[AWS] 7. 모니터링 : CloudTrail, CloudWatch, AWS Config

- **References**
  - AWS 공인 솔루션스 아키텍트 스터디 가이드 - 어소시에이트 3/e - 7장
  - https://ch4njun.tistory.com/184 (CloudTrail)
  - https://nearhome.tistory.com/134#:~:text=보기 추가해보기-,CloudWatch 란%3F,를 생성할 수 있습니다. (CloudWatch)
  - https://aws.amazon.com/ko/config/#:~:text=AWS Config는 AWS 리소스,자동으로 평가해 줍니다. (AWS Config)

※ AWS CloudTrail, CloudWatch, Config 개요

AWS CloudTrail	AWS 리소스와 관련 된 모든 읽기 및 쓰기 작업에 대한 상세한 로그를 기록
AWS CloudWatch	AWS 및 온프레미스 서버 등 비-AWS 리소스로부터 숫자형 성능 지표 또는 메트릭스 수집 로그파일은 파일 형태로 겁ㅁ색하기 쉽게 관리되며, 알람 및 이벤트 또는 스케줄 기반의 자동화된 대응 액션이 가능하다.

AWS Config	AWS 리소스 환경설정 변경 내역을 추적하고 시간에 따라 어떻게 변화했는지 파악

- **AWS CloudTrail 이란**
  - AWS의 이벤트 (API, 비-API 액션에 대한 로그) 를 분류하고 관리
  - AWS 계정의 운영 및 위험 감사, 거버넌스 및 규정 준수를 활성화 하는데 도움이 되는 AWS 서비스
  - AWS 사용자,역할 및 AWS 서비스가 수행하는 작업이 CloudTrail에 이벤트로 기록 (= AWS 전체 서비스의 Audit Log 역할)
  - 활동 분석 및 응답에 도움이 되는 기타 세부 정보를 식별가능하며, AWS CloudTrail Insights를 활성화하여 비정상적인 활동을 식별하고 이에 대응 가능
  - 이벤트 기록을 통해 AWS 계정에서 이루어진 지난 90일간의 활동을 확인, 검색 및 다운로드
  - 추적을 구성하여 Amazon S3 버킷에 이벤트를 전달 또는 Amazon CloudWatch Logs 및 Amazon EventBridge를 사용하여 이벤트를 전달하고 분석 가능
  - CloudTrail 콘솔, AWS CLI 또는 CloudTrail API 를 사용하여 추적을 생성 가능
  - AWS Organizations 에서 조직을 생성한 후, 해당 조직의 모든 AWS 계정에 대한 모든 이벤트를 로깅하는 트레일 생성 가능, 멤버 계정은 조직 트레일을 볼 수 는 있지만, 수정 삭제 불가능하며 기본적으로 멤버 계정은 S3 버킷의 조직 트레일에 대한 로그파일에 엑세스 할 수 없다.
  - 트레일 생성 후 트레일 구성 변경이 가능하다.

- **CloudTrail 의 Event**
  - IAM 유저 및 Role 등, 사용자가 AWS 리소스에 대해 행한 엑션의 기록
  - 관리이벤트, 데이터 이벤트로 분류된다.

관리 이벤트	AWS 리소스와 관련 사용자가 실행한 작업 또는 실행하려고 시도한 모든 작업이 포함, Control-plan operations (제어 측면의 작업) 이라고 부르기도

	함, Write-Only 작업(리소스 수정 작업 포함) 과 Read-Only 작업(리소스를 읽지만 변경이 되지 않는 작업) 으로 분류된다
데이터 이벤트	데이터 작업 유형에 따라 S3 객체 레벨 작업, Lambda 함수 실행 두가지 타입이 있다. S3 객체 레벨 작업의 경우 Read-Only, Write-Only 이벤트를 구분

- **AWS CloudTrail 의 Trail(트레일 또는 추적)**
  - 90일 이상의 이벤트 히스토리를 저장, 로그에서 특정 서비스 또는 액션을 제외, S3 다운로드나 업로드 등을 포함, CloudTrail의 커스텀 이벤트 로그를 생성하려는 경우 트레일(Trail) 을 이용
  - CloudTrail 로그를 S3버킷에 전달하는 이벤트 기록과 관련된 환경설정 옵션
  - 로그 엔트리는 리소스에 대한 개별 액션
  - 액션 세부 내용
    - eventTime : 액션이 행해진 날짜와 시간, UTC로 표기
    - userIdentity : 액션을 요청한 사용자(Principal)에 대한 상세한 정보 제공, 사용자 유형, 리소스 네임(ARN) IAM 유저 네임 포함
    - eventSource : 액션이 일어난 서비스 엔드포인트 정보
    - eventName : API 작업 명
    - awsRegion : 리소스가 위치한 리전정보, Route 53, IAM 과 같은 글로벌 서비스의 경우 us-east-1이 된다.
    - sourceIPAddress : 작업 요청자의 IP 주소
  - 단일 리전에 최대 5개의 트레일 생성 가능 (모든 리전 트레일 포함)
  - 트레일 생성 후 CloudTrail이 이벤트를 기록하고 이를 S3 버킷에 로그파일로 저장하는데 약 15분의 시간이 걸린다.

- **Trail 생성 시 관리 이벤트 로그 기록**
  - 웹 콘솔에서 트레일 생성, 관리 이벤트 로그를 기록하는 경우 트레일은 자동으로 글로벌 서비스 이벤트도 기록 (us-east-1에서 발생한 것으로 기록) 하여, 불필요한 로그 기록이 중복 되는 경우 발생

        - aws cloudtrail update-trail —name mytrail —no-include-global-service-events (AWS CLI 명령으로 기록되지 않도록 설정)
        - 모든 리전에 대한 로그가 아닌 개별 리전의 로그만 기록되도록 하여 CloudTrail 이 해당 트레일에 대한 글로벌 이벤트 로그를 비활성화 시키도록 함
        - 웹 콘솔에서 트레일 생성하지 않고, AWS CLI 로 —no-include-global-service-events 플래그를 넣어서 트레일 생성

- **Trail 생성 시 데이터 이벤트 로그 기록**
    - 트레일 당 Lambda 함수, S3 버킷과 프리픽스를 포함 최대 250개 객체를 선택할 수 있다.
    - 250개 이상 객체 로그를 기록하려면 Lambda 함수 또는 S3 버킷의 CloudTrail을 통해 모든 로그를 기록하도록 한다.
    - Lambda 이벤트 로그 트레일 생성 시 단일 리전으로 생성 시 해당 리전의 함수 기록만 남으며 모든 리전 트레일 생성 시 모든 리전에서의 Lambda 함수 로그가 기록

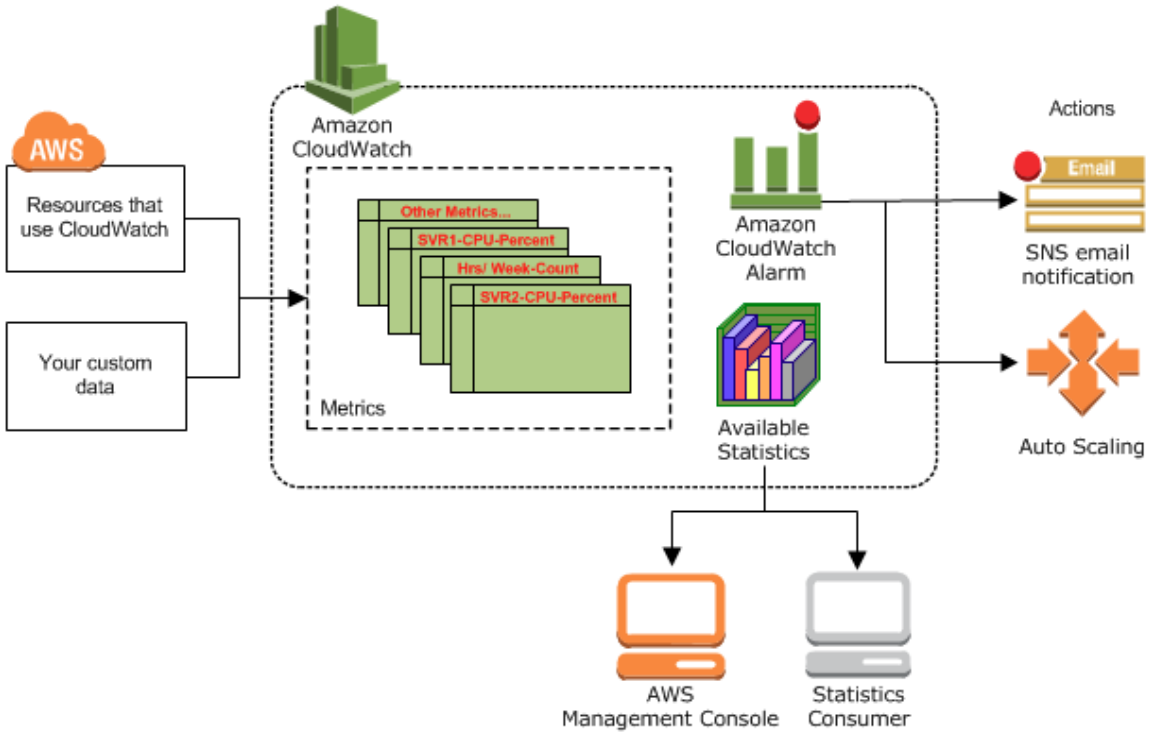
- **Trail 유형**
    - 모든 리전에 적용되는 트레일
        - 모든 리전에 트레일을 생성 시, CloudTrail 은 각 리전에 이벤트를 기록하고 CloudTrail 이벤트 로그파일을 지정된 S3 버킷으로 전송
        - 모든 리전에 적용되는 트레일을 생성한 뒤 리전 추가 시, 새 리전은 자동으로 포함되며 해당 리전의 이벤트는 로깅
        - AWS 권장 방식
        - AWS CLI를 사용하여 단일 리전 트레일만 업데이트 하여 모든 리전을 로그 가능
    - 단일 리전에 적용되는 트레일
        - 단일 리전에서 적용되는 트레일 생성 시, 해당 리전에서만 이벤트 기록하고, 지정 된 S3 버킷에 CloudTrail 이벤트 로그파일 전송
        - AWS CLI를 사용하면 단일 리전 트레일만 생성가능, 단일 트레일을 추가 생성 시, 해당 트레일의 CloudTrail 이벤트 로그파일을 동일한 S3 버킷 또는 별도의

버킷에 전송가능

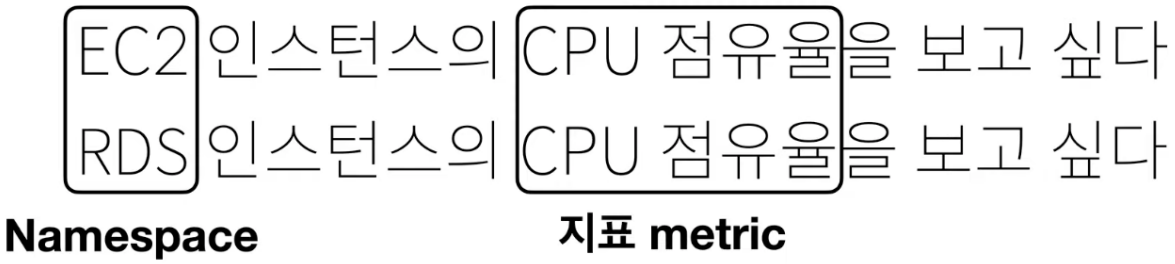
- AWS CLI 또는 CloudTrail API를 사용하여 트레일을 생성할 시 기본 옵션
- 19/04/12 부터 트레일이 이벤트를 로깅하는 AWS 리전에서만 해당 트레일을 볼 수 있다 (단일 리전에서 이벤트를 로그하는 트레일 생성 시), 모든 리전에서 이벤트 로그하는 트레일 생성 시 해당 트레일이 모든 AWS 리전의 콘솔에 표시

- **로그파일 진실성 검증 옵션 (Log File Integrity Validation)**
  - CloudTrail 에서 로그파일 진실성 검증 옵션을 활성화 할 시, S3 버킷에 로그파일이 전송될 때 마다 파일의 암호 해시를 연산 및 확인하게 된다.
  - 매시간, 전송된 로그 파일의 암호화된 해시를 포함한 다이제스트 파일(digest file)을 생성하며, 로그파일과 동일한 위치에 새 폴더를 생성하여 저장한다.
  - CloudTrail 은 리전별 프라이빗 키를 이용하여 다이제스트 파일에 암호화 서명을 하게 되며, 이 서명은 파일의 S3 객체 메타데이터에 포함된다.

※ AWS CloudWatch



- **AWS CloudWatch 란**
  - AWS 리소스 및 비-AWS 리소스의 성능 지표 또는 매트릭스를 수집, 수정, 시각화하는 서비스
  - 주요 성능 매트릭스로 EC2 인스턴스의 CPU 활성화, EBS 볼륨의 읽기 및 쓰기 IOPS, S3 버킷용량, DynamoDB의 읽기 및 쓰기 용량 유닛 등
  - CloudWatch Alarms : 매트릭스 값에 따라 알림메시지를 전송하거나 특정 액션이 가능
  - CloudWatch Logs : AWS 리소스, 비-AWS 리소스의 로그를 수집, 저장, 시각화하고 검색 기능 제공, 커스텀 매트릭스 추출 및 제공 기능



- **CloudWatch 매트릭스**
  - 하나의 매트릭은 개별 성능지표 변수이자 시간순으로 작성된 데이터 포인트의 집합
  - 각 매트릭은 네임스페이스, 네임, Dimension (차원) 으로 구분
  - 네임스페이스(namespaces) 기준 매트릭스 관리
    - AWS 서비스 매트릭스는 AWS 네임스페이스에 저장되며, AWS/{서비스명} 으로 저장
    - 커스텀 매트릭스를 위한 커스텀 네임스페이스 사용 가능

- **CloudWatch 기본 모니터링 및 상세 모니터링**
  - AWS 서비스가 CloudWatch에 매트릭스를 전송하는 주기는 해당 서비스의 모니터링 타입에 따라 달라진다, 대부분의 서비스는 기본 모니터링(basic monitoring)을 지원하며 일부 서비스는 기본 모니터링과 상세 모니터링(detailed monitoring)을 지원
  - 기본 모니터링 (Basic Monitoring)
    - 5분 주기로 CloudWatch에 성능지표 전송
    - EC2는 gp2 볼륨에대해 기본 모니터링을 제공하며, 1분마다 지표를 수집하지만 평균적으로 5분마다 지표 전송
    - EC2가 데이터포인트를 CloudWatch에 전송하는 방식은 하이퍼바이저에 따라 다르다

하이퍼바이저	전송 방식 상세
Xen	5분 간격 마지막에 해당 기간의 성능지표 평균 값을 전송 (평균 값 전송 시 타임스탬프 값은 전송기간이 시작되는 지점)

하이퍼바이저	전송 방식 상세
Nitro	5분간 매분마다 데이터포인트를 전송하지만, 각 데이터 포인트는 바로 앞의 값과의 평균, 지속적으로 값을 갱신하는 것이라 생각하면 된다.

    - 상세 모니터링 (Detailed Monitoring)
        - 매분마다 성능 지표 전송,
        - EC2, EBS, RDS, DynamoDB, ECS, Lambda 등 70여개의 서비스가 상세 모니터링을 지원
        - EBS의 경우 io1볼륨의 기본설정으로 상세 모니터링을 제공

- **CloudWatch의 타임스탬프 (일반 해상도, 고 해상도 매트릭스)**
    - 일반 해상도 매트릭스 : 측정 시간의 특정 매트릭을 지정된 분 단위 타임스탬프로 설정
    - 고 해상도 매트릭스 : 1분 미만으로 측정되는 성능지표
    - 타임스탬프를 지정하지 않으면 CloudWatch는 UTC 단위로 성능 지표 기록

- **CloudWatch 매트릭스의 기한 만료 및 삭제**
    - CloudWatch의 매트릭스는 임의로 삭제 불가능하며, 정해진 기간에 자동으로 만료되어 삭제된다.
    - 만료기한은 해상도에 따라 다르며 시간이 경과함에 따라 고해상도 매트릭스는 차츰 저해상도 매트릭스에 편입
    - 고해상도 매트릭스는 3시간 동안 저장, 모든 분 단위 수집 데이터는 1분 해상도의 데이터 포인트에 편입되고, 고해상도 매트릭스 데이터 포인트는 만료와 동시에 삭제
    - 15일 경과 후 1분 해상도에 저장된 5개의 데이터 포인트는 5분 해상도의 데이터 포인트에 편입되어 63일간 유지
    - 63일의 유보기간 경과 후 12개의 데이터 보인트는 1시간 해상도의 매트릭스에 편입된 뒤 15개월간 유지된 후 삭제

- **CloudWatch 매트릭스의 그래프 표현**
  - CloudWatch는 일정 시간 동안의 데이터 포인트에 대한 통계적 분석을 한 뒤 이를 시계열 그래프로 제공
  - 그래프화 하기 위해서는 매트릭스의 통계량, 기간을 지정해야 한다.
  - 주요 통계량

Sum	하나의 단위 기간 동안 모든 데이터 포인트의 합
Minimum	하나의 단위 기간 동안 최저 데이터 포인트
Maximum	하나의 단위 기간 동안 최대 데이터 포인트
Average	하나의 단위 기간 동안 데이터 포인트의 평균
Sample count	하나의 단위 기간 동안 데이터 포인트의 수
Percentile	특정 백분위의 데이터 포인트, 두 자리 수로 백분위를 표시하며, 예를들어 50을 입력 하면 중앙 값을 반환한다. 백분위 입력 시 p50형식으로 입력

  - 기간은 1초부터 30일 까지 설정 가능하며 기본값은 60초
  - CPU 활성화율에 대한 장기적인 패턴을 파악하고 싶으면 15분 기간 단위로 Average 통계량 확인하는 것이 좋음

- **CloudWatch 매트릭스 연산 및 시각화 도구 (Metric Math)**
  - CloudWatch는 매트릭스 연산 및 그래프화를 위해 다양한 수학적 함수를 제공
  - 사칙연산 외에도 다양한 수학식을 제공
    - AVG
    - MAX
    - MIN
    - STDDEV : 표준편차
    - SUM
  - 통계 함수는 시계열 값이 아닌 스칼라 값을 반환하여 바로 그래프화가 불가능하기에 METRICS 함수와 결합해 선택한 지표에 대한 시계열 값을 반환하도록 하여야 함 ex) SUM(METRICS()), METRICS()/STDDEV(m1)

※ CloudWatch Logs

- **CloudWatch Logs 란**
  - AWS 소스는 물론 비 AWS 소스의 로그를 수집할 수 있으며, 이렇게 수집된 로그에서의 검색 및 커스텀 매트릭스 추출이 가능
  - CloudWatch Logs 는 CloudTrail의 로그수집, 인스턴스 애플리케이션 로그수집, Route 53의 DNS 쿼리 로그를 수집할 수 있다.

- **CloudWatch Logs의 로그 스트림과 로그 그룹 (??) → 로그스트림 개념 추가 스터디 필요**
  - CloudWatch Logs에 전달하는 로그 이벤트는 타임스탬프와 UTF-8로 인코딩된 이벤트 메시지가 포함되어야 한다, 보통의 바이너리 데이터는 저장할 수 없다.
  - CloudWatch는 여러개의 로그 스트림을 로그 그룹에 넣어 조직화 가능하며, 하나의 로그스트림은 하나의 로그 그룹에만 존재
  - 로그 그룹의 유지 기간은 1일에서 10년, 또는 무기한으로 설정 가능하며 기본은 무기한
  - 장기 보관이 필요할 경우 로그그룹을 S3버킷으로 내보낼 수 있다.
  - 로그스트림에 매트릭 필터를 적용하여 데이터를 추출해 CloudWatch 매트릭스 생성에 사용할 수 있다.
  - 매트릭 필터는 로그 그룹에 적용되어 로그그룹이 생성된 뒤 필터를 생성할 수 있다.

※ CloudWatch Agent

- **CloudWatch Agent**
  - 온프레미스 서버의 로그를 수집하는 명령줄 기반의 프로그램
  - EC2의 메모리 활성화율 등 기본적으로 생성하지 못하는 성능 지표도 수집 가능하다.

- Agent가 생성한 지표는 커스텀 매트릭스 이며 사용자 지정 커스텀 네임스페이스에 저장된다.

- **CloudTrail 로그를 CloudWatch Logs 로그스트림에 전송**
  - JSON 포맷의 트레일 로그는 S3에 저장할 수 는 있지만 로그를 검색하는 방법은 제공하지 않는다, 하지만 트레일로그를 CloudWatch Logs의 로그스트림으로 전송하면 각종 지표를 추출하거나 검색하는 것이 가능하다.
  - 256KB를 초과하는 로그 이벤트를 CloudWatch Logs에 전송 불가능

※ CloudWatch Alarms

- **CloudWatch Alarms**
  - 단일 지표를 모니터링 하다가 값의 변화가 발생하면 이메일 알림, 인스턴스 리부팅, Auto Scaling 액션 실행 등 미리 지정된 특정 액션을 취하는 서비스
  - 모니터링 하려는 CloudWatch의 지표를 먼저 정의해야 한다.
  - CloudWatch Alarms 는 시간 흐름에 따른 지표 통계량 변화를 분석한다.

- **모니터링을 위한 데이터 포인트 (?)**

- **CloudWatch Alarms 알람 생성 기준치**
  - 알림을 울리는 기준이 되는 데이터 포인트 값을 기준치로 정할 수 있으며, 두가지 타입이 존재
    - 정적 기준치(Static Threshold) : 특정 정적 값을 기준으로 알람
    - 이상점 감지(Anomaly Detection) : 밴드(band)라 부르는 일정 값의 범위를 벗어 났을 때 알림이 울리도록 설정, 밴드의 크기는 표준편차를 이용하여 정의

- **CloudWatch Alarms 알람 상태**
  - 데이터 포인트 모니터링 기간은 알람상태를 변화시킬 수 있도록 기준치 또는 기준 범위 내에 있어야 한다. 알람의 상태는 세가지로 표현 된다
    - ALARM : 데이터 포인트가 알람 조건에 부합하고, 기간 기준치를 경과한 경우
    - OK : 데이터 포인트가 알람 조건에 부합하지 않지만, 기간 기준치를 경과한 경우
    - INSUFFICIENT_DATA : 알람을 울릴 수 있을 정도의 데이터 포인트가 수집되지 못한 경우

- **CloudWatch Alarms 알람을 위한 데이터 포인트 및 검증 기간**
  - 데이터 포인트는 기준치에 도달 했지만, 기간이 경과하지 않은 상태에서 작동여부를 확인해야 하는 경우, 알람 데이터 포인트 이상으로 검증기간을 설정 가능하다.

- **CloudWatch Alarms 누락 데이터**
  - 검증 기간에 발생하며 예를 들어 인스턴스에서 EBS 볼륨을 분리할 때 발생할 수 있다.
  - CloudWatch는 검증 기간동안 데이터 누락 시 네가지 옵션을 제공
    - As Missing : 기본 설정 옵션으로, 검증 기간 동안 누락 데이터가 발생하지 않은 것처럼 처리
    - Not Breaching : 누락 데이터가 있더라도 기준치를 넘어선 것은 아닌 것으로 처리
    - Breaching : 데이터가 누락될 경우 기준치를 넘어선 상황으로 판단
    - Ignore : 지정된 횟수만큼 연속적으로 데이터 포인트가 누적되기 전 까 지는 알람 상태를 변경하지 않음

- **CloudWatch Alarms 의 액션**

- 하나의 알람에 여러개의 액션을 추가 가능하지만 하나의 액션은 하나의 알람상태 전환에만 적용 가능하다
- SNS 액션 : Amazon SNS 는 토픽 채널을 사용해 소통하며, 퍼블리셔와 구독 개체(subscriber)에게 알림을 보내도록 토픽을 설정 가능
  - 구독 개체는 프로토콜과 엔드포인트로 구분되어 HTTP,.HTTPS,SQS, Lambda, 모바일 푸시알림, 이메일 등이 될 수 있다.
- Auto Scaling 액션 : Auto Scaling 정책을 생성해 인스턴스를 추가 삭제, 알람 생성 전 정책 생성 필요
- EC2 액션 : 알람상태 변경에 따라 인스턴스의 중지, 폐쇄, 재부팅, 복원 등의 액션 가능, 모니터링 지표 차원에 InstanceID가 포함되어 있는 경우에만 가능하며, 액션은 해당 인스턴스에 적용, EC2 액션에는 AWSServiceRoleForCloudWatchEvents 라는 롤이 필요
  - StatusCheckFailed_Instace = 1 일 때 인스턴스 메모리 소진, 파일시스템 오류 등 문제가 발생한 것 → 재부팅 액션
  - StatusCheckedFailed_System = 1 일 때 네트워크 연결 중단, 전원 차단, 하이퍼바이저 오류 등 → 기존 인스턴스를 동일한 설정의 새호스트로 이전해 복원, 인메모리 데이터는 삭제

- **Amazon EventBridge**
  - EventBridge(CloudWatch Events 의 새 버전)는 특정 이벤트 또는 스케줄을 모니터링 하다 관련 액션을 취할 수 있다.
  - EventBridge 는 이벤트에 반응해 액션을 취하는 반면 CloudWatch Alarms 는 성능 지표에 반응해 액션을 취한다는데 차이
  - Event Buses를 모니터링한다.
    - 모든 AWS 계정은 모든 AWS 서비스의 이벤트를 수신하는 하나의 이벤트 버스를 지님
    - 커스텀 이벤트 버스 생성 가능 (애플리케이션 또는 서드파티 서비스의 이벤트 수신)

※ AWS Config

- **AWS Config 란**
  - 특정 시간 대의 AWS 리소스의 환경설정 내역을 추적, 통합적으로 관리한다.
  - 하나의 리소스와 다른 리소스간의 관계파악, 특정요소의 설정 변경이 다른요소에 어떤 영향을 미칠지 파악 등 으로 활용 가능
  - 주요 기능
    - 보안 유지 : 리소스 환경설정 변경 시 알림 제공, 잠재적 위험을 경고
    - 감사 보고 : 환경설정 스냅샷 리포트를 통해 특정 시간대의 리소스 환경설정 내역 파악
    - 문제 해결 : 문제가 발생할 무렵 해당 리소스의 환경설정 내역을 분석 가능
    - 변경 관리 : 특정 리소스의 설정 변화가 다른 리소스에 어떤 영향을 줄지 예측 가능
  - 최저 기준선을 정의한 커스텀 룰 작성 가능
    - 리소스의 최적 환경설정 기준 또는 룰을 정의할 수 있다.
    - 룰을 활성화 하여 정기적 검증 (3, 6, 12, 24 시간 단위) 가능
    - 레코드를 끈 상태에서도 예정대로 실행

- **환경설정 레코더**
  - 기존 리소스 발견, 환경설정 내역 기록, 변경 사항 기록, 시간 흐름에 따른 변경 추적 등의 기능 수행 → 환경설정 아이템을 생성
  - 기본적으로 해당 리전 내 모든 아이템을 모니터링 하며 글로벌 서비스의 리소스도 모니터링
  - 리전당 하나만 사용 가능하다
  - 콘솔 또는 CLI로 언제든지 레코더의 이름을 명시하여 시작, 중지가 가능하다.

- **환경설정 아이템**
  - 환경설정 레코더는 모니터링 대상 리소스 마다 환경설정 아이템을 생성
  - 특정 시점의 리소스 타입, ARN, 생성 시점 등 리소스 설정값, 다른 리소스와의 관계성 정보 등을 포함하고 있다
  - 직접 삭제는 불가능 하며, AWS Config 설정에서 아이템 보관 기간을 최소 30일에서 7년까지 조절 가능하다, 단 보관 기간은 S3로 전송된 환경설정 히스토리 및 스냅샷에는 적용 되지 않음

- **환경설정 히스토리**
  - AWS Config는 환경설정 아이템을 이용하여 리소스별 환경설정 히스토리를 생성한다.
  - 히스토리는 특정 리소스의 시간대별 환경설정 아이템 집합
  - AWS Config 는 리소스 변경 후 6시간 마다 지정 된 S3 버킷(딜리버리 채널)에 환경설정 히스토리를 저장
  - 리소스 타입별로 그룹화
  - AWS Config 콘솔에서 바로 확인 가능하며, 딜리버리 채널에 SNS 토픽을 추가할 수 있도록 할 수 있다.

- **환경설정 스냅샷**
  - 특정 시간대의 모든 환경설정 아이템 집합으로 모니터링 대상이 되는 모든 리소스의 환경설정 백업
  - AWS Config는 일정 간격에 따라 환경설정 스냅샷을 딜리버리 채널에 전송할 수 있으며, 사용자는 콘솔이 아닌 CLI를 이용하여 관련 작업을 수행 가능하다. (JSON 파일-deliveryChannel.json 필요)
    - 딜리버리 채널 명(기본설정)
    - S3 버킷 네임
    - 전송 주기 (3, 6, 12, 24 시간 단위)

- **소프트웨어 인벤토리 기록**
  - AWS Config 는 온프레미스 서버의 소프트웨어 인벤토리 변경 사항도 기록한다
    - 애플리케이션
    - CLI 및 SDK 등 AWS 컴포넌트
    - 운영체제의 이름 및 버전
    - IP주소, 게이트웨이, 서브넷 마스크
    - 방화벽 환경설정
    - Windows 업데이트
  - AWS System Manager 를 이용하여 인벤토리 컬렉션 기능을 활성화 해야하며, SSM:ManagedInstanceInventory 리소스 타입을 모니터링 하도록 하여야 한다.