘솔에서 해당 지표를 볼 수 있습니다. CloudWatch 콘솔을 열고 **Container Insights**, 리소스로 이동하면 시각적(맵 보기) 또는 테이블 형식(목록 보기) 표시가 표시됩니다. 이 슬라이드에서는 CloudWatch Container Insights 콘솔의 성능 모니터링 보기를 보여줍니다. 필터가 설정되지 않아 클러스터 수준 지표가 표시되어 있습니다.

Container Insights가 Amazon EKS 및 Kubernetes를 위해 수집하는 지표에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 'Amazon EKS 및 Kubernetes Container Insights'

지표'(<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/Container-Insights-metrics-EKS.html>)를 참조하십시오.

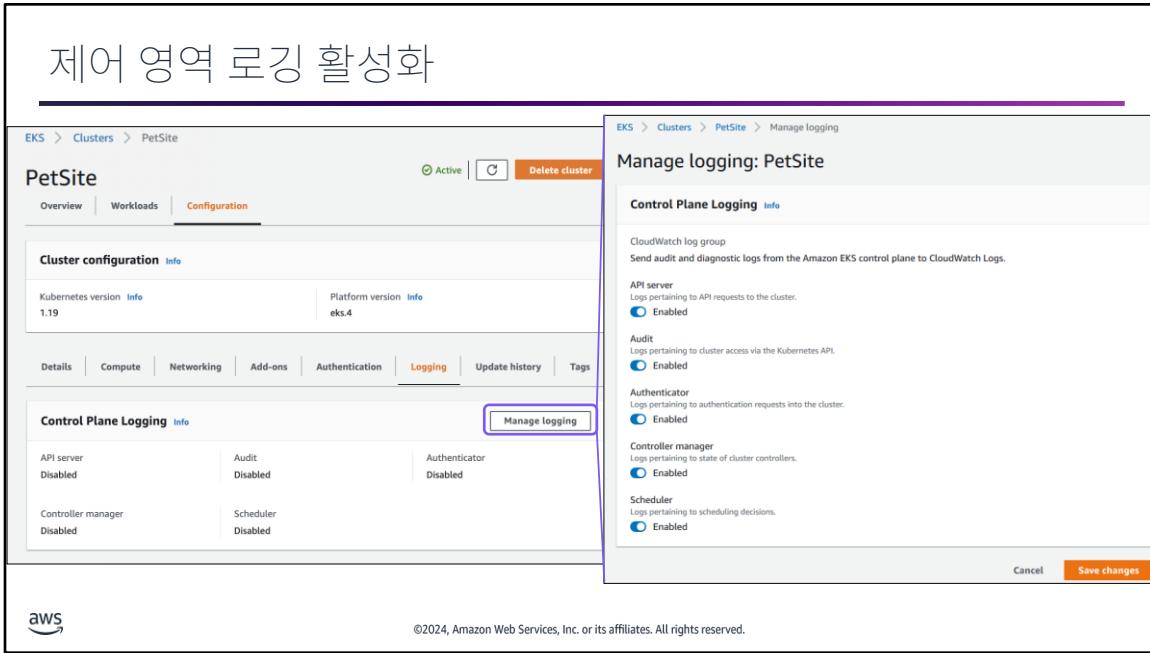


Running Containers on Amazon EKS

로그 관리



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.



~Alt text

~로깅 관리 버튼이 강조 표시된 클러스터의 세부 정보입니다.

~로깅 관리 버튼을 선택하여 액세스하는 제어 영역 로깅 설정입니다. 모든 로그가 활성화되었습니다.

~

|수강생용 노트

Amazon EKS 제어 영역 로깅을 활성화하여 감사 및 진단 로그를 Amazon EKS 제어 영역에서 CloudWatch Logs로 직접 제공할 수 있습니다. 필요한 로그 유형을 정확하게 선택할 수 있습니다. 그런 다음 로그는 CloudWatch의 각 Amazon EKS 클러스터에 대한 그룹으로 로그 스트림으로 전송됩니다.

다음 클러스터 제어 영역 로그 유형을 사용할 수 있습니다. 각 로그 유형은 Kubernetes 제어 영역의 구성 요소에 해당합니다.

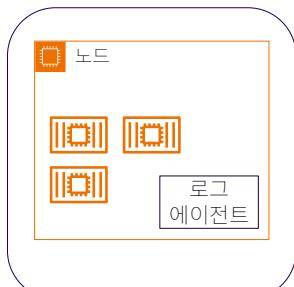
- **Kubernetes API** 서버 구성 요소 로그(**API**) - 클러스터의 API 서버는 Kubernetes API를 노출하는 제어 영역 구성 요소입니다.
- 감사(**audit**) - Kubernetes 감사 로그는 클러스터에 영향을 미친 개별 사용자, 관리자 또는 시스템 구성 요소에 대한 기록을 제공합니다.
- 인증자(**authenticator**) - 인증자 로그는 Amazon EKS에 고유합니다. 이 로그는 Amazon EKS가 IAM 자격 증명을 사용하여 Kubernetes 역할 기반 액세스 제어(RBAC) 인증에 사용하는 제어 영역 구성 요소를 나타냅니다.
- 컨트롤러 관리자(**Controller Manager**) - 컨트롤러 관리자는 Kubernetes와 함께 제공되는 핵심 제어 루프를 관리합니다.

- **스케줄러(scheduler)** - 스케줄러 구성 요소는 클러스터에서 Pod를 실행할 시기와 위치를 관리합니다.

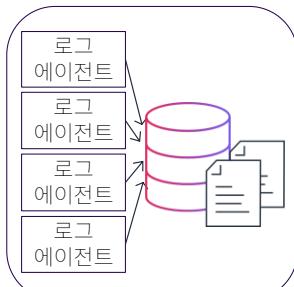
자세한 내용은 **Amazon EKS** 사용 설명서의 'Amazon EKS 제어 영역
로깅'(<https://docs.aws.amazon.com/eks/latest/userguide/control-plane-logs.html>)을 참조하십시오.

로깅 워크플로

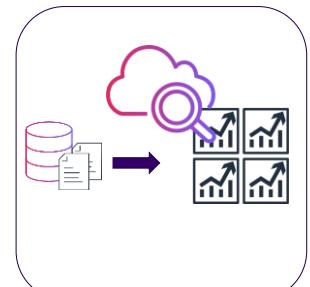
로그에서 인사이트를 얻으려면 3가지 주요 워크플로가 필요합니다.



로그 수집 및 전달



로그 집계 및 저장



데이터 시각화



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

| 강사용 노트 및 애니메이션

|<1> 로그 집계 및 저장이 나타납니다.

|<2> 시각화 데이터가 나타납니다.

| 수강생용 노트

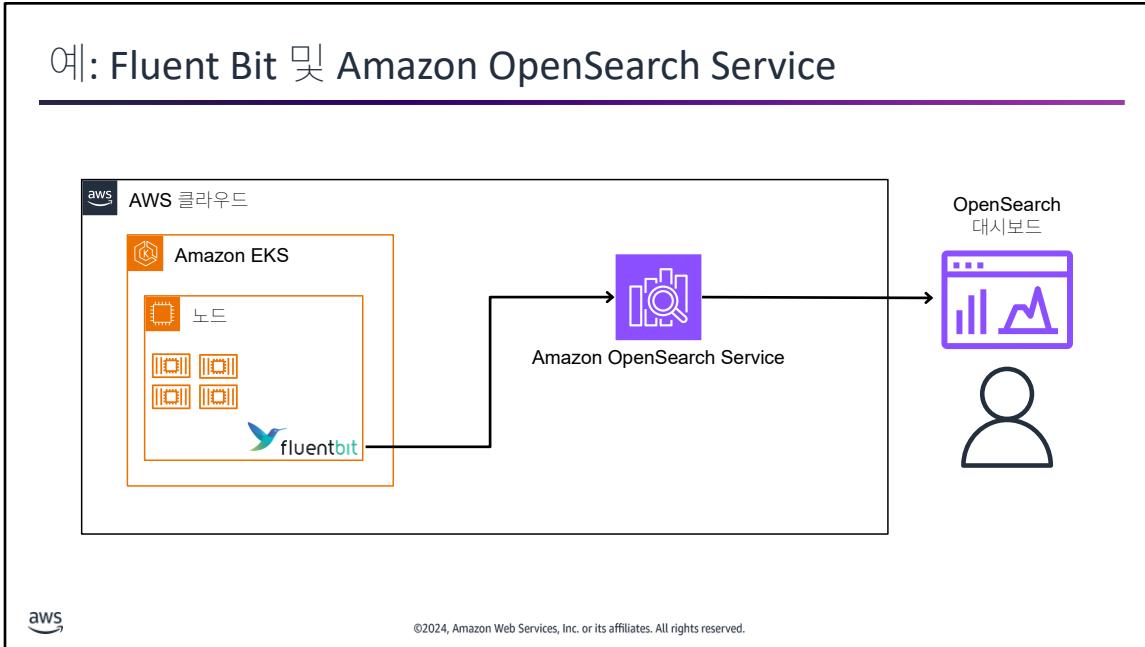
클러스터에서 인사이트를 얻는 과정은 크게 3가지 범주의 노력으로 나눌 수 있습니다:

- 로그 수집 - 애플리케이션을 지원하는 Pod와 해당 Pod를 지원하는 노드에서 로그를 수집해야 합니다.
- 로그 집계 및 스토리지 - 로그를 함께 수집하고 검색할 수 있는 위치에 저장해야 합니다. 의미 있는 인사이트를 반환하는 쿼리를 더 쉽게 만들려면 로그에 태그 추가와 같은 다른 유형의 처리를 수행하는 것이 바람직한 경우가 많습니다.
- 데이터 시각화 - 로그 파일에 있는 방대한 양의 실행 불가능한 인사이트에서 실행 가능한 인사이트를 필터링할 수 있는 방법이 있어야 합니다. 최소한 집계된 로그 그룹을 검색할 수 있는 기능이 필요합니다. 원칙적으로 대시보드 또는 기타 그래픽 출력에서 데이터를 시각화할 수 있어야 합니다.

다음을 포함하여 많은 도구가 이러한 워크플로를 지원할 수 있습니다.

- AWS 서비스(예: CloudWatch)
- 오픈 소스 도구(예: Kibana)
- APN 파트너 도구(예: Datadog, Splunk)

예: Fluent Bit 및 Amazon OpenSearch Service



~dev notes

~Stack: An architectural drawing showing a flow of logging data. Logs flow from Fluent Bit on a node, to Amazon OpenSearch Service, to OpenSearch Dashboards, where they are viewed by a user.

|수강생용 노트

Fluent Bit로 로그를 수집하고 Amazon OpenSearch Service를 사용하여 집계하고 OpenSearch 대시보드를 사용하여 보는 예입니다. 이 예는 Elasticsearch, Fluentd, Kibana 제품을 사용하는 'EFK 스택'의 대안입니다.

- **Amazon OpenSearch Service**

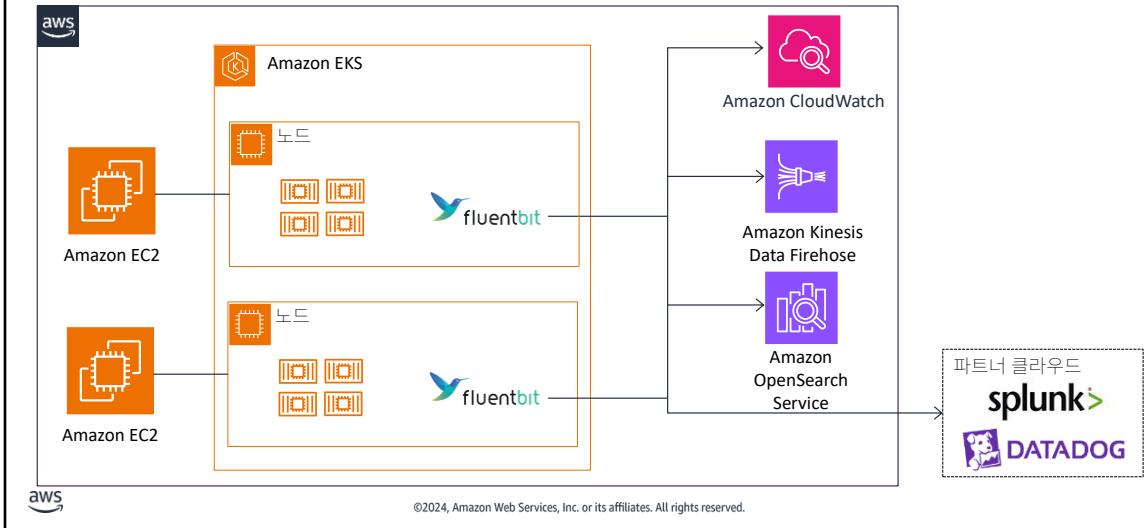
OpenSearch는 오픈 소스 Elasticsearch와 Kibana의 Apache 라이선스 버전 2.0(ALv2) 라이선스 포크입니다. OpenSearch 제품군은 검색 엔진인 OpenSearch와 OpenSearch 대시보드라는 시각화 및 사용자 인터페이스로 구성됩니다. 자세한 내용은 온라인에서 제공되는 'Amazon OpenSearch'(<https://aws.amazon.com/opensearch-service/>)를 참조하십시오.

- **Fluent Bit**

Fluent Bit는 임베디드 및 IoT 디바이스에 사용하는 데 최적화된 가벼운 고성능 데이터 수집기입니다. Fluent Bit는 Fluentd보다 지원하는 플러그 인이 적어 유연성이 떨어집니다. 그러나 사용 공간이 더 작으므로 리소스 효율성이 중요한 마이크로서비스 환경에 적합합니다. AWS가 실행한 테스트에서 Fluentd는 Fluent Bit 플러그 인이 소비하는 것보다 평균 3배 이상의 CPU와 4배의 메모리를 사용했습니다. 이 데이터는 보증을 나타내지 않으며 실제 사용 공간과 다를 수 있습니다. 하지만 데이터 포인트는 Fluent Bit 플러그 인이 Fluentd보다 더 효율적임을 시사합니다.

이 벤치마크에 대한 자세한 내용은 AWS 오픈 소스 블로그의 'Centralized Container Logging with Fluent Bit'(<https://aws.amazon.com/blogsopensource/centralized-container-logging-fluent-bit/>)를 참조하십시오.

예: 컨테이너 로그 라우팅 및 Amazon Elastic Compute Cloud(Amazon EC2)



~ALT text

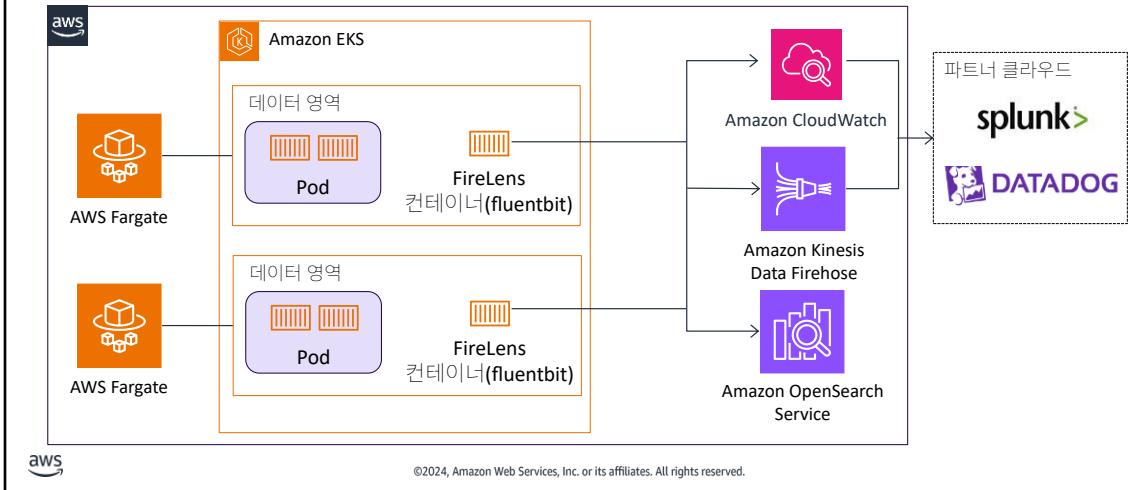
~자세한 내용은 노트를 참조하십시오.

| 수강생용 노트

이 슬라이드에는 로깅 데이터의 흐름을 보여주는 아키텍처 그림이 포함되어 있습니다. 로그는 2개의 노드에서 Fluent Bit에 의해 수집됩니다. 그런 다음 로그는 Amazon CloudWatch, Amazon Kinesis Data Firehose, Amazon OpenSearch Service 또는 파트너 클라우드의 4가지 가능한 대상 중 하나로 라우팅됩니다. Splunk와 Datadog라는 두 파트너가 예시되어 있습니다.

이 예에서는 각 노드에서 DaemonSet로 실행되는 Fluent Bit를 사용하는 로그 라우팅을 보여줍니다. AWS는 로그 보존 및 분석을 위해 컨테이너식 애플리케이션에서 AWS 및 파트너 솔루션으로 스트리밍 로그를 활성화할 수 있도록 Fluent Bit 플러그인을 개발했습니다. 이 플러그인을 사용하면 로그를 Amazon CloudWatch 및 Amazon Kinesis Data Firehose 대상(Amazon S3, Amazon OpenSearch, Amazon Redshift 포함)으로 라우팅할 수 있습니다. 자세한 내용은 **Amazon CloudWatch** 사용 설명서의 'Fluent Bit를 DaemonSet로 설정하여 로그를 CloudWatch로 전송'(<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/Container-Insights-setup-logs-FluentBit.html>)을 참조하십시오.

예: 컨테이너 로그 라우팅 및 AWS Fargate



~ALT text

~자세한 내용은 노트를 참조하십시오.

| 수강생용 노트

이 슬라이드에는 로깅 데이터의 흐름을 보여주는 아키텍처 그림이 포함되어 있습니다. 로그는 Fargate에서 실행되는 2개의 Pod에서 FireLens에 의해 수집됩니다. 그런 다음 로그는 Amazon CloudWatch, Amazon Kinesis Data Firehose, Amazon OpenSearch Service 또는 파트너 클라우드의 4가지 가능한 대상 중 하나로 라우팅됩니다. 이 예에서는 데이터가 FireLens에서 파트너 클라우드로 직접 이동하는 것이 아니라 먼저 Amazon CloudWatch 또는 Amazon Kinesis Data Firehose를 통과합니다. Splunk와 Datadog라는 두 파트너가 예시되어 있습니다.

Fargate를 사용할 때 이전에는 사이드카를 실행하여 AWS Fargate에서 실행 중인 Amazon EKS Pod에서 컨테이너 로그를 라우팅해야 했습니다. 이제 기본 제공 로그 라우터인 FireLens를 사용할 수 있으며 사이드카를 설치하거나 유지 관리할 필요가 없습니다. 데이터를 전송할 위치를 선택하기만 하면 로그가 선택한 대상으로 라우팅됩니다. 백그라운드에서 Amazon EKS on Fargate는 AWS에서 관리하는 Fluent Bit의 업스트림 준수 배포인 Fluent Bit for AWS 버전을 사용합니다.

FireLens를 사용하면 Amazon CloudWatch, Amazon OpenSearch Service, Amazon Kinesis Data Firehose, Amazon Kinesis Data Streams로 로그를 전송할 수 있습니다. 또한 Kinesis Firehose 또는 CloudWatch를 통해 Datadog, Splunk 등과 같은 파트너 대상에 로그를 전송할 수도 있습니다.

컨테이너 로그를 어디로 전송할지 **Fargate**에 알려주려면, **Fluent Bit**의 구성을 데이터 값으로 사용하여 클러스터에 **ConfigMap**을 적용해야 합니다. 로깅 **ConfigMap**을 전체 클러스터에 적용하려면 이를 **aws-observability**라는 고정 **Namespace**에 배치해야 합니다.

자세한 내용은 **AWS Containers Blog**의 'Fluent Bit for Amazon EKS on AWS Fargate'(<https://aws.amazon.com/blogs/containers/fluent-bit-for-amazon-eks-on-aws-fargate-is-here/>)를 참조하십시오.

예: Fargate 로그 라우팅 구성

```
kind: Namespace
apiVersion: v1
metadata:
  name: aws-observability
  labels:
    aws-observability: enabled

kind: ConfigMap
apiVersion: v1
metadata:
  name: aws-logging
  namespace: aws-observability
data:
  output.conf: |
    [OUTPUT]
    Name kinesis_firehose
    Match *
    region us-west-2
    delivery_stream my-stream-firehose
```



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

| 수강생용 노트

AWS Fargate에서 실행되는 FireLens 컨테이너는 Amazon에서 관리합니다. 사용자는 ConfigMap을 사용하여 로그 라우터를 구성하기만 하면 됩니다. 이 슬라이드는 필요한 2개 단계의 예를 보여줍니다.

1. 전용 Kubernetes 네임스페이스를 생성합니다. name 값은 **aws-observability**이어야 하며 **aws-observability: enabled** 레이블이 필요합니다.
2. 컨테이너 로그를 전송할 대상을 지정하는 **aws-logging**이라는 ConfigMap을 생성합니다. data: 값은 기본 Fluent Bit 구성 파일과 동일한 구문을 사용합니다. Fargate 로그 라우터에는 Filter, Output 및 Parser 섹션만 필요합니다.

data: 값은 로그를 보낼 대상에 따라 달라집니다. 이 예에서는 로그를 Kinesis Data Firehose로 보냅니다. 추가 예를 포함하여 Fargate 로그 라우터 구성에 대한 자세한 내용은 **Fargate** 로깅

(<https://docs.aws.amazon.com/eks/latest/userguide/fargate-logging.html>)을 참조하십시오.



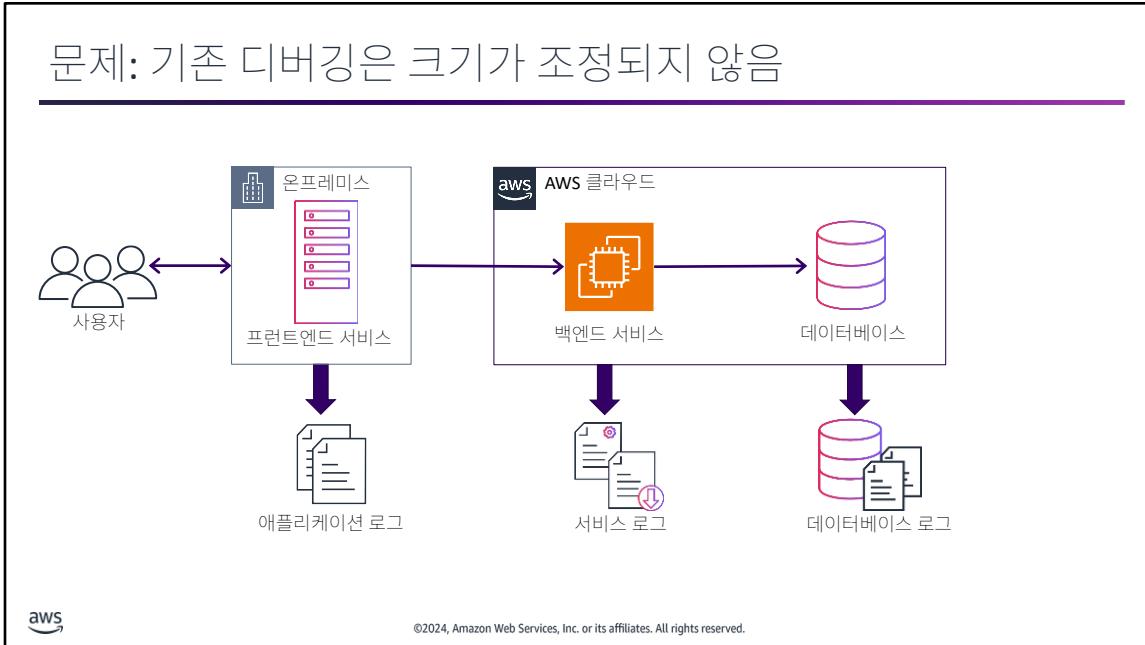
Running Containers on Amazon EKS

Amazon EKS에서 애플리케이션 추적



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

문제: 기존 디버깅은 크기가 조정되지 않음



~ALT text

- ~프런트엔드: 온프레미스 프런트엔드 서비스에서 로그를 수집합니다.
- ~백엔드: 백엔드 서비스 및 데이터베이스에서 서비스 로그 및 데이터베이스 로그를 수집합니다.

|수강생용 노트

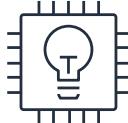
기존 디버깅은 최신 애플리케이션에 맞게 크기가 조정되지 않습니다. 최신 애플리케이션에는 로컬 서비스와 클라우드에 배포 가능한 다양한 서비스가 있을 수 있습니다. 이러한 환경에는 데이터베이스뿐 아니라 게이트웨이, 로드 밸런서 등과 같은 모든 연결 파트가 포함될 수 있습니다.

이러한 모든 구성 요소에 연결된 로그는 비슷한 정도로 서로 다르고 분리되어 있습니다. 프런트엔드 서비스에 문제가 있다고 생각했는데 로그를 조사해 보니 데이터베이스가 원인인 것을 알게 될 수 있습니다. 그래서 데이터베이스 로그를 조사하자 실제로는 예상치 못한 입력을 보내고 있는 클라이언트 문제라는 것을 알게 됩니다.

추적을 통한 글로벌 인사이트



여러 서비스를
검색합니다.



개별 운영에 대한
인사이트를 얻습니다.



서비스 내에서 격리된
문제를 확인합니다.



특정 문제의 근본 원인
분석을 수행합니다.



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

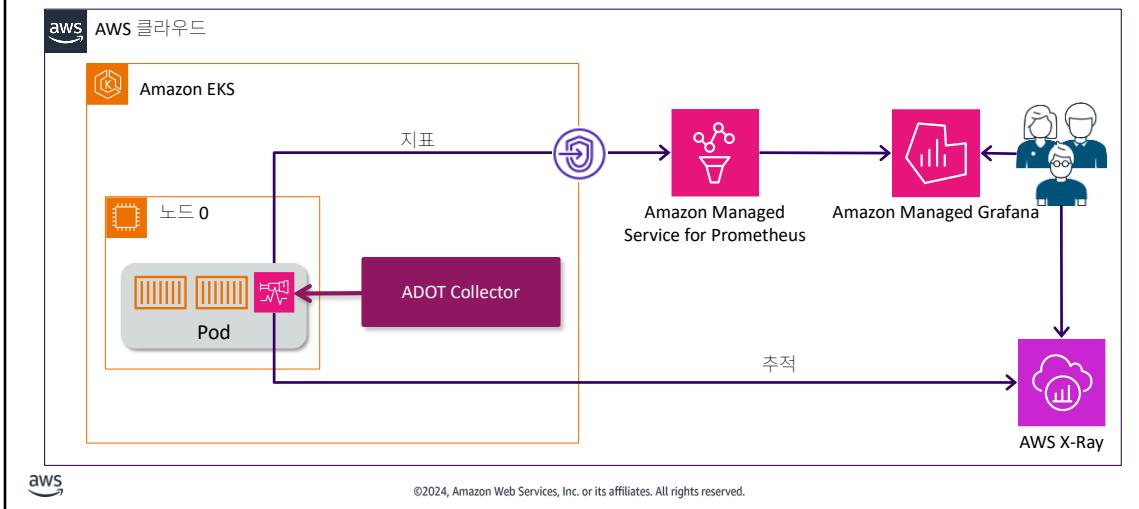
| 수강생용 노트

추적 솔루션을 사용하면 환경의 다양한 구성 요소를 통과하는 특정 요청을 추적할 수 있습니다.

추적을 중요한 도구로 사용하여 다음 활동을 수행합니다.

- 성능 병목 현상 식별
- 특정 서비스 문제를 정확하게 파악
- 오류 식별
- 사용자에 대한 영향 식별

예: AWS Distro for Open Telemetry 및 AWS X-Ray



~Alt text

~아키텍처 다이어그램으로 노트에 자세히 설명되어 있습니다.

~

|수강생용 노트

이 예에서는 **ADOT**를 사용하여 추적 및 지표를 수집하는 방법을 보여줍니다. 추적은 **Pod**에서 사이드카 컨테이너로 실행되는 **ADOT**에 의해 애플리케이션에서 수집되어 검사를 위해 **AWS X-Ray**로 전송됩니다.

이 슬라이드의 다이어그램은 하나의 노드에서 하나의 Pod가 실행되는 Amazon EKS 클러스터를 보여줍니다. Pod에는 사이드카로 실행되는 ADOT Collector가 있습니다. 지표는 ADOT Collector에서 Amazon Managed Service for Prometheus로, 그런 다음 Amazon Managed Grafana로 흐릅니다. 추적은 ADOT Collector에서 AWS X-Ray로 흐릅니다.

AWS X-Ray는 애플리케이션이 처리하는 요청에 대한 데이터인 추적을 수집하는 서비스입니다. **X-Ray**는 데이터를 보고 필터링하고 인사이트를 확보하여 문제 및 최적화 기회를 파악하는 데 사용할 수 있는 도구를 제공합니다. 애플리케이션에 대한 추적된 요청의 경우 요청 및 응답에 대한 자세한 정보를 볼 수 있습니다. 또한 애플리케이션이 다운스트림 AWS 리소스, 마이크로서비스, 데이터베이스, HTTP 웹 API에 대해 실행하는 호출에 대한 정보도 볼 수 있습니다.

AWS X-Ray를 사용하면 다음 작업을 수행할 수 있습니다.

- 분산 애플리케이션의 성능 분석 및 디버깅

- 자연 시간 분포 확인 및 성능 병목 현상 식별
- 애플리케이션 전반에서 특정 사용자에게 미치는 영향 파악
- 다양한 AWS 및 비 AWS 서비스에서 작동

ADOT는 사이드카 대신 서비스로 배포할 수도 있습니다. 즉, 시스템의 모든 애플리케이션에서 액세스할 수 있도록 일정한 수의 **Collector** 인스턴스를 실행하는 것입니다. 이는 **Collector**의 전체 리소스 사용량을 낮추는 이점이 있습니다. AWS에서는 **Collector**를 애플리케이션과 함께 실행할 것을 권장합니다. 시스템 상태를 가장 잘 파악하고 가장 많은 기능을 제공할 수 있기 때문입니다. 이 장단점에 대한 자세한 내용은 **AWS Distro for OpenTelemetry Collector** 배포 유형(<https://aws-otel.github.io/docs/getting-started/collector/sidecar-vs-service>)을 참조하십시오.

모듈 요약



aws

이 모듈에서 학습한 내용:

- 관찰 가능성은 모니터링 그 이상입니다.
- 지표는 상태 모니터링 및 자동 크기 조정을 위한 강력한 도구입니다.
- 마이크로서비스 아키텍처에는 로그 관리를 위한 강력한 전략이 필요합니다.
- 애플리케이션 추적을 통해 관찰 가능성으로부터 인사이트를 얻을 수 있습니다.

©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

| 강사용 노트

| 수강생용 노트



Running Containers on Amazon EKS

실습 4: Amazon EKS 모니터링



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

| 수강생용 노트

실습 4에서는 **Amazon EKS**에서의 모니터링에 대해 자세히 배웁니다.

실습 4



aws

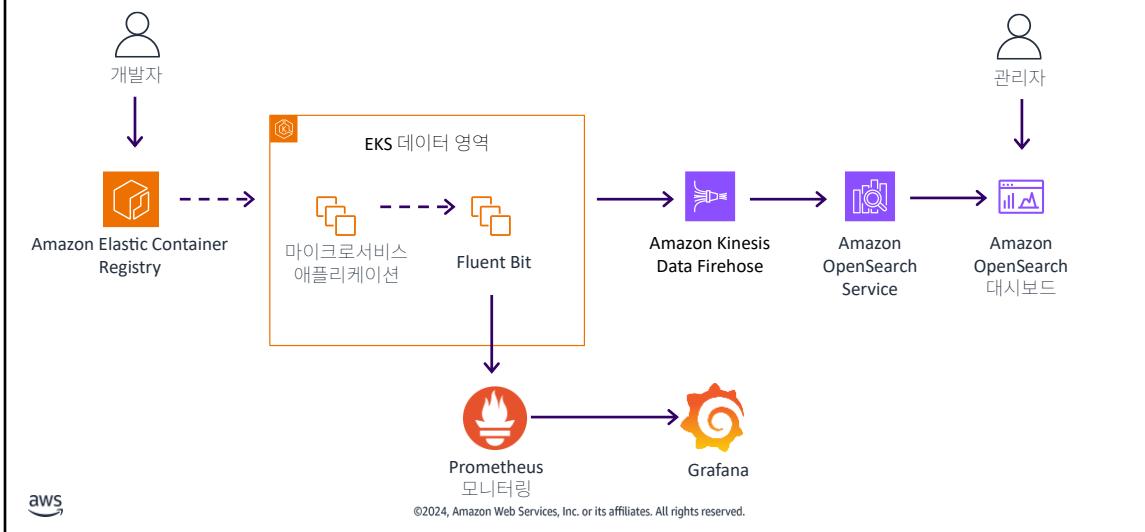
Amazon EKS 모니터링

이 실습에서는 다음 과제를 수행합니다.

- FluentBit Daemonset를 사용하여 Amazon EKS 클러스터에서 로그를 라우팅합니다.
- Amazon Kinesis Data Firehose에서 Amazon S3 버킷으로 데이터를 구성하고 스트리밍합니다.
- Amazon OpenSearch Service를 사용하여 Amazon S3 버킷의 데이터를 구성하고 분석합니다.
- Prometheus를 사용하여 컨테이너식 애플리케이션 지표를 수집 및 요약하고 Grafana를 사용하여 확인합니다.

©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

실습 4 아키텍처 다이어그램.



****접근성을 위한 설명:** 이 슬라이드에는 아키텍처 그림이 포함되어 있습니다. 개발자가 **Amazon Elastic Container Registry(Amazon ECR)**에 이미지를 푸시합니다. 해당 이미지를 기반으로 하는 마이크로서비스 애플리케이션이 실행되어 추적을 **Prometheus**로 전송합니다. 노드의 **Fluent Bit DaemonSet**는 로그를 **Amazon Kinesis Data Firehose**로 전송합니다. **Amazon Kinesis Data Firehose**는 로그를 **Amazon OpenSearch Service**로 전송합니다. 관리자는 **OpenSearch 대시보드**를 사용하여 로그를 쿼리합니다. 설명 끝.



Running Containers on Amazon EKS

감사합니다.



수정 사항이나 피드백 또는 기타 질문이 있으십니까?

<https://support.aws.amazon.com/#/contacts/aws-training>에서 문의해 주십시오.
모든 상표는 해당 소유자의 자산입니다.

©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

| 강사용 노트

| 수강생용 노트