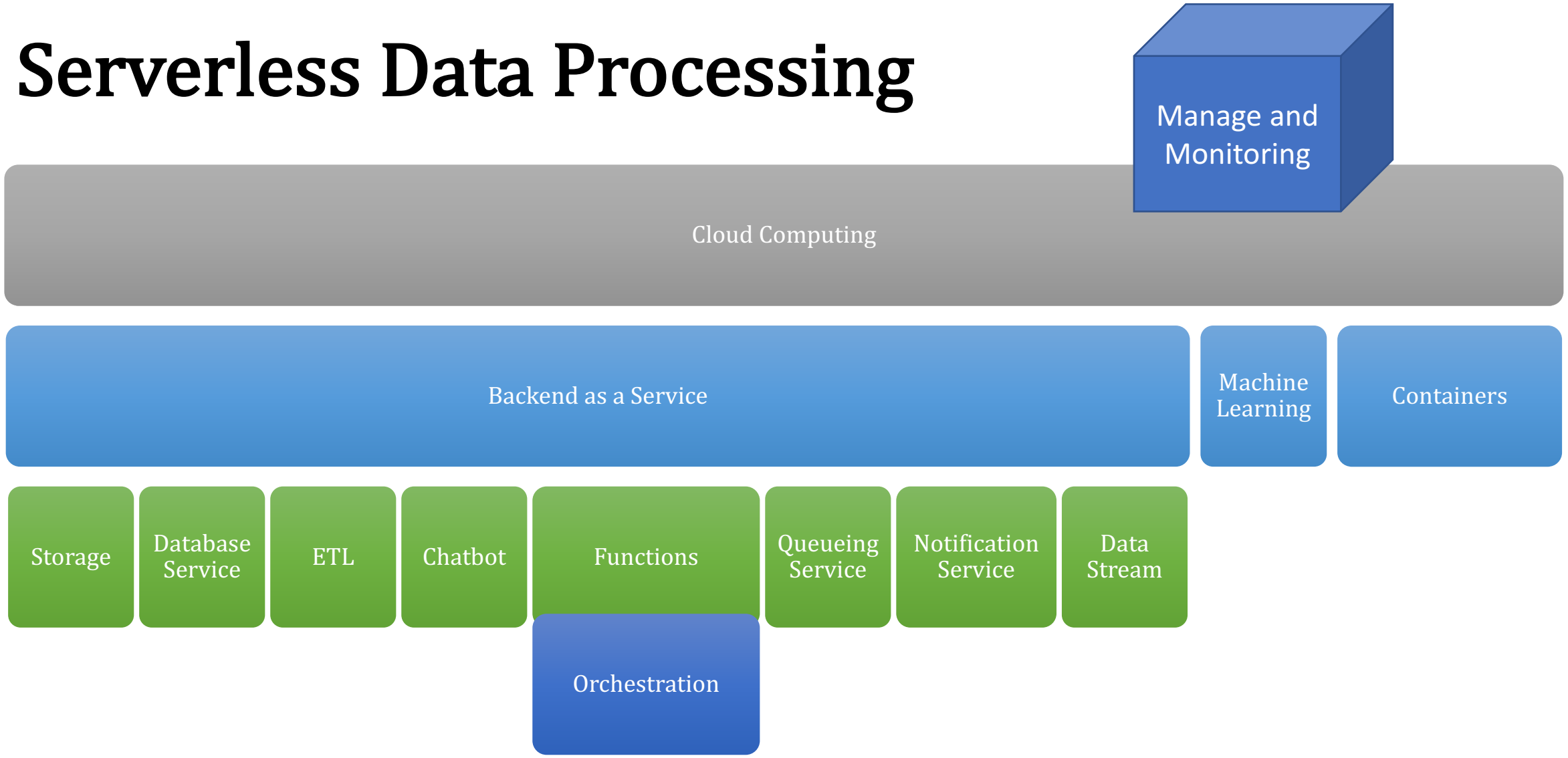
Serverless Data Processing (CSCI 5410)

Dr. Saurabh Dey

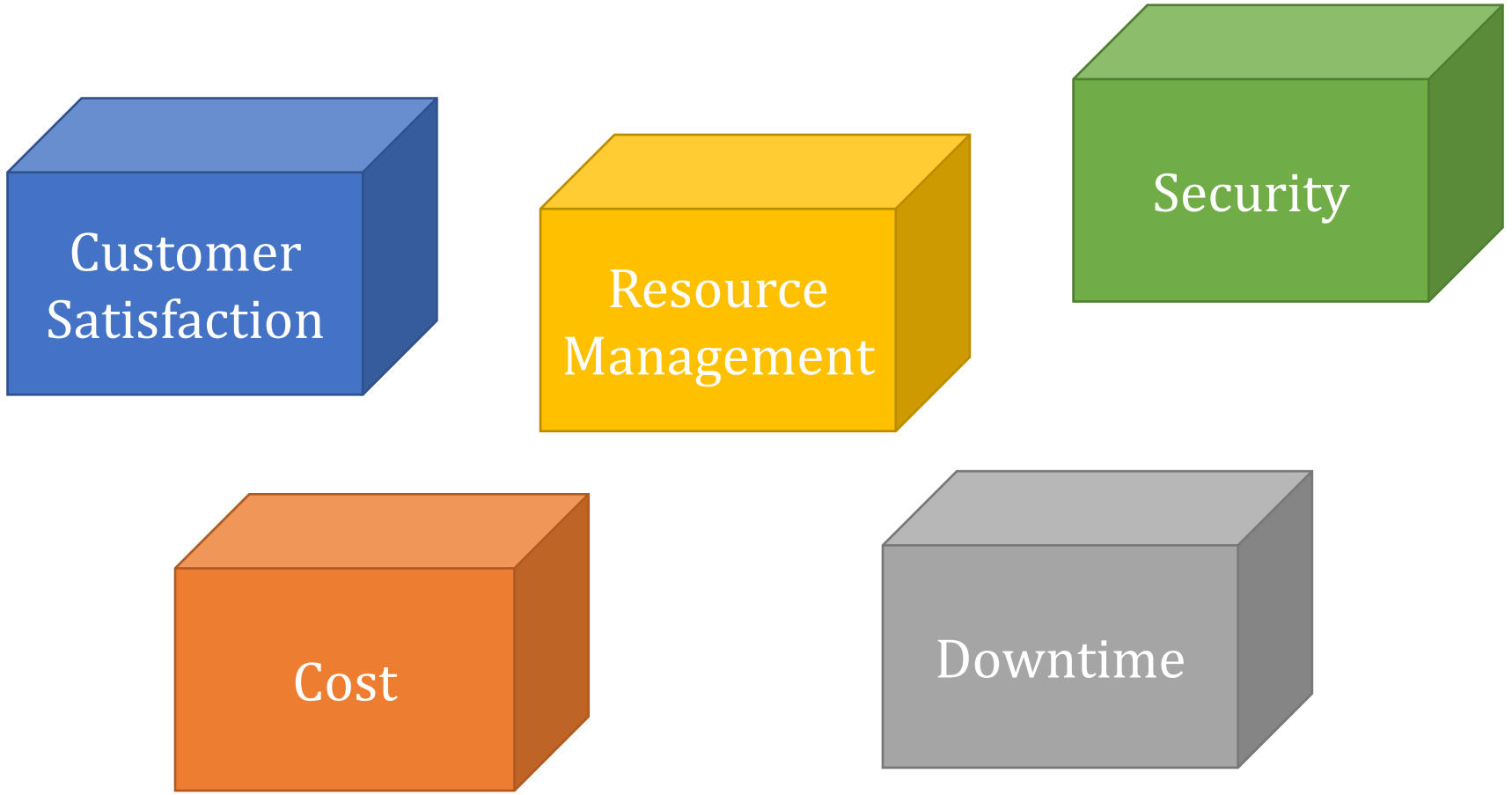
Outline

Cloud Monitoring with CloudWatch

Serverless Data Processing



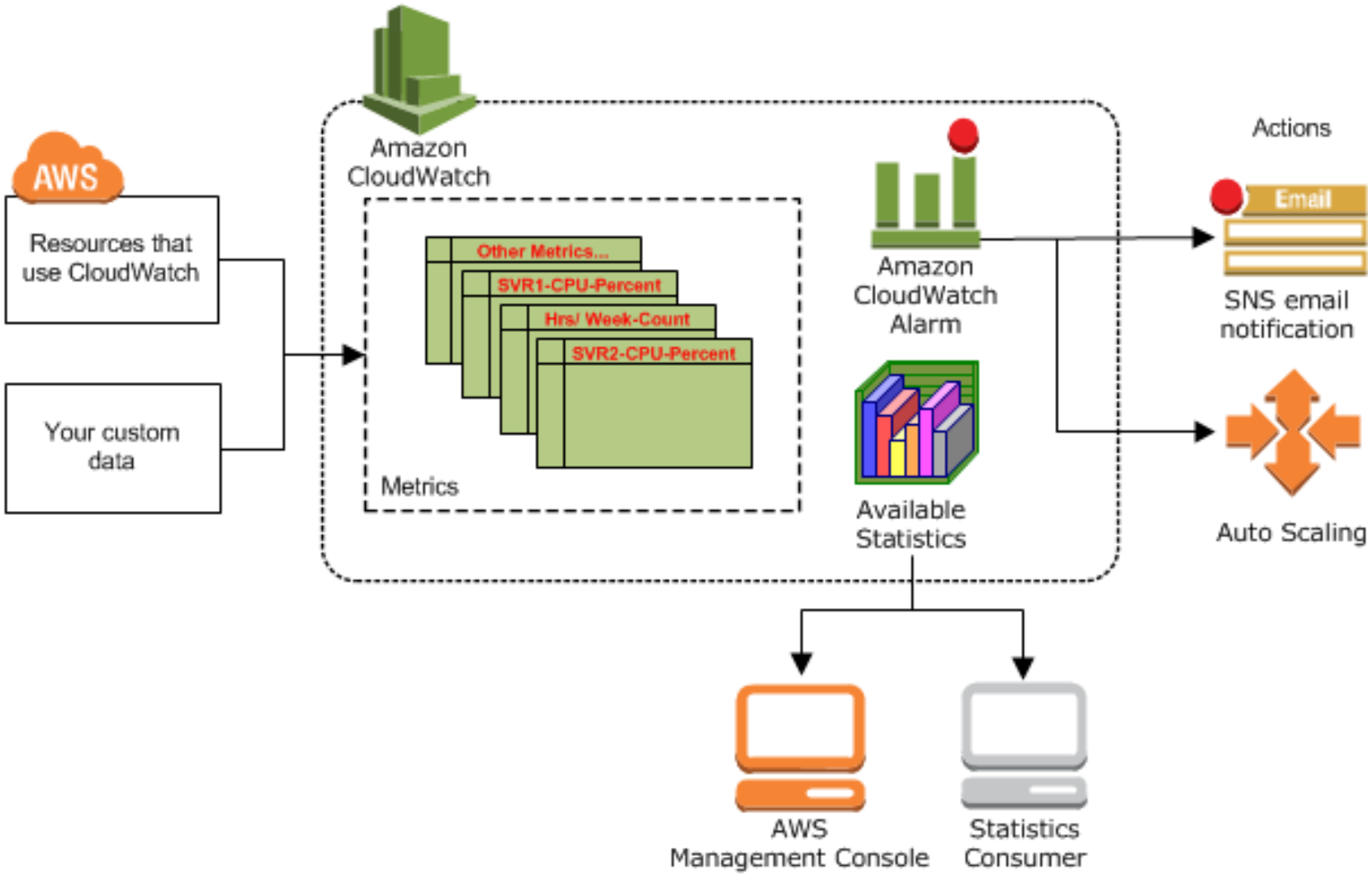
Why do we need cloud monitoring?



CloudWatch

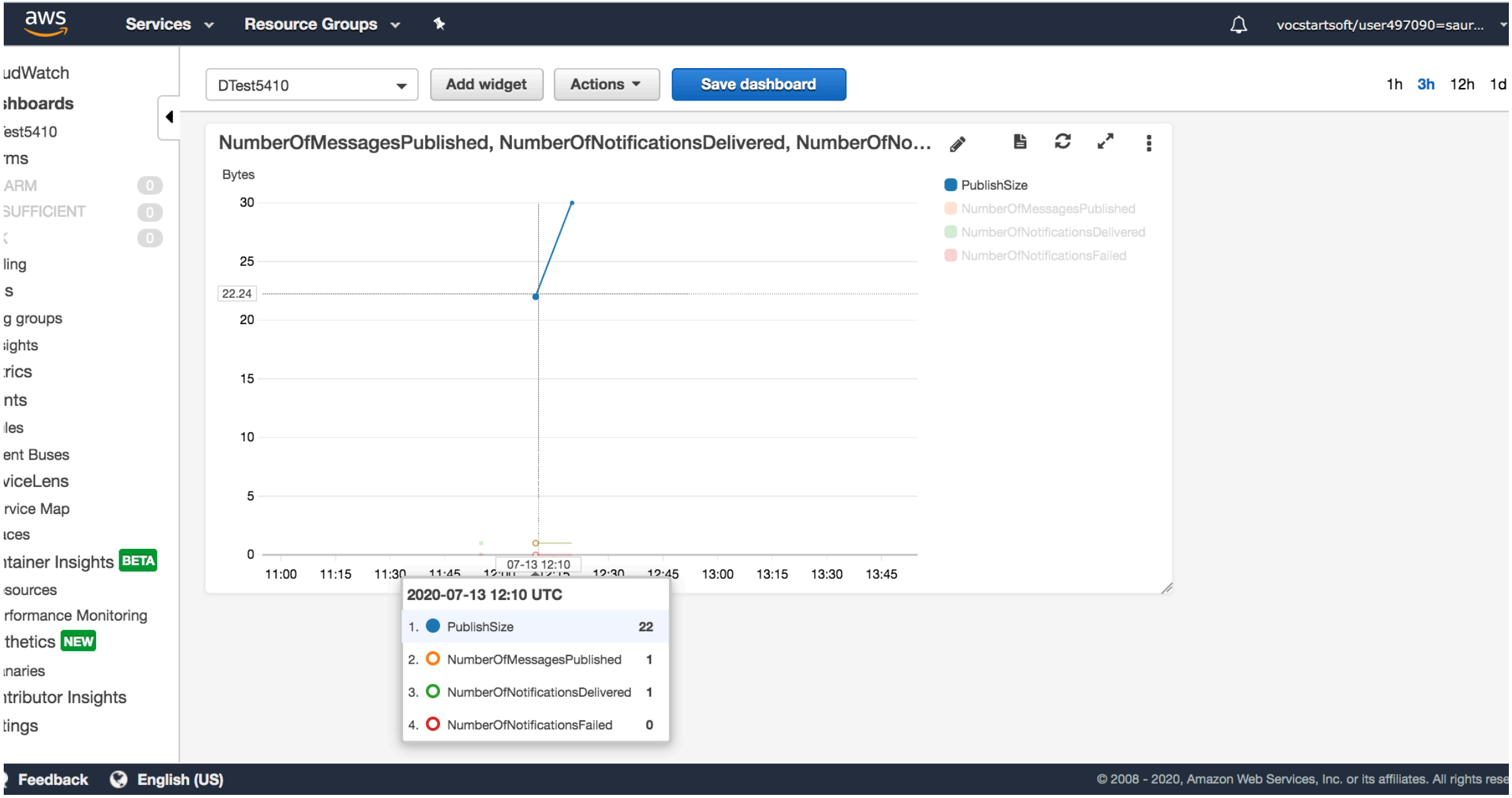
- Amazon CloudWatch is a monitoring and management service that provides data and actionable insights for AWS, hybrid, and on-premises applications and infrastructure resources. With CloudWatch, we can collect and access all our performance and operational data in form of logs and metrics from a single platform.

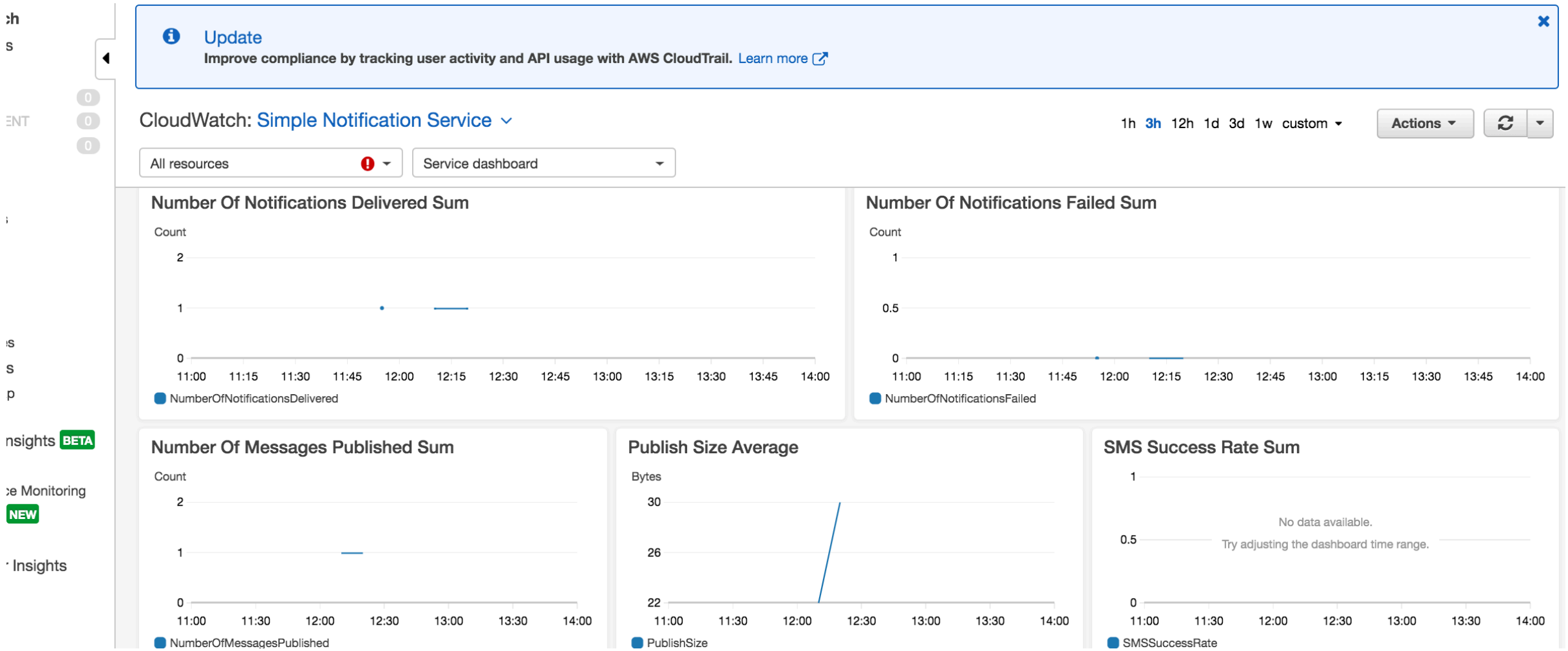
How CloudWatch Works



CloudWatch Concepts

- **Namespaces**: A namespace is a container for CloudWatch metrics.
- **Metrics**: Metrics are the fundamental concept in CloudWatch. A metric represents a time-ordered set of data points that are published to CloudWatch.
- **Timestamps**: Each metric data point must be associated with a time stamp. Timestamps are dateTime objects.
- **Dimensions:** A dimension is a name/value pair that is part of the identity of a metric. We can assign up to 10 dimensions to a metric.
- **Alarms**: We can use an alarm to automatically initiate actions on our behalf.


aws
Services
Resource Groups
vocstartsoft/user497090=saur...
udWatch
hboards
est5410
ms
ARM 0
SUFFICIENT 0
0
ling
s
g groups
ights
rics
nts
les
ent Buses
viceLens
rvice Map
ces
tainer Insights BETA
sources
rformance Monitoring
thetics NEW
naries
tributor Insights
tings
DTest5410
Add widget
Actions
Save dashboard
1h 3h 12h 1d
NumberOfMessagesPublished, NumberOfNotificationsDelivered, NumberOfNo...
Bytes
30
25
22.24
20
15
10
5
0
11:00
11:15
11:30
11:45
12:30
12:45
13:00
13:15
13:30
13:45
07-13 12:10
PublishSize
NumberOfMessagesPublished
NumberOfNotificationsDelivered
NumberOfNotificationsFailed
2020-07-13 12:10 UTC
1. PublishSize 22
2. NumberOfMessagesPublished 1
3. NumberOfNotificationsDelivered 1
4. NumberOfNotificationsFailed 0
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights rese


Update
Improve compliance by tracking user activity and API usage with AWS CloudTrail. Learn more
CloudWatch: Simple Notification Service
1h 3h 12h 1d 3d 1w custom
Actions
All resources
Service dashboard
Number Of Notifications Delivered Sum
Count
2
1
0
11:00 11:15 11:30 11:45 12:00 12:15 12:30 12:45 13:00 13:15 13:30 13:45 14:00
NumberOfNotificationsDelivered
Number Of Notifications Failed Sum
Count
1
0.5
0
11:00 11:15 11:30 11:45 12:00 12:15 12:30 12:45 13:00 13:15 13:30 13:45 14:00
NumberOfNotificationsFailed
Number Of Messages Published Sum
Count
2
1
0
11:00 11:30 12:00 12:30 13:00 13:30 14:00
NumberOfMessagesPublished
Publish Size Average
Bytes
30
26
22
11:00 11:30 12:00 12:30 13:00 13:30 14:00
PublishSize
SMS Success Rate Sum
1
0.5
0
No data available.
Try adjusting the dashboard time range.
11:00 11:30 12:00 12:30 13:00 13:30 14:00
SMSSuccessRate
Insights BETA
Monitoring
NEW
Insights

Using Amazon CloudWatch Metrics

- Metrics are data about the performance of your systems. By default, several services provide free metrics for resources (such as Amazon EC2 instances, Amazon EBS volumes, and Amazon RDS DB instances).

- You can also enable detailed monitoring for some resources, such as your Amazon EC2 instances, or publish your own application metrics.

aws cloudwatch list-metrics --namespace AWS/SNS

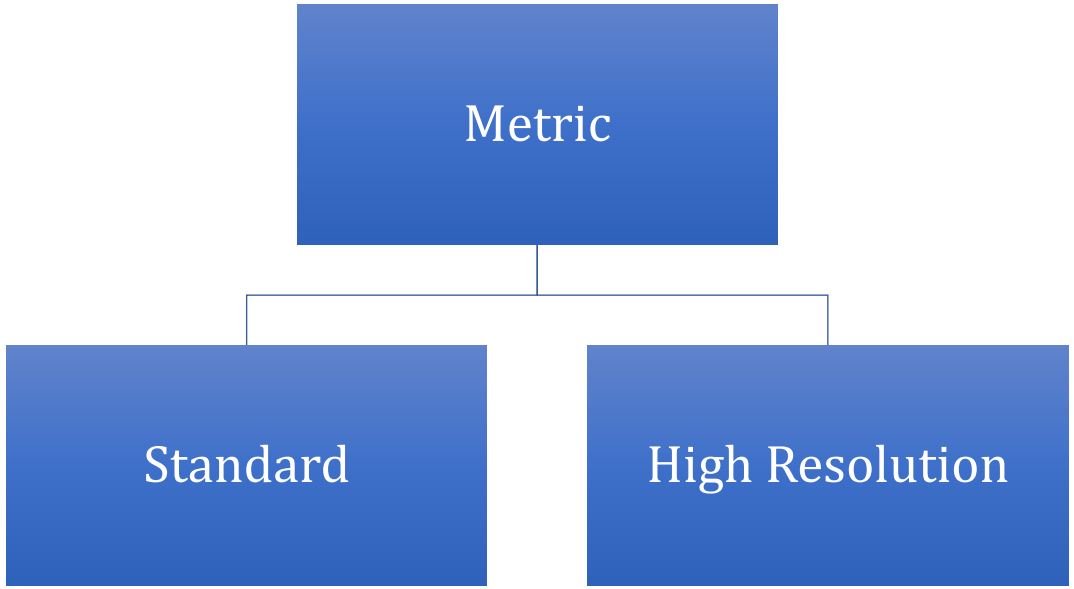
Viewing available Metrics

- Metrics are grouped first by namespace, and then by the various dimension combinations within each namespace.
- For example, you can view all EC2 metrics, EC2 metrics grouped by instance, or EC2 metrics grouped by Auto Scaling group.
- Only the AWS services that you're using send metrics to Amazon CloudWatch.

Publishing Custom Metrics

- You can publish your own metrics to CloudWatch using the AWS CLI or an API. You can view statistical graphs of your published metrics with the AWS Management Console.

- CloudWatch stores data about a metric as a series of data points. Each data point has an associated time stamp. You can even publish an aggregated set of data points called a statistic set.



Using Dimensions

- In custom metrics, the **--dimensions** parameter is common. A dimension further clarifies what the metric is and what data it stores. You can have up to 10 dimensions in one metric, and each dimension is defined by a name and value pair.

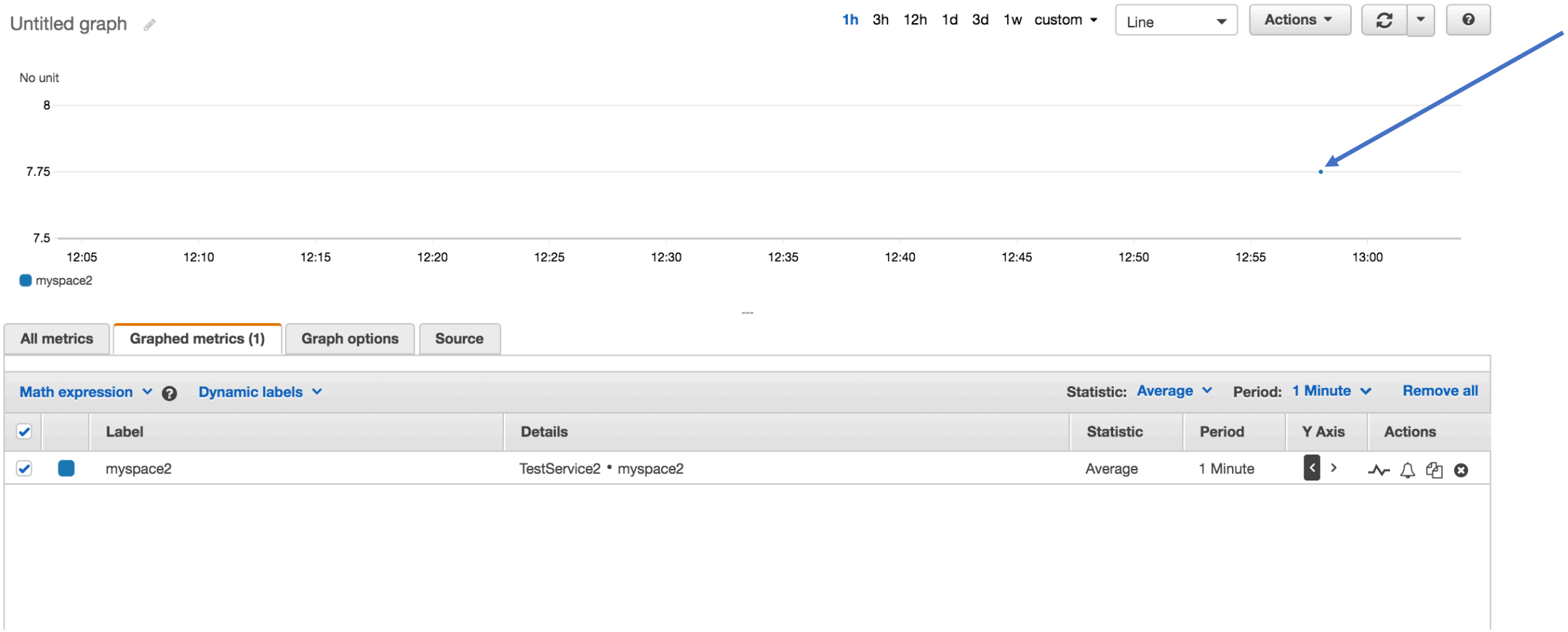
- How you specify a dimension is different when you use different commands. With put-metric-data, you specify each dimension as MyName=MyValue, and with get-metric-statistics or put-metric-alarm you use the format Name=MyName, Value=MyValue.

```
Saurabhs-MacBook-Pro:~ saurabh$ aws cloudwatch put-metric-data --metric-name myspace1 --namespace TestService --value 3 --dimensions Applicat
ion=testApplication
```

Publishing Single Data Points

- To publish a single data point for a new or existing metric, use the **put-metric-data** command with one value and time stamp. For example, the following actions each publish one data point.

```
Saurabhs-MacBook-Pro:~ saurabh$ aws cloudwatch put-metric-data --metric-name myspace2 --namespace TestService2 --value 3 --timestamp 2020-07-15T12:58:00.000Z
Saurabhs-MacBook-Pro:~ saurabh$ aws cloudwatch put-metric-data --metric-name myspace2 --namespace TestService2 --value 7 --timestamp 2020-07-15T12:58:01.000Z
Saurabhs-MacBook-Pro:~ saurabh$ aws cloudwatch put-metric-data --metric-name myspace2 --namespace TestService2 --value 9 --timestamp 2020-07-15T12:58:02.000Z
Saurabhs-MacBook-Pro:~ saurabh$ aws cloudwatch put-metric-data --metric-name myspace2 --namespace TestService2 --value 12 --timestamp 2020-07-15T12:58:03.000Z
```

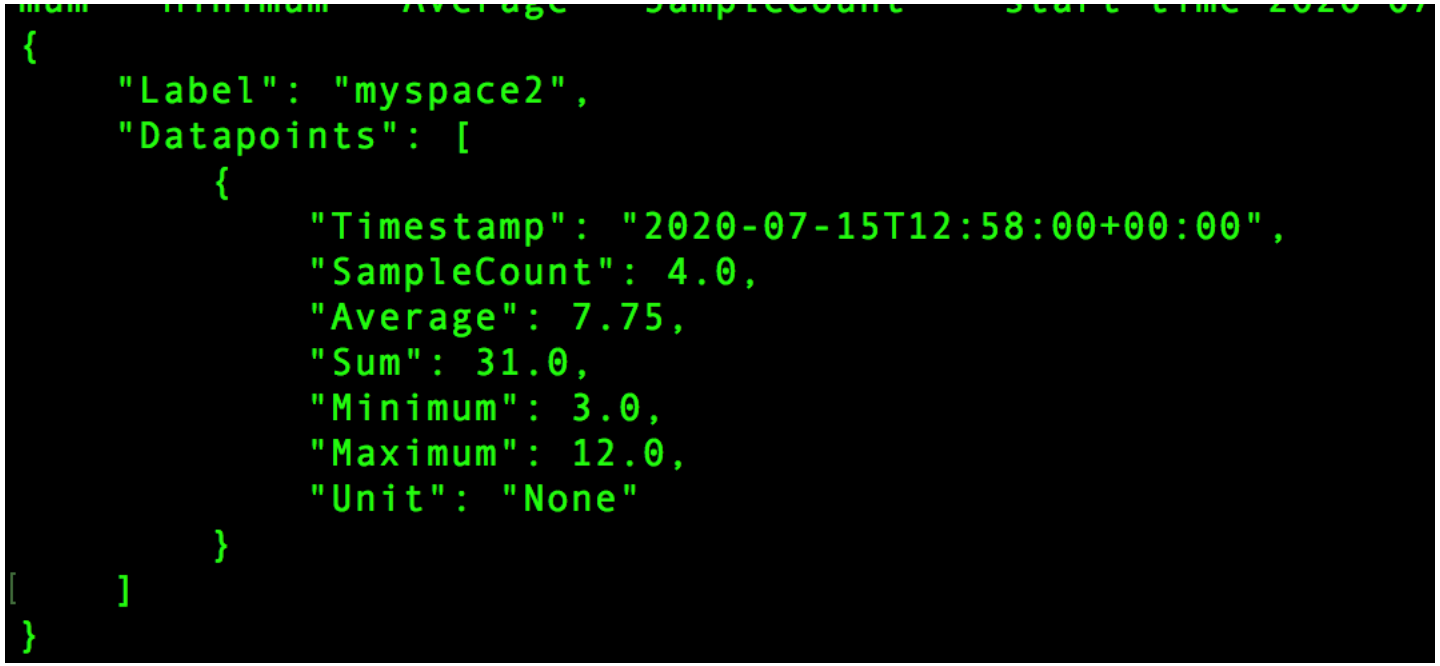


If we select metric in CloudWatch, and explore the NameSpace, we get the graph. It shows the put_metric_data with no dimensions

Retrieve Statistics on published data points

- You can use the **get-metric-statistics** command to retrieve statistics based on the data points that you published.

```
Saurabhs-MacBook-Pro:~ saurabh$ aws cloudwatch get-metric-statistics --namespace TestService2 --metric-name myspace2 --statistics "Sum" "Maxi
mum" "Minimum" "Average" "SampleCount" --start-time 2020-07-15T12:58:00.000Z --end-time 2020-07-15T12:59:00.000Z --period 60
```



```
{
    "Label": "myspace2",
    "Datapoints": [
        {
            "Timestamp": "2020-07-15T12:58:00+00:00",
            "SampleCount": 4.0,
            "Average": 7.75,
            "Sum": 31.0,
            "Minimum": 3.0,
            "Maximum": 12.0,
            "Unit": "None"
        }
    ]
}
```

Questions to Consider

- Can you check the process utilization of EC2 using CloudWatch?
- What is the difference between CloudWatch and CloudTrail?
- How will you monitor Lambda function events?

Thank You