Netsurion. | EventTracker®

How to - Configure AWS Route 53 to forward logs to EventTracker

EventTracker v9.2x and above

Publication Date: January 25, 2021

Abstract

This guide helps you in configuring **AWS Route 53** for **EventTracker** to receive AWS Route 53 events. You will find the detailed procedures required for monitoring AWS Route 53.

Scope

The configurations detailed in this guide are consistent with **EventTracker v9.2x** and later, **AWS Route 53**

Audience

AWS Route 53 users, who wish to forward Events to EventTracker and monitor events using EventTracker.

The information contained in this document represents the current view of Netsurion on the issues discussed as of the date of publication. Because Netsurion must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Netsurion, and Netsurion cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. Netsurion MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, this paper may be freely distributed without permission from Netsurion, if its content is unaltered, nothing is added to the content and credit to Netsurion is provided.

Netsurion may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Netsurion, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The example companies, organizations, products, people and events depicted herein are fictitious. No association with any real company, organization, product, person or event is intended or should be inferred.

Table of Contents

1. Overview

Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. It is designed to provide developers and businesses a way to route end users to Internet applications. Amazon Route 53 is fully compliant with IPv6 as well.

EventTracker helps to monitor events from AWS Route 53. The dashboard and reports help in monitoring DNS query activities.

EventTracker's built-in knowledge pack enables you to gather business intelligence providing increased security, performance, availability, and reliability of your systems.

Through alerts, knowledge base solutions, and reports, EventTracker helps you correct problems long before a disastrous failure occurs.

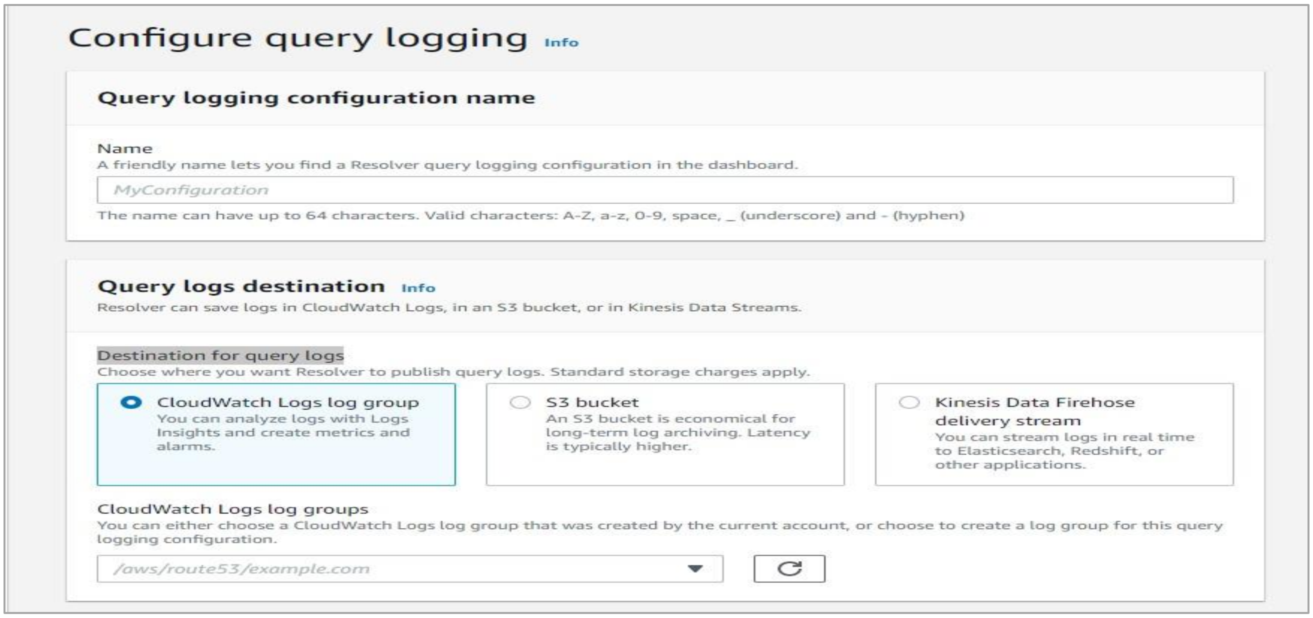
2. Prerequisites

- AWS Subscription
- EventTracker Public Manager IP

3. Integrating of AWS Route 53 with EventTracker

Note: We need to enable **DNS query logging** before sending logs.

1. Sign-in to **AWS Management** Console and open **Route 53** console at https://console.aws.amazon.com/route53/
2. In the navigation pane, choose **Hosted zones**.
3. Click on the hosted zone that you want to configure query logging for.
4. In the Hosted zone details pane, choose **Configure query logging**.
5. Choose an existing log group or create a new log group.



Configure query logging Info

Query logging configuration name

Name

A friendly name lets you find a Resolver query logging configuration in the dashboard.

MyConfiguration

The name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, space, _ (underscore) and - (hyphen)

Query logs destination Info

Resolver can save logs in CloudWatch Logs, in an S3 bucket, or in Kinesis Data Streams.

Destination for query logs

Choose where you want Resolver to publish query logs. Standard storage charges apply.

● CloudWatch Logs log group
You can analyze logs with Logs Insights and create metrics and alarms.

○ S3 bucket
An S3 bucket is economical for long-term log archiving. Latency is typically higher.

○ Kinesis Data Firehose delivery stream
You can stream logs in real time to Elasticsearch, Redshift, or other applications.

CloudWatch Logs log groups

You can either choose a CloudWatch Logs log group that was created by the current account, or choose to create a log group for this query logging configuration.

/aws/route53/example.com

Figure 1

6. In the **Destination for query logs,** choose **CloudWatch Logs log group** option.
7. If you receive an alert about permissions (this happens if you have not configured query logging with the new console before), do one of the following:
   - If you have 10 resource policies already, you cannot create any more. Select any of your resource policies and click **Edit**. Editing will give Route 53 permissions to write logs to your log groups. Click **Save**. Once the alert disappears and you can continue.
   - If you have never configured query logging before (or if you have not created 10 resource policies already), you need to grant permissions to Route 53 to write logs to your CloudWatch Logs groups. Choose Grant permissions. Once the alert disappears and you can continue.
8. Choose Permissions - **optional** to see a table that shows whether the resource policy matches the CloudWatch log group, and whether the Route 53 has the permission to publish logs to CloudWatch.
9. Click on **Configure query logging**.

Once we enabled query logging on route 53. We need to integrate CloudWatch with EventTracker using EventTracker lambda function.

3.1 Integrate CloudWatch with EventTracker using EventTracker lambda function

1. Click on **services** and select **lambda.**

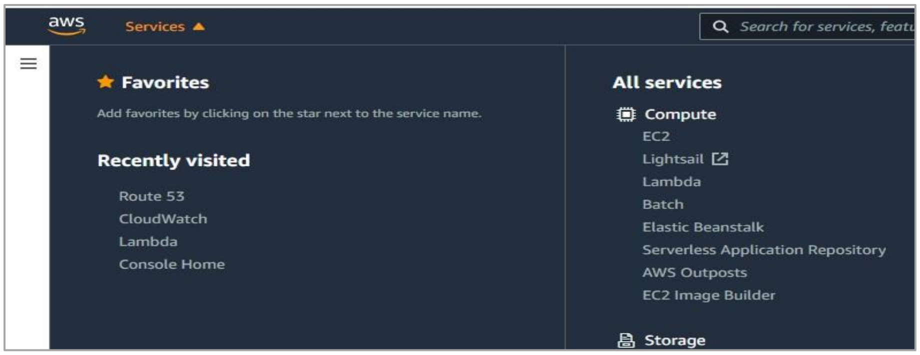


Figure 2

2. In the navigation pane choose **Functions**, then click on **create function**.

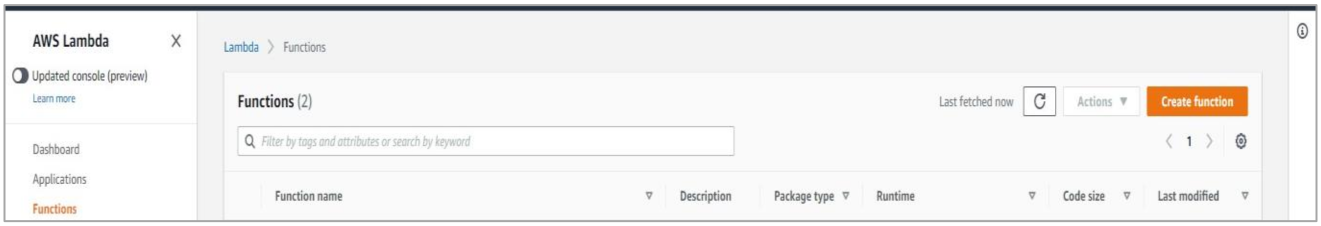
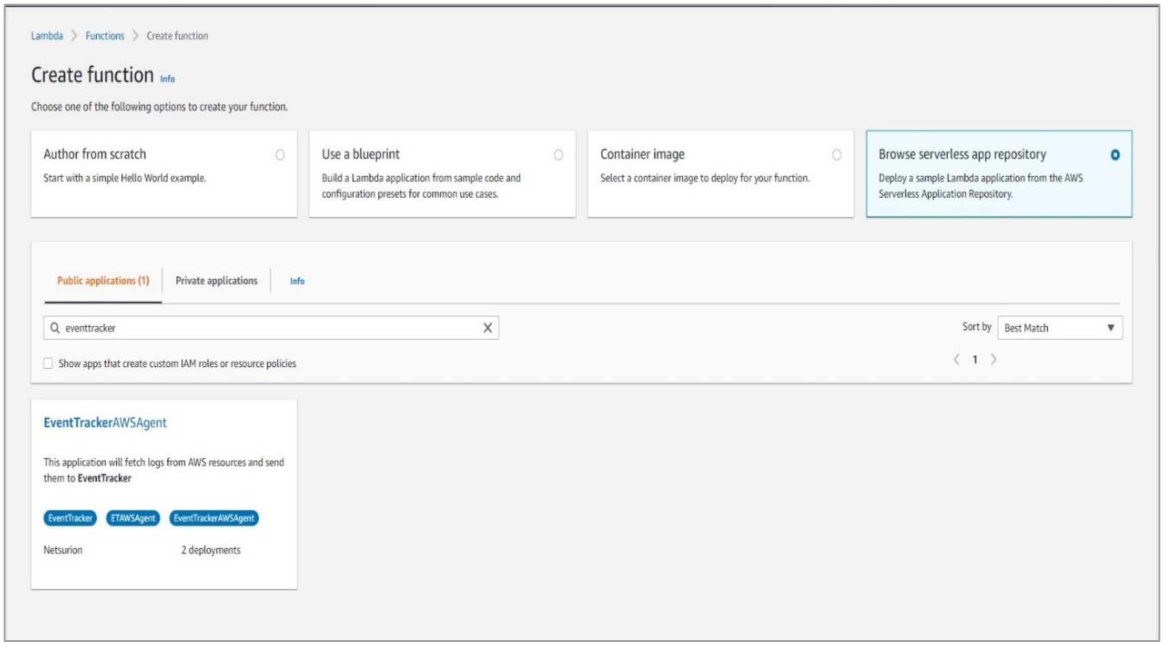


Figure 3

3. Select **Browse serverless app repository.**
4. Search **EventTracker** in public applications. You will get the **EventtrackerAWSAgent** in results.



Choose one of the following options to create your function.

○ Author from scratch
Start with a simple Hello World example.

○ Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

○ Container image
Select a container image to deploy for your function.

● Browse serverless app repository
Deploy a sample Lambda application from the AWS Serverless Application Repository.

eventtracker

Sort by Best Match

□ Show apps that create custom IAM roles or resource policies

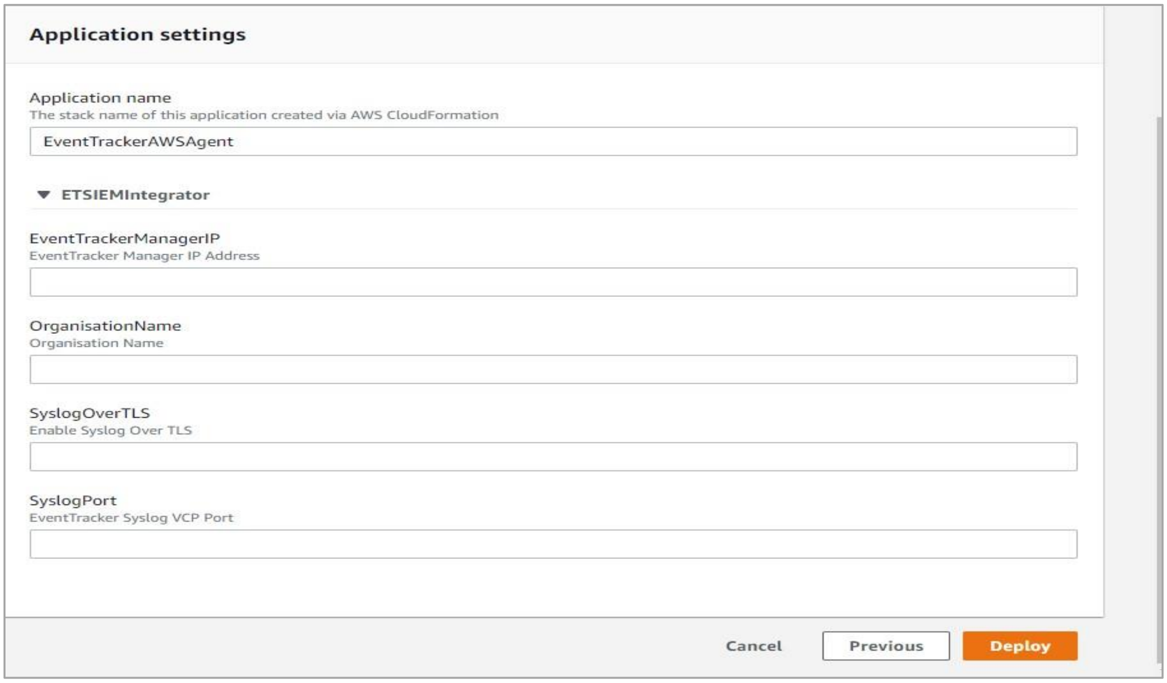
EventTrackerAWSAgent

This application will fetch logs from AWS resources and send them to EventTracker

Netsurion 2 deployments

Figure 4

5. Fill the details and click on **deploy**.



Application settings

Application name
The stack name of this application created via AWS CloudFormation
EventTrackerAWSAgent

▼ ETSIEMIntegrator

EventTrackerManagerIP
EventTracker Manager IP Address

OrganisationName
Organisation Name

SyslogOverTLS
Enable Syslog Over TLS

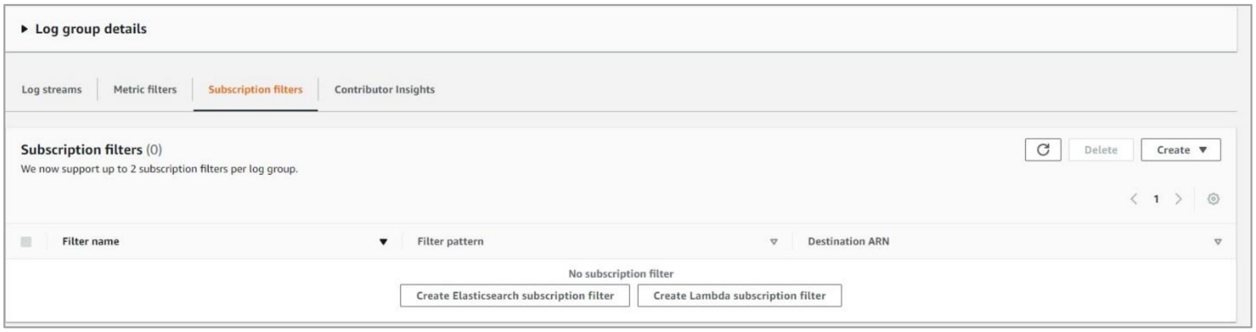
SyslogPort
EventTracker Syslog VCP Port

Figure 5

6. Enter the EventTracker Public Manager IP.
7. Enable syslog obver TLS as **True** or **False.**
8. Enter the syslog port.
9. After you click deploy, a function is created.

3.2 Create Subscription Filters

1. Click on **services** and select **CloudWatch.**
2. In the navigation pane, choose **log group**.
3. Click on the **log group** provided while creating **query logging**.
4. Go to **subscription filter**.



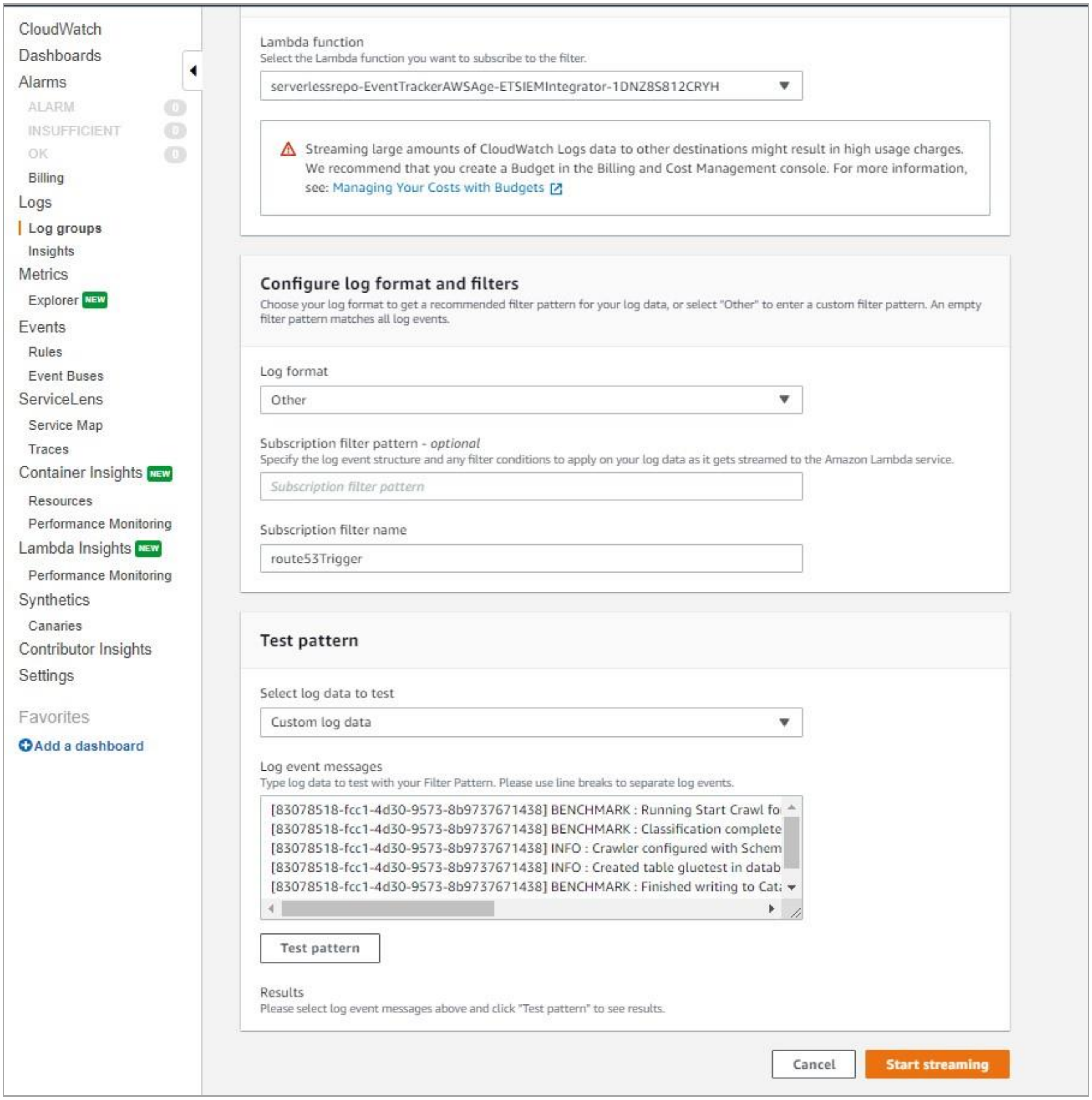
Subscription filters (0)

We now support up to 2 subscription filters per log group.

No subscription filter

Figure 6

5. Click on **create lambda subscription filter**.
6. Under lambda function, select the lambda function (created after deploying the application) created from the dropdown.
7. Enter subscription filter name, i.e. **route53Trigger**.
8. Click on **start streaming**.



Lambda function
Select the Lambda function you want to subscribe to the filter.
serverlessrepo-EventTrackerAWSAge-ETSIEMIntegrator-1DNZ8S812CRYH

⚠ Streaming large amounts of CloudWatch Logs data to other destinations might result in high usage charges. We recommend that you create a Budget in the Billing and Cost Management console. For more information, see: Managing Your Costs with Budgets

Configure log format and filters

Choose your log format to get a recommended filter pattern for your log data, or select "Other" to enter a custom filter pattern. An empty filter pattern matches all log events.

Log format: Other

Subscription filter pattern - *optional*
Specify the log event structure and any filter conditions to apply on your log data as it gets streamed to the Amazon Lambda service.
Subscription filter pattern

Subscription filter name: route53Trigger

Test pattern

Select log data to test: Custom log data

Log event messages
Type log data to test with your Filter Pattern. Please use line breaks to separate log events.

```
[83078518-fcc1-4d30-9573-8b9737671438] BENCHMARK : Running Start Crawl fo
[83078518-fcc1-4d30-9573-8b9737671438] BENCHMARK : Classification complete
[83078518-fcc1-4d30-9573-8b9737671438] INFO : Crawler configured with Schem
[83078518-fcc1-4d30-9573-8b9737671438] INFO : Created table gluetest in datab
[83078518-fcc1-4d30-9573-8b9737671438] BENCHMARK : Finished writing to Cat:
```

Results
Please select log event messages above and click "Test pattern" to see results.

Figure 7

Integration is complete. CloudWatch logs will be sent to Eventtracker.