Lecture #11

Campus	Lecture Date	Lecture Recording Availability Date
Boston	07/20/2019	N/A
Online	N/A	07/20/2019

Agenda

- Review Assignment #05 [../../assignments/05/]
- Cloud & Networking Monitoring
  - Logging
  - Metrics
  - Alerting
- Demo Logging & Metrics
- Discuss Assignment #06 [../../assignments/06/]

Slides

- Logging, Metrics, Monitoring & Alerting [https://northeastern-my.sharepoint.com/:b:/g/personal/tejasparikh_northeastern_edu/ESlWCZJq5jFFm8kn6Q73qAABaVsboCk9N7Lt-6ot61W8Vg?e=3ZzmFp]

Metrics, StatsD, CloudWatch, & Web Application

Assignment #06 [../../assignments/06/] requires you to instrument your application, specifically count the number of times your APIs are called and publish those metrics to AWS CloudWatch. You can use either `StatsD` or `CollectD` (or something else) for collecting metrics. Given how popular `StatsD` has become recently, I am going to use StatsD.

What is StatsD?

StatsD [https://github.com/etsy/statsd] is a simple daemon developed and released by Etsy to aggregate and summarize application metrics. With StatsD, applications are to be instrumented by developers using language-specific client libraries [https://github.com/etsy/statsd/wiki#client-implementations]. These libraries will then communicate with the StatsD daemon using its dead-simple protocol, and the daemon will then generate aggregate metrics and relay them to virtually any graphing or monitoring backend.

How Does StatsD Work?

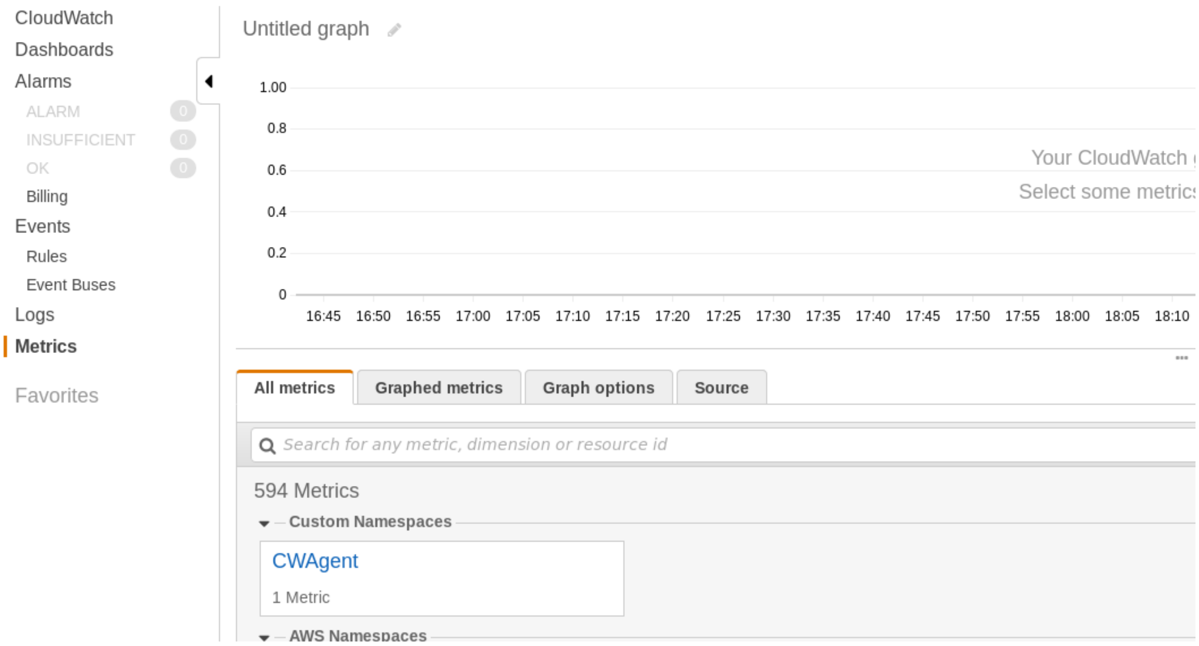
1. You instrument your code with one of the many StatsD libraries corresponding to your app language.
2. StatsD allows you to capture different types of metrics such as `Gauges`, `Counters`, `Timing Summary Statistics`, and `Sets`.
3. The StatsD client library then sends each individual call to the StatsD server over a UDP datagram.
4. The StatsD daemon will then listen to the UDP traffic from all application libraries, aggregate data over time and “flush” it at the desired interval to the backend of your choice.
5. The monitoring backend will turn your metrics from a stream of numbers on the wire into usable charts and alert you when needed.

What do you need to do?

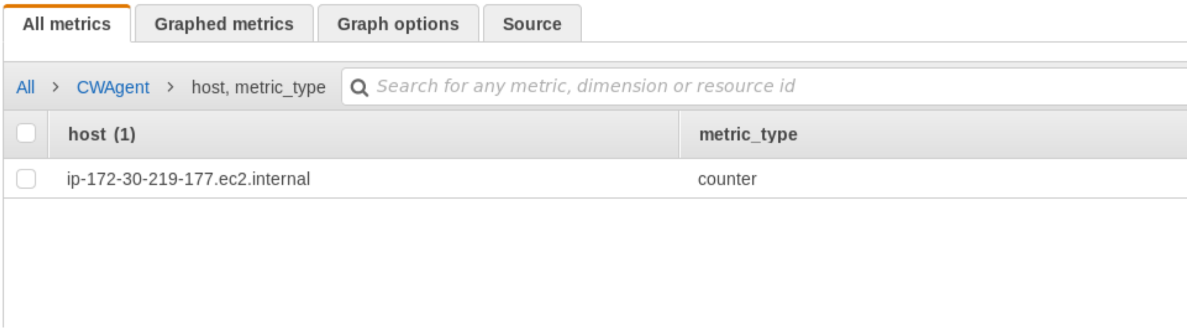
1. Pick a StatsD client library [https://github.com/etsy/statsd/wiki#client-implementations] for your application based on the programming language. Go through the documentation to see how to connect to StatsD daemon.
2. Modify your application to add counters and publish the counters using the StatsD client instance created in step 1. Your StatsD daemon is usually running on `localhost` on port `8125`.
3. Update your CloudWatch configuration to run a StatsD daemon on your EC2 instances. Example configuration can be found here [https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch-Agent-custom-metrics-statsd.html].
4. Update your code and CloudWatch configuration on the EC2 instance to see your new metrics published to CloudWatch.

Screenshots

CWAgent

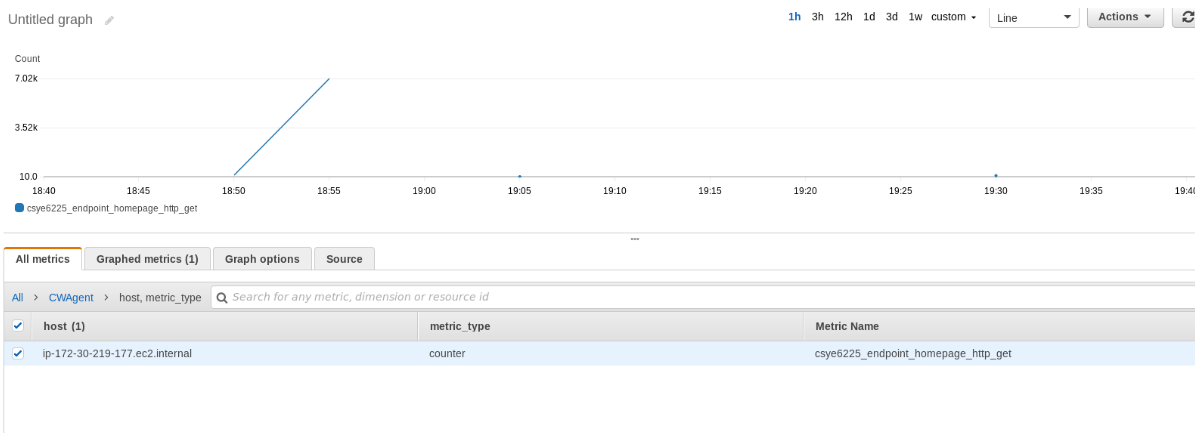


Custom Metrics


All metrics
Graphed metrics
Graph options
Source
All > CWAgent > host, metric_type
Search for any metric, dimension or resource id

□	host (1)	metric_type
□	ip-172-30-219-177.ec2.internal	counter

Graph Showing Metrics



☑	host (1)	metric_type	Metric Name
☑	ip-172-30-219-177.ec2.internal	counter	csye6225_endpoint_homepage_http_get

Reading

- What is Amazon CloudWatch Logs? [http://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/WhatIsCloudWatchLogs.html]
- Amazon CloudWatch [https://aws.amazon.com/cloudwatch]
- The Log: What every software engineer should know about real-time data's unifying abstraction [https://engineering.linkedin.com/distributed-systems/log-what-every-software-engineer-should-know-about-real-time-datas-unifying]
- Apache logging services [http://logging.apache.org/]
- Effective Logging [http://www.kdgregory.com/index.php?page=java.logging]
- Java Logging Standards and Guidelines [https://wiki.base22.com/display/btg/Java+Logging+Standards+and+Guideline

s]

- Retrieve Custom Metrics with StatsD [https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch-Agent-custom-metrics-statsd.html]
- StatsD Client Libraries [https://github.com/etsy/statsd/wiki#client-implementations]