

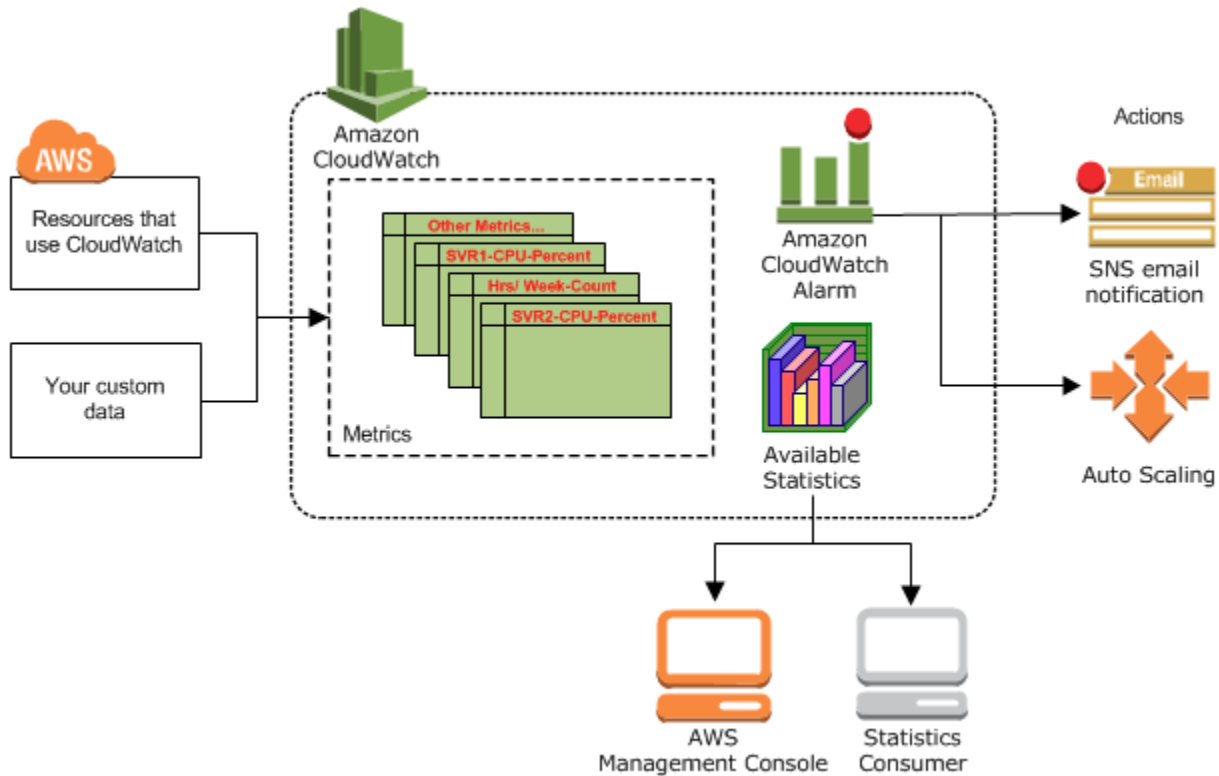
# AWS CloudWatch 메트릭 수신 연구

- CloudWatch와 Grafana 를 함께 쓰는 방향

<https://omty.tistory.com/51>

## CloudWatch

: AWS 리소스와 AWS 실시간 실행 중인 애플리케이션 모니터링 서비스



## AWS SDK

<https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/setup-project-gradle.html>

```
dependencies {  
    implementation platform('software.amazon.awssdk:bom:2.20.56')  
  
    implementation 'software.amazon.awssdk:cloudwatch'  
        implementation 'software.amazon.awssdk:ec2'  
        implementation 'software.amazon.awssdk:rds'  
        implementation 'software.amazon.awssdk:route53'  
        implementation 'software.amazon.awssdk:route53domains'  
  
    ...  
}
```

- GetMetricData & GetMetricStatistics

<https://repost.aws/knowledge-center/cloudwatch-getmetricdata-api>

## GetMetricData VS GetMetricStatistics

AWS cli/API를 사용하여 AWS CloudWatch 지표를 검색하는 방법으로는 GetMetricData 또는 GetMetricStatistics를 사용할 수 있다.

GetMetricData는 대규모 데이터를 빠르게 검색할 수 있을 뿐만 아니라 지표 수식도 지원하지만, 프리 티어에 포함되어 있지 않다.

	호출당 지표 수	호출당 데이터 포인트 수	<a href="#">지표 수식 지원</a>	정렬된 결과를 여러 페이지로 반환	프리 티어 한도에 포함됨
GetMetricData	500	100,800	예	예	아니요
GetMetricStatistics	1	1,440	아니요	아니요	예

### 1. 요청 범위:

- GetMetricData: 한 번의 요청으로 여러 메트릭 데이터를 조회할 수 있습니다. 여러 리소스에 대한 여러 메트릭을 한 번의 요청으로 가져올 수 있어서, 대량의 데이터를 효율적으로 가져오는 데 유용합니다.
- GetMetricStatistics: 한 번의 요청으로 한 가지 메트릭에 대한 통계만 가져올 수 있습니다.

### 2. 시간 범위:

- GetMetricData: StartTime과 EndTime을 지정하여 원하는 시간 범위 내의 데이터를 조회할 수 있습니다.
- GetMetricStatistics: StartTime, EndTime, 그리고 Period (통계를 얻기 위한 기간)을 지정하여 데이터를 조회합니다.

### 3. 응답 형식:

- GetMetricData: 응답은 MetricDataResults 형식으로 반환되며, 이는 요청된 각 메트릭 데이터 쿼리의 결과를 포함합니다.
- GetMetricStatistics: 응답은 Datapoints 및 Label로 반환되며, 지정된 시간 범위와 주기에 대한 지정된 통계의 데이터 포인트를 제공합니다.

### 4. 비용:

- GetMetricData는 요청당 메트릭 데이터 쿼리 수에 따라 비용이 청구됩니다. 여러 메트릭 데이터를 한 번에 조회할 수 있는 능력 때문에 비용 효율적일 수 있습니다.
- GetMetricStatistics는 요청당 비용이 청구됩니다.

### 5. 사용 사례:

- GetMetricData는 대시보드나 복잡한 모니터링 솔루션에서 여러 메트릭을 동시에 조회해야 할 때 유용합니다.
- GetMetricStatistics는 특정 메트릭에 대한 정보가 필요할 때 빠르고 간단하게 사용할 수 있습니다.

요약하면, GetMetricData는 복잡하고 큰 범위의 데이터를 한 번에 조회할 때 유용하며, GetMetricStatistics는 특정 메트릭에 대한 통계를 빠르게 얻을 때 사용됩니다.

\*GetMetricData

```

GetMetricDataRequest request = GetMetricDataRequest.builder()
    .startTime(startTime)
    .endTime(endTime)
    .metricDataQueries(
        Collections.singletonList(
            MetricDataQuery.builder()
                .id("m1")
                .metricStat(
                    MetricStat.builder()
                        .metric(
                            Metric.builder()
                                .namespace
                                ("AWS/EC2")
                                .metricName
                                ("CPUUtilization")
                                .build())
                        .period(300)
                        .stat("Average")
                        .build())
                .returnData(true)
                .build())
        .build());

GetMetricDataResponse response = cloudWatchClient.getMetricData(request);

```

#### \*GetMetricStatisticsRequest

```

GetMetricStatisticsRequest request = GetMetricStatisticsRequest.builder()
    .namespace(namespace)
    .metricName(metricName)
    .startTime(startTime)
    .endTime(endTime)
    .period(300)
    .dimensions(dimension)
    .statistics(MetricStatistic.AVERAGE.getValue())
    .build();

GetMetricStatisticsResponse response = cloudWatchClient.getMetricStatistics
(request);

```

- namespace : 메트릭의 네임스페이스.  
예 ) AWS/EC2, AWS/RDS, AWS/S3 등
- metric-name : 지표 이름  
예 ) EC2의 CPUUtilization 등
  - demensions : 지표 차원. 'Name=string,Value=string' 형식으로 여러 차원을 나열할 수 있다.  
차원은 공백으로 구분한다.
- start-time : 반환할 첫 번째 데이터의 시작 시간. ISO 8601 UTC 형식이어야 한다.  
예 ) 2016-10-03T23:00:00Z
- end-time : 반환할 데이터의 마지막 시간. ISO 8601 UTC 형식이어야 한다.  
start-time과 end-time 범위 내의 데이터가 반환
- period : 데이터를 반환하는 시간 간격.  
가령 start-time이 2016-10-03T23:00:00Z, end-time이 2016-10-03T23:05:00Z, period가 60이면 1분 간격의 데이터를 5개 반환
- statistics : 메트릭 통계. 최대값, 최소값, 평균값 등에 해당
- unit : 측정 값의 단위.  
예 ) Seconds, Byte, Gigabytes 등

## 기능 검토

<https://github.com/awsdocs/aws-doc-sdk-examples/tree/main/javav2>

### EC2 →

- 인스턴스명
- 메트릭(CPU, Memory, Storage, Network)
- 수치값

RDS (현재 등록된 RDS 없어, 확인 필요)

- RDS명
- 매트릭
- IP 엔진
- 수치값

Route53

- 도메인명 →
- 상태값 ( → 헬스체크 먼저 생성 필요)

Billing → 코드 찾지 못함

- 서비스명
- 비용
- 총비용 (점검)

## EC2



response\_1694564492909.json



response\_1694407352235.json

```
String namespace = "AWS/EC2";String metricName = "CPUUtilization";
```

```
// statistics
[
  {
    "instanceId": "i-02169d575ab129ec0",
    "instanceType": "t3a.large",
    "tagName": "Nexus",
    "timestamp": "2023-09-11T04:37:00Z",
    "average": 0.8875002817222234,
    "maximum": 0,
    "minimum": 0,
    "unit": "Percent"
  },
  {
    "instanceId": "i-0286106c2c0aa1574",
    "instanceType": "t3a.medium",
    "tagName": "Platform-PRD-Jenkins-Master",
    "timestamp": "2023-09-11T04:37:00Z",
    "average": 29.48431614387146,
    "maximum": 0,
    "minimum": 0,
    "unit": "Percent"
  },
  .
  .
  .
]
```

```
// statistics
[
  {
    "instanceId": "i-02169d575ab129ec0",
    "instanceType": "t3a.large",
    "tagName": "Nexus",
    "namespace": "AWS/EC2",
    "metricName": "CPUUtilization",
    "timestamp": "2023-09-13T00:16:00Z",
    "average": 0,
    "maximum": 0.8999850002499958,
    "minimum": 0,
    "unit": "Percent"
  },
  {
    "instanceId": "i-0286106c2c0aa1574",
    "instanceType": "t3a.medium",
    "tagName": "Platform-PRD-Jenkins-Master",
    "namespace": "AWS/EC2",
    "metricName": "CPUUtilization",
    "timestamp": "2023-09-13T00:16:00Z",
    "average": 0,
    "maximum": 27.024549590840152,
    "minimum": 0,
    "unit": "Percent"
  },
  .
  .
  .
  .
]
```