



sumo logic

Observability Fundamentals

Student Lab Guide

Rev 08.09.2024 K

Disclaimer

This course material is Sumo Logic, Inc. confidential information. Sumo Logic, Inc. provides it as-is solely for your use and assessment as an individual partaking in the training and certification program on our Learning Portal and in specified other venues. The course material may not be reproduced, sold, or otherwise processed or transferred in any other way or for any other purpose without prior written permission from Sumo Logic. Sumo Logic owns the copyright and other intellectual property rights in the text, graphics, information, designs, data, and other content on this website (including exam materials and certifications as well as audio files and their scripts) with the exception of our partners', licensors', and other third parties' trademarks and other intellectual property. United States export control laws and regulations apply to this material, and you agree to comply with such laws and regulations.

Lab 0: Log in to the training environment

You'll need **two different browser** windows open for the lab activities. In one browser, you will be logged in as yourself, in **your own account**. In another browser, you'll be logged into a **Sumo Logic training lab environment**.

The training lab environment is separate from your other accounts. So, you'll need to create another login. To access the training lab environment:

1. Open a new window in a different browser or incognito window.
2. Navigate to <https://service.sumologic.com> in the new browser window.
3. Choose a number between 001 and 999. Remember this number, since you'll use it in all your labs.
4. Enter **training+analyst###@sumologic.com** in the **Email** field. Replace **###** with the number you chose in Step 3.
5. Enter the **Password** provided to you by your instructor.

Note: The password changes monthly. You can find the password on the training homepage for more information.

If you're reading this on Chrome, open a Firefox window. Using separate browsers will keep you logged in to your regular Sumo Logic account and the training lab environment at the same time. If you don't have two different browsers on your machine, you can open a Private or Incognito window instead.

In one browser, log in to **your own account**. Use this browser to navigate the lab activities, and take the exam. This way, you'll get credit for taking the course and passing the exam. Make sure you're logged in as yourself by clicking the profile icon in the upper right corner.

In another browser, log in to a **Sumo Logic training lab environment**. You can explore this environment freely, without impacting your dev, prod, or trial environments. Our lab activities are designed for the training lab environment, and may not work in other Sumo Logic accounts.

Note: The training accounts are public. Be careful what kind of personal information you share, like your real name or email address. These training accounts are wiped clean weekly, so make sure you complete any lab activities and save any data you want to keep, as you may not be able to recover it later.

Lab 1: Finding the root cause of latency using Log Search

In this lab, you'll learn how dashboards created based on metrics are instrumental in identifying the root cause of an outage. Once you've identified a potential latency issue, you'll investigate it further using Log Search.

Note: All labs assume you're using a **training+analyst###@sumologic.com** account. The data you see may vary depending on your environment if you're using your own credentials or a Sumo Logic trial account instead of a training account. To access a training account, review **Lab 0**.

Here's the scenario you are trying to troubleshoot and isolate the root cause.

You, a DevOps specialist for your company, are supporting an app on AWS that uses following AWS services:

- ELB (Application Load Balancer)
- Host/EC2
- DynamoDB

Your company app is reporting unusual latency in loading the app web page, along with occasional outages in the US area. You need to find out the root cause of the latency and outages ASAP.

The lab begins:

1. In the Sumo Logic UI, use the left nav menu to navigate to **Observability > AWS Observability**.

The screenshot shows the Sumo Logic interface. On the left, there's a sidebar with icons for Home, Library, Recent, Favorites, Logs, Dashboards, Observability (which is selected and highlighted in blue), Cloud SIEM, App Catalog, and Alerts. Below the sidebar is a link to 'Return to classic UI'. The main content area is titled 'Observability' and contains several sections: Metrics, Metrics Search, All Metric Searches, Infrastructure Monitoring, Kubernetes, AWS Observability (which is highlighted with a red box), Application Monitoring, Applications, Services, Transaction Traces, Span Analytics, Real User Monitoring, User Experience, and User Transactions.

2. In the left pane, click the **us-east-1** region under the **Prod** account.
3. You will see a number of separate AWS services listed under **us-east-1**. Locate and click the entry for **aws/dynamodb**

Explore By AWS Observability

- prod
 - us-east-1
 - aws/apigateway namespace
 - aws/applicationlb namespace
 - aws/dynamodb namespace**
 - aws/ec2 namespace
 - aws/ecs namespace
 - aws/elasticache namespace
 - aws/elb namespace
 - aws/lambda namespace
 - aws/networklb namespace
 - aws/rds namespace
 - aws/sns namespace
 - aws/sqs namespace
 - us-east-2
 - 351617048536 account
 - 524597330176 account
 - dev account
 - mobilebankingprod account
 - stage account
 - teststack account

Dashboards 1. AWS Region Overview

account prod region us-east-1

Activity Overview

Incoming Activity Locations

Application Load Balancer (ALB)

Requests Served by ALB

ALB Errors

EC2

4. In the right pane, in the Dashboards dropdown on top, select **AWS DynamoDB - Events of Interest**.

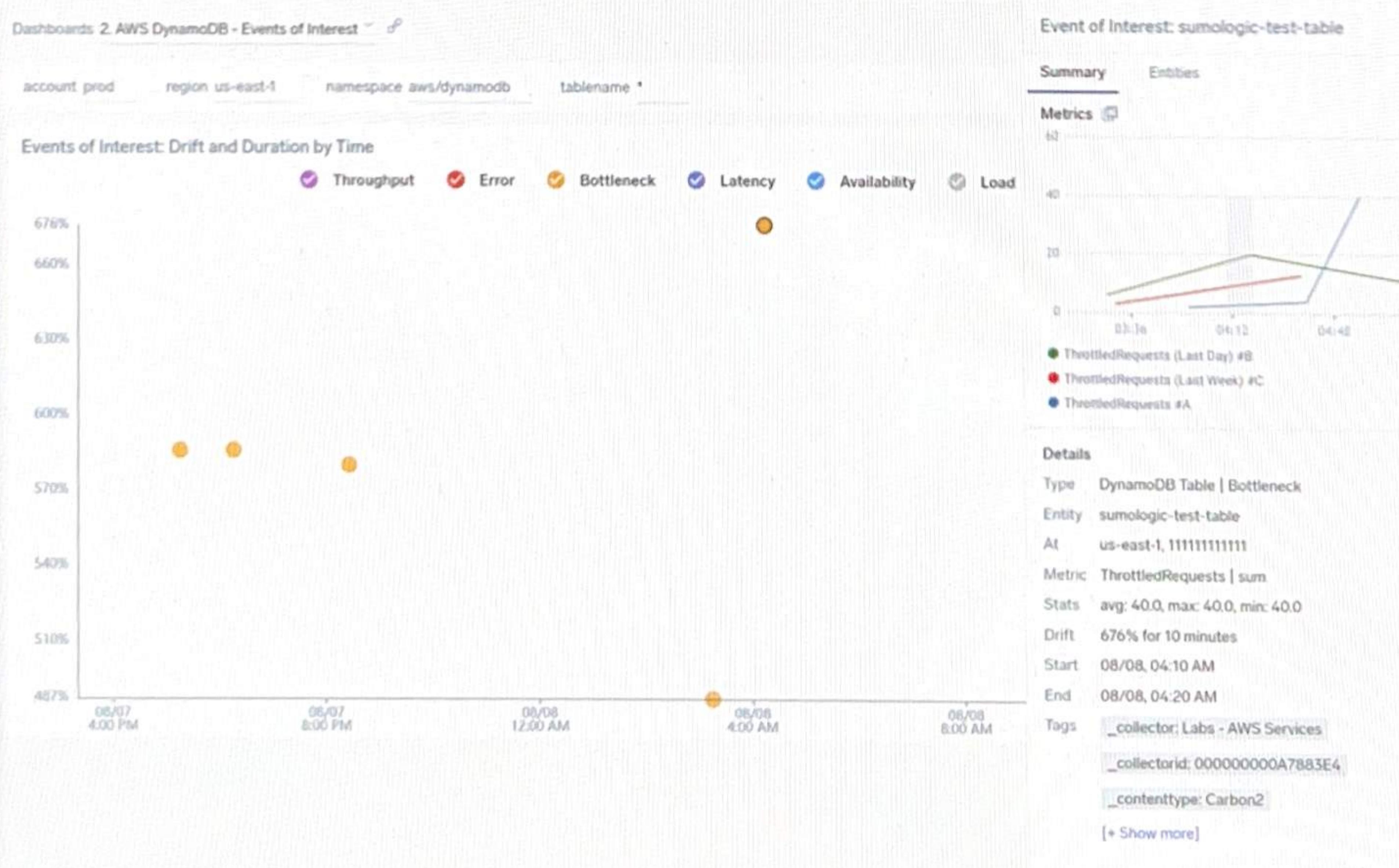


5. In this view you'll see a graph displaying "events of interest" that have been recorded in the Dynamo DB, such as anomalous errors, latency spikes, or bottlenecks. You should see one or more yellow circles representing detected bottlenecks in DB processing.



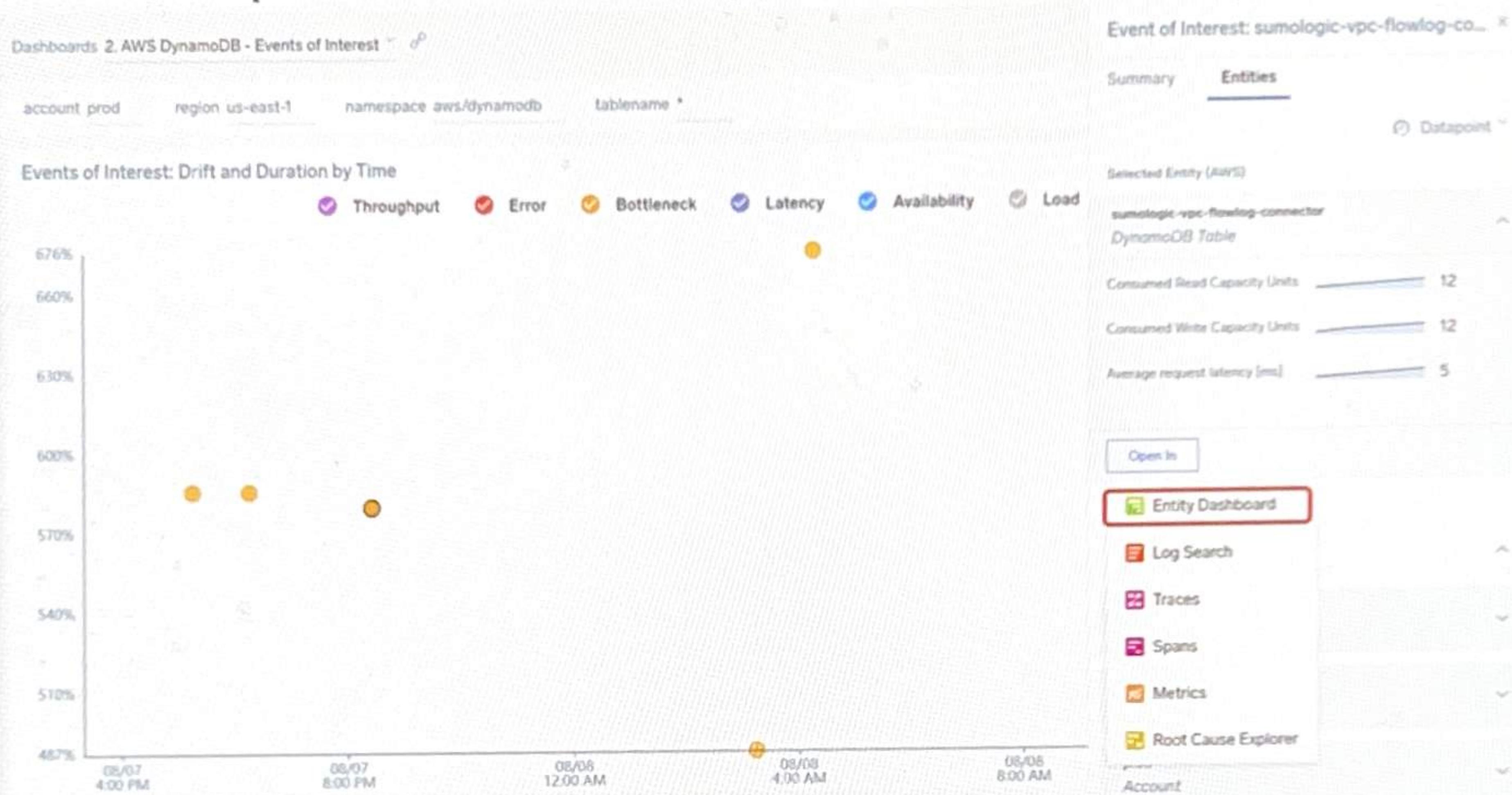
The X-axis represents time, and the Y-axis represents the "severity" of the event (the percentage increase between the amount of errors recorded at that time versus "normal" operation).

6. Click on any of the yellow circles and open up the right sidebar window with more information.

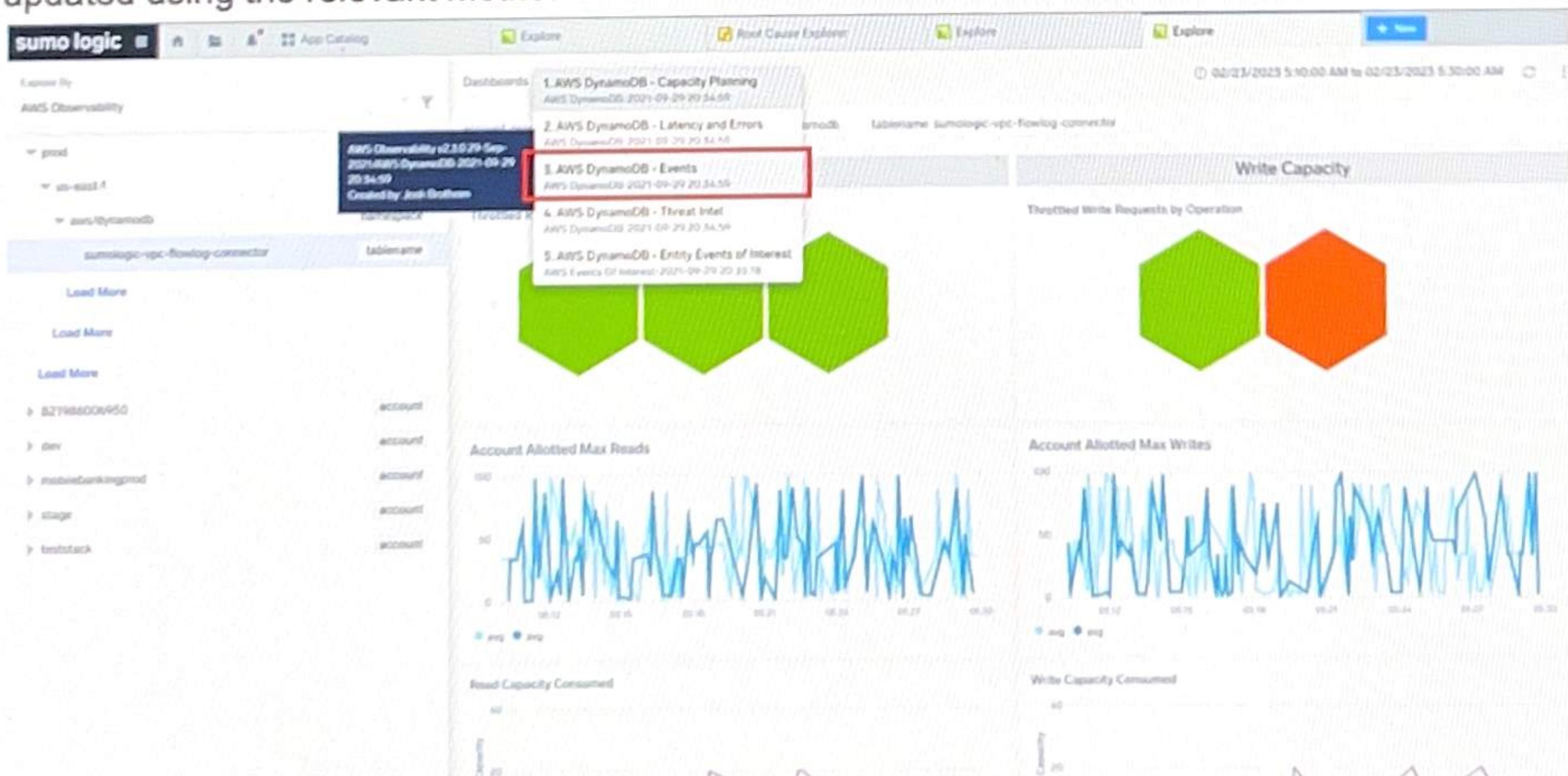


7. The information here indicates that DB requests are being throttled for some reason. Now we want to dig into how and why this is happening. Click on the **Entities** tab in the right sidebar.

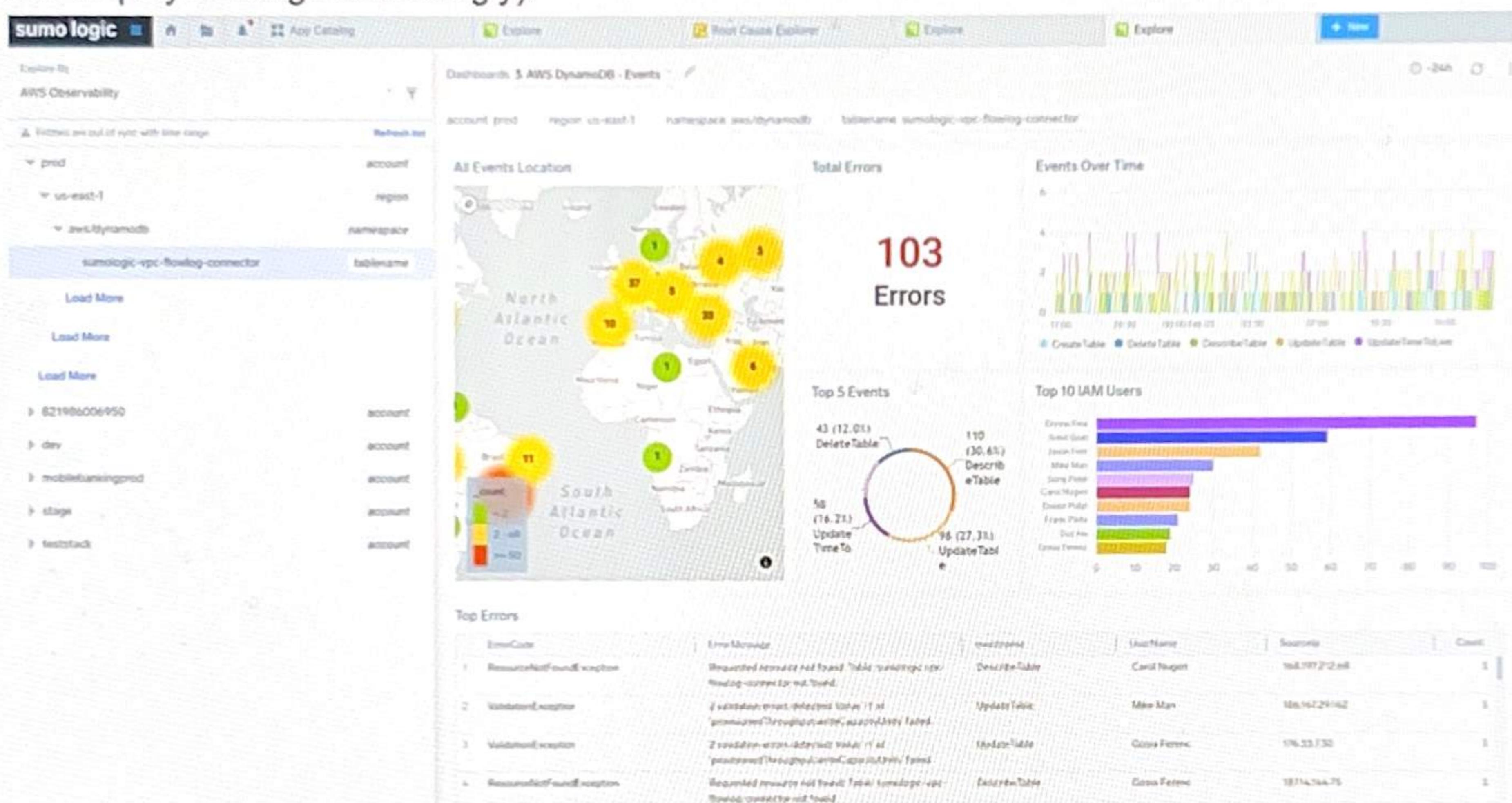
8. Then click the **Open In** button and select **Entity Dashboard**.



9. On the following screen, open the relevant DynamoDB Events dashboard by clicking the **AWS DynamoDB - Events** option in the dropdown. The data in the dashboard will be updated using the relevant metrics.

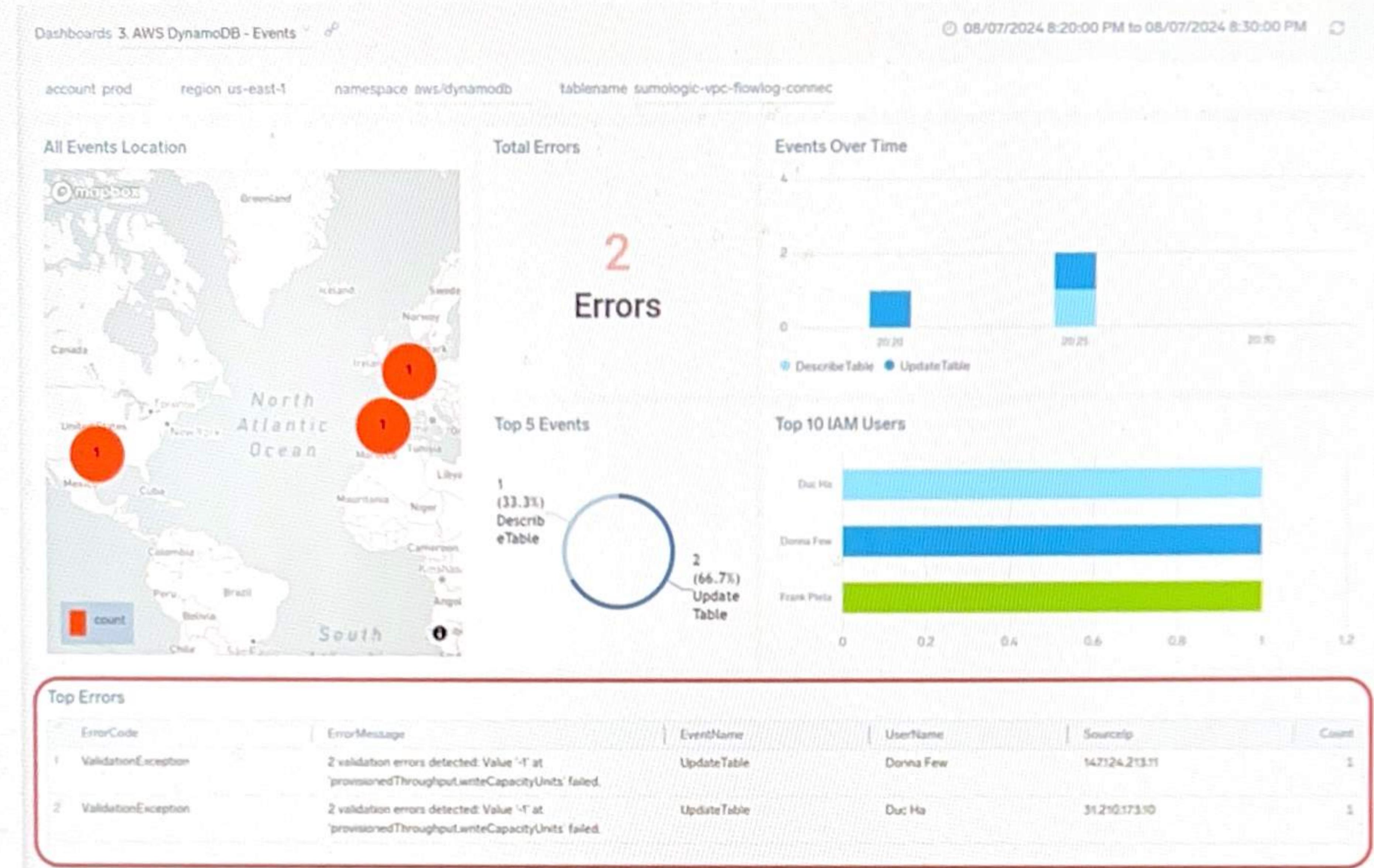


10. The dashboard displays a number of panels including No. of Errors, Top 5 Events, and Events over Time. You can see the total number of errors and also the associated events that occurred in the DynamoDB. (Note that you can change the time span for the dashboard in the upper right to expand or contract the time range to see how the metrics and displays change accordingly).



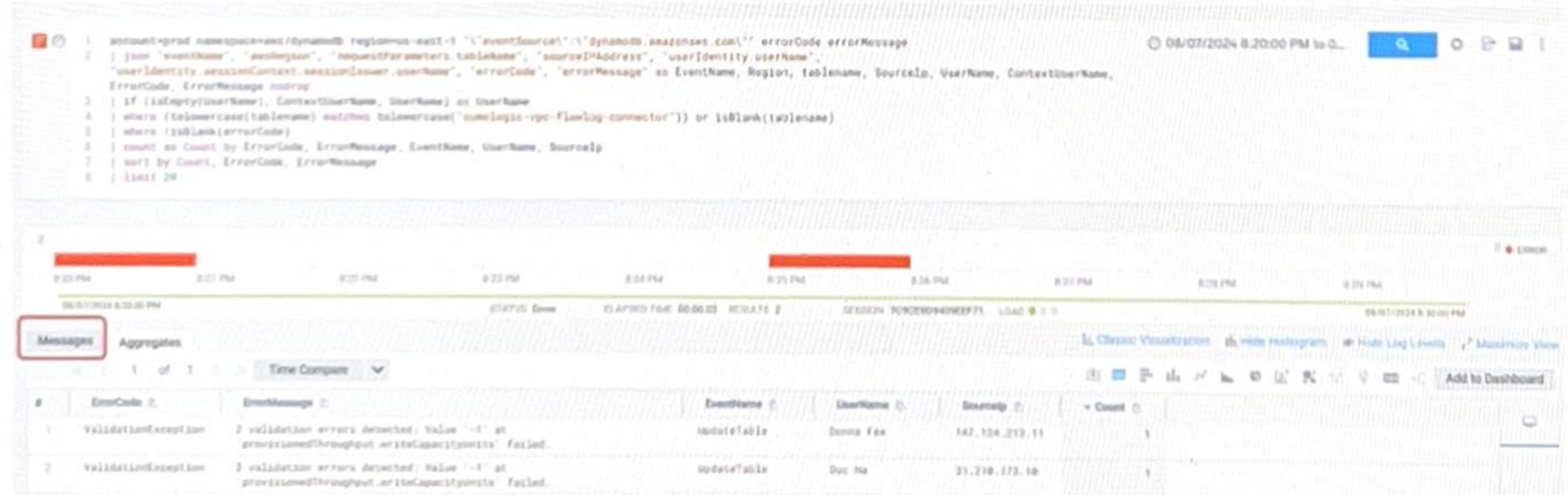
11. Once you find the relevant data and metrics in the events dashboard, you can drill down further into the associated logs. Scroll down to see the **Top Errors** table for clues as to

the root cause of the DB throttling.



12. Click the “three dot” icon in the top right corner of the **Top Errors** table, then select **Open in Log Search** to open the relevant logs.

13. When you open the event logs in Search mode, you will first see the **aggregate** events, where you see someone has updated the table. Click the **Messages** tab to view the individual logs.



14. Now all the relevant individual log entries are displayed. If desired, you can apply filters in the left pane to focus on specific fields to better search through the events you need to investigate. For instance, try deselecting the fields ‘**ContextUserName**’ or ‘**ErrorCode**’ with the checkbox and see how the data updates.

15. When you go through the resulting update table log events, you find that someone named **Mike Man** called `UpdateTable` and changed the read/write parameter (IOPS). (NOTE that the names in these training log entries are randomized so your log entry will likely have a different name attached)

Field	Value
ContextUserName	randomized
MFAAuthenticated	True
MessageDate	18/07/2021 18:03:24 UTC
ProvisionedThroughput	False
Region	us-east-1
TableName	randomized
Message	<pre>{ "DynamoDBAutoScaleRole": "updateTable" }</pre>

This developer had evidently reconfigured DynamoDB to use lower-provisioned IOPS (Input/Output Operations Per Second) which caused throttling and the subsequent outages. This will give your organization the clue to reconfigure the IOPS and the throttling issue should be resolved.

Lab 2: When are Logs the best bet in Observability?

In this lab, you'll learn that when alerted to an incident, diving deep into the logs can help you troubleshoot and resolve an issue.

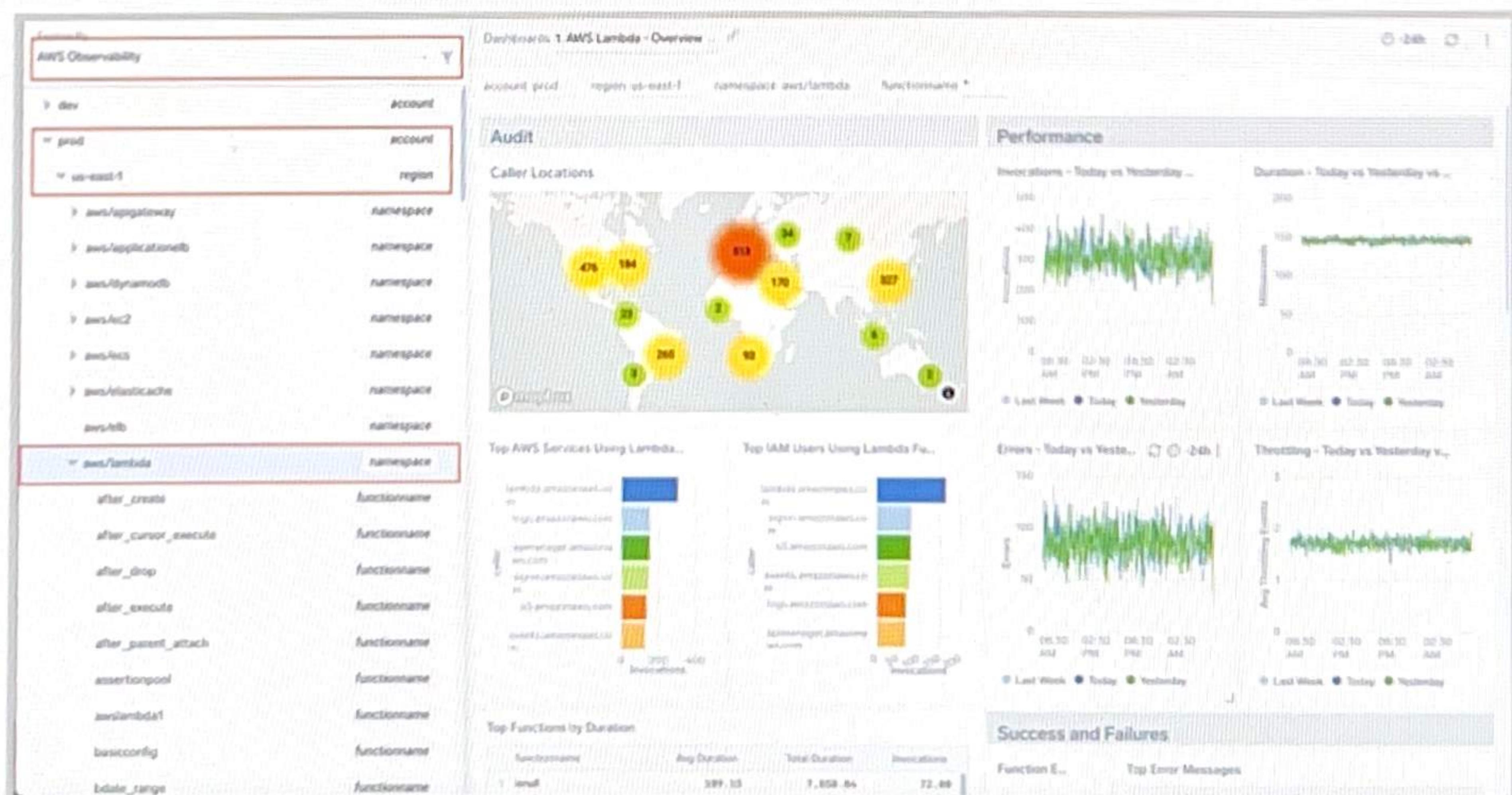
Note: All labs assume you're using a training+analyst###@sumologic.com account. The data you see may vary depending on your environment if you're using your own credentials or a Sumo Logic trial account instead of a training account. To access a training account, review **Lab 0**.

Here's the scenario you are trying to troubleshoot.

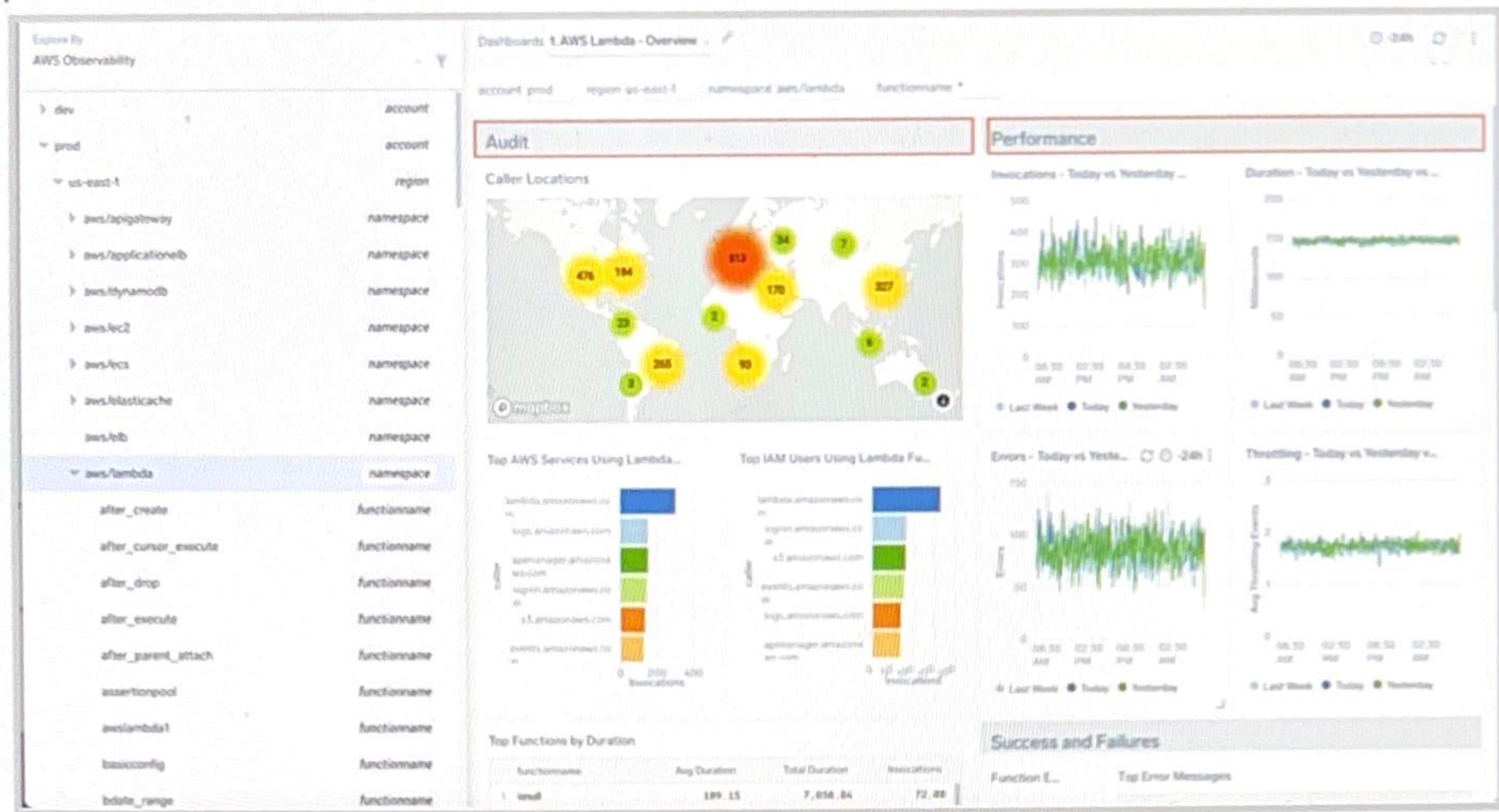
You are an SRE and you've got an alert on an incident. System end users on the **US east coast** have been experiencing an outage in the app in the **last 24 hours**. You've got an indication that the problem is on the **Lambda** server. Now you need to take a look to see when all of this started. What metrics are you seeing and how can you determine when this happened?

The lab begins.

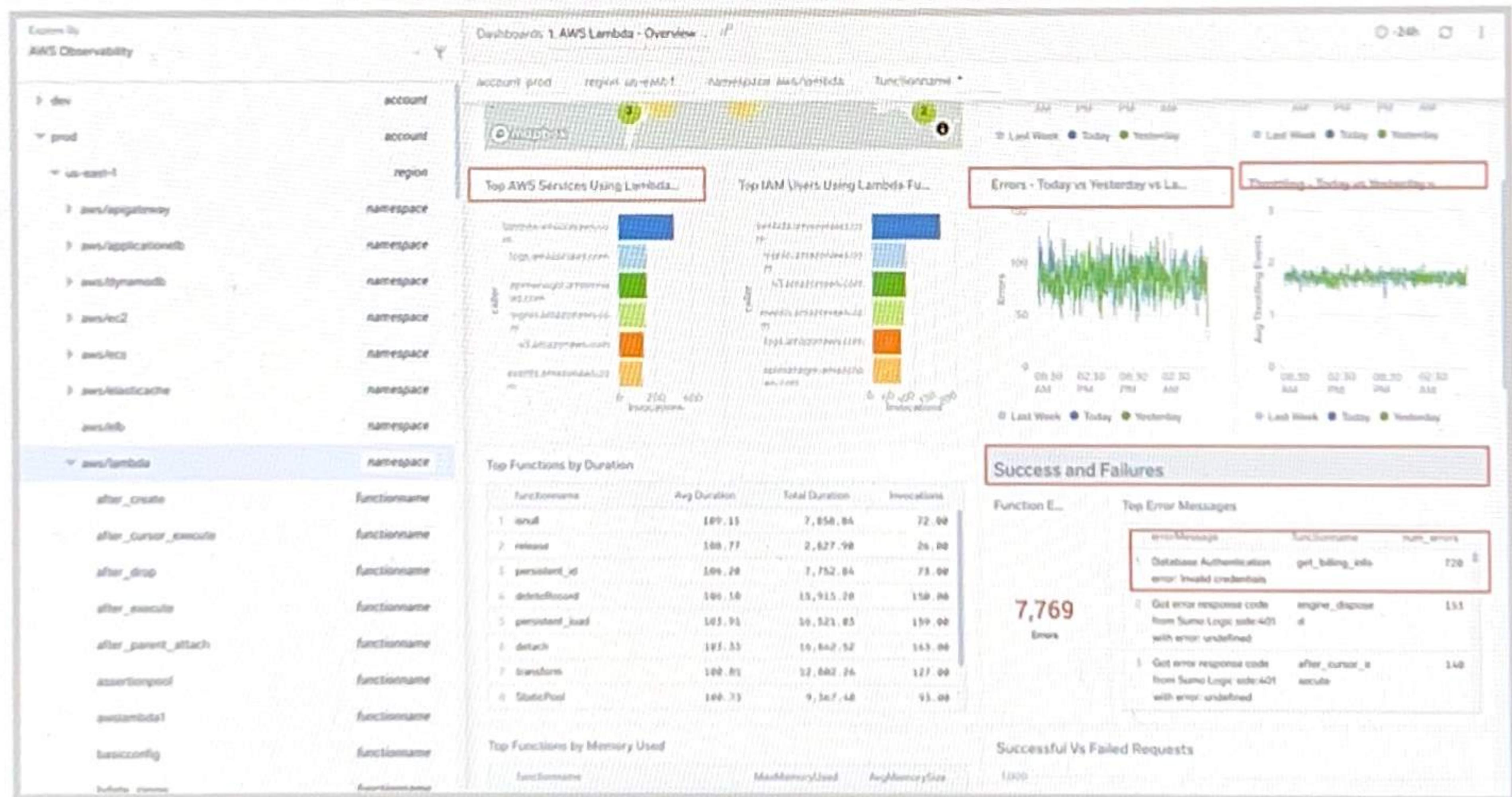
1. In the left tab menu of the Sumo UI, click **Observability**, then select AWS Observability.
2. In the left pane, click the **us-east-1** region under the **Prod** account, then click on **aws/lambda**.



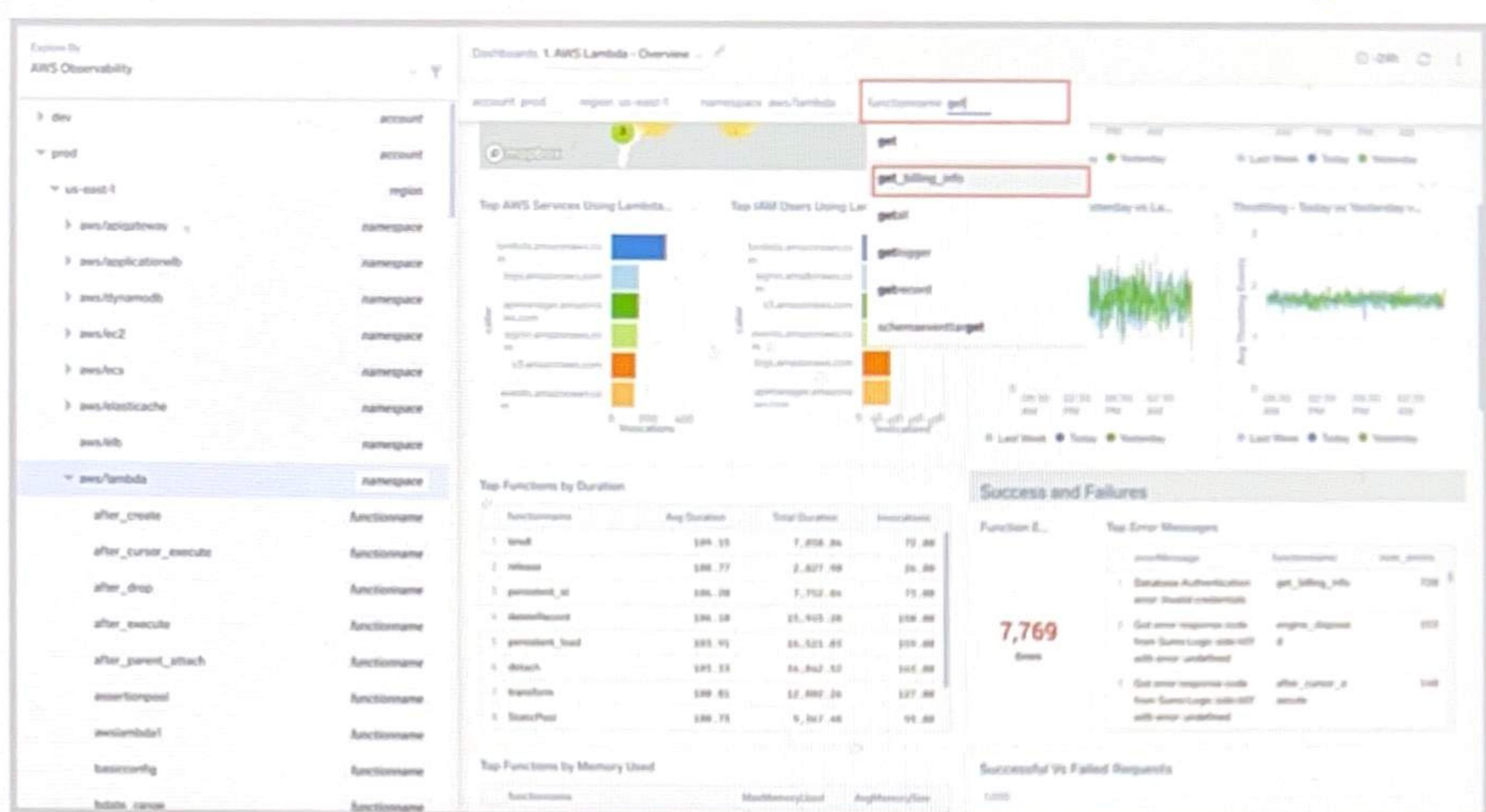
3. The AWS Lambda - Overview dashboard displays the Audit and Performance panels.



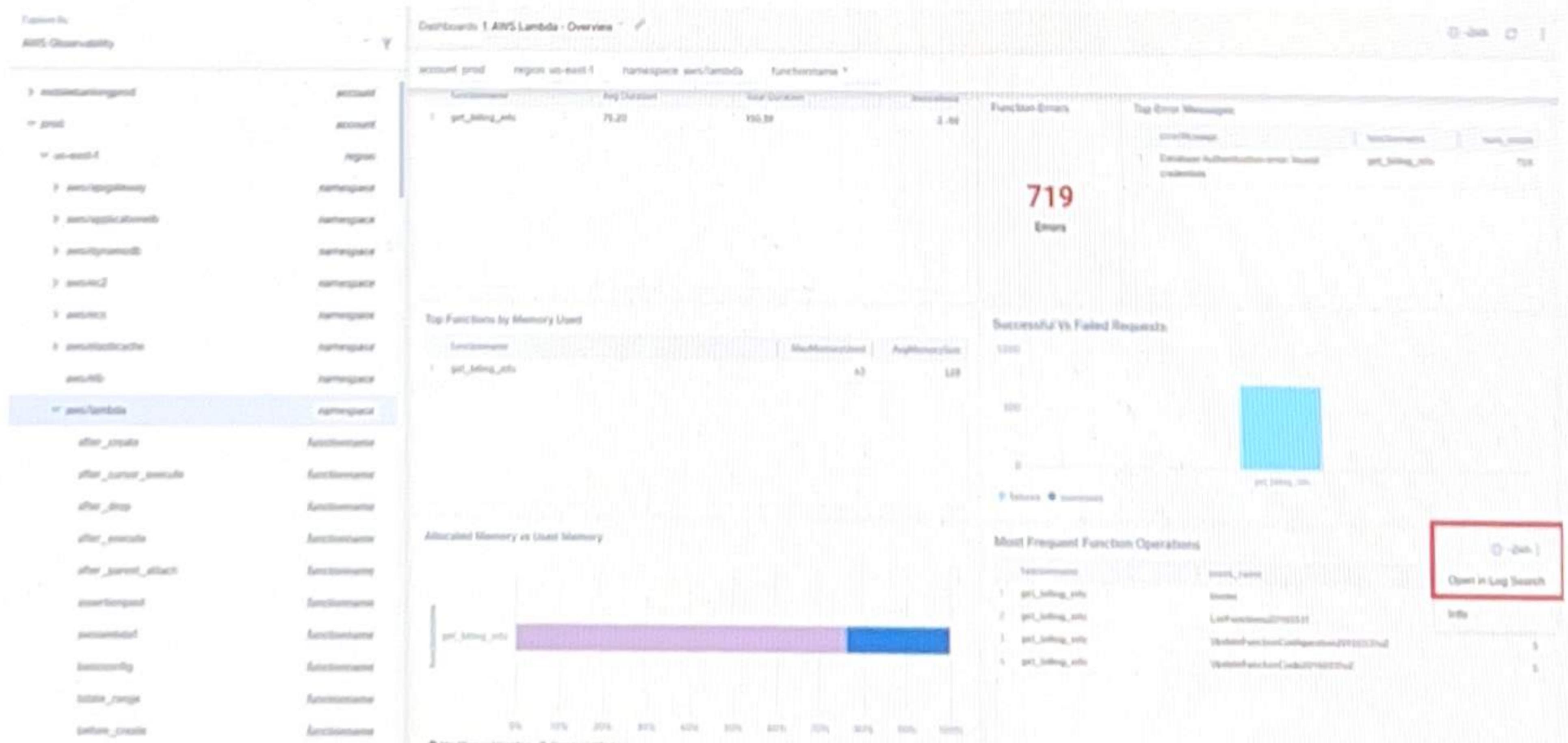
If you scroll down, you can examine the data to find out which functions are taking the longest to run along with which ones consume the highest amount of memory. Using the **Success and Failure** panel a little further down on the right, you can see that the `get_billing_info` function has recorded the highest number of errors in this time period. Sounds like a good lead to investigate!



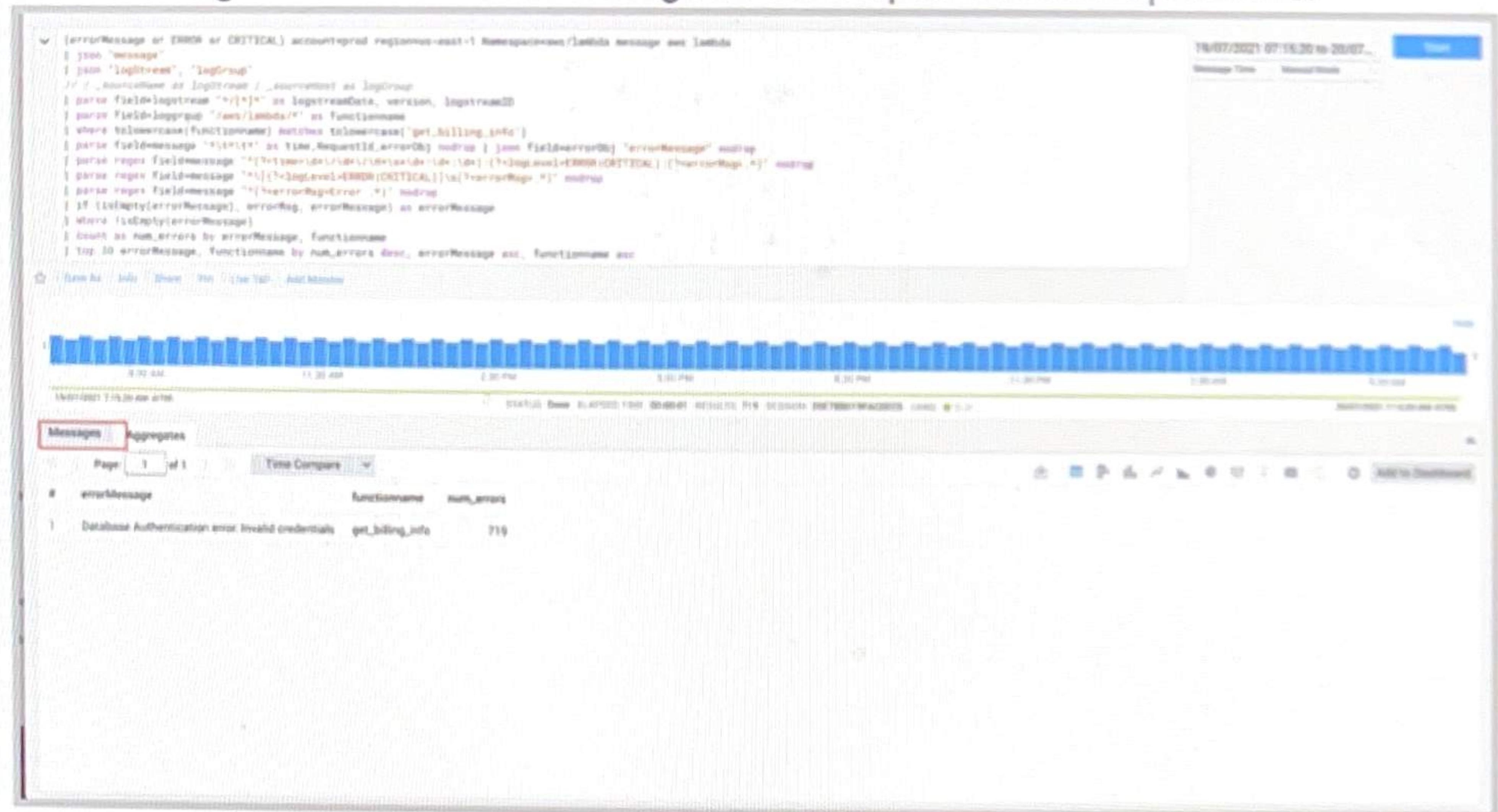
4. Apply a filter by selecting 'get_billing_info' in the **functionname** drop down on the top of the screen. Now we will see only the specific info related to calls to the 'get_billing_info' function.



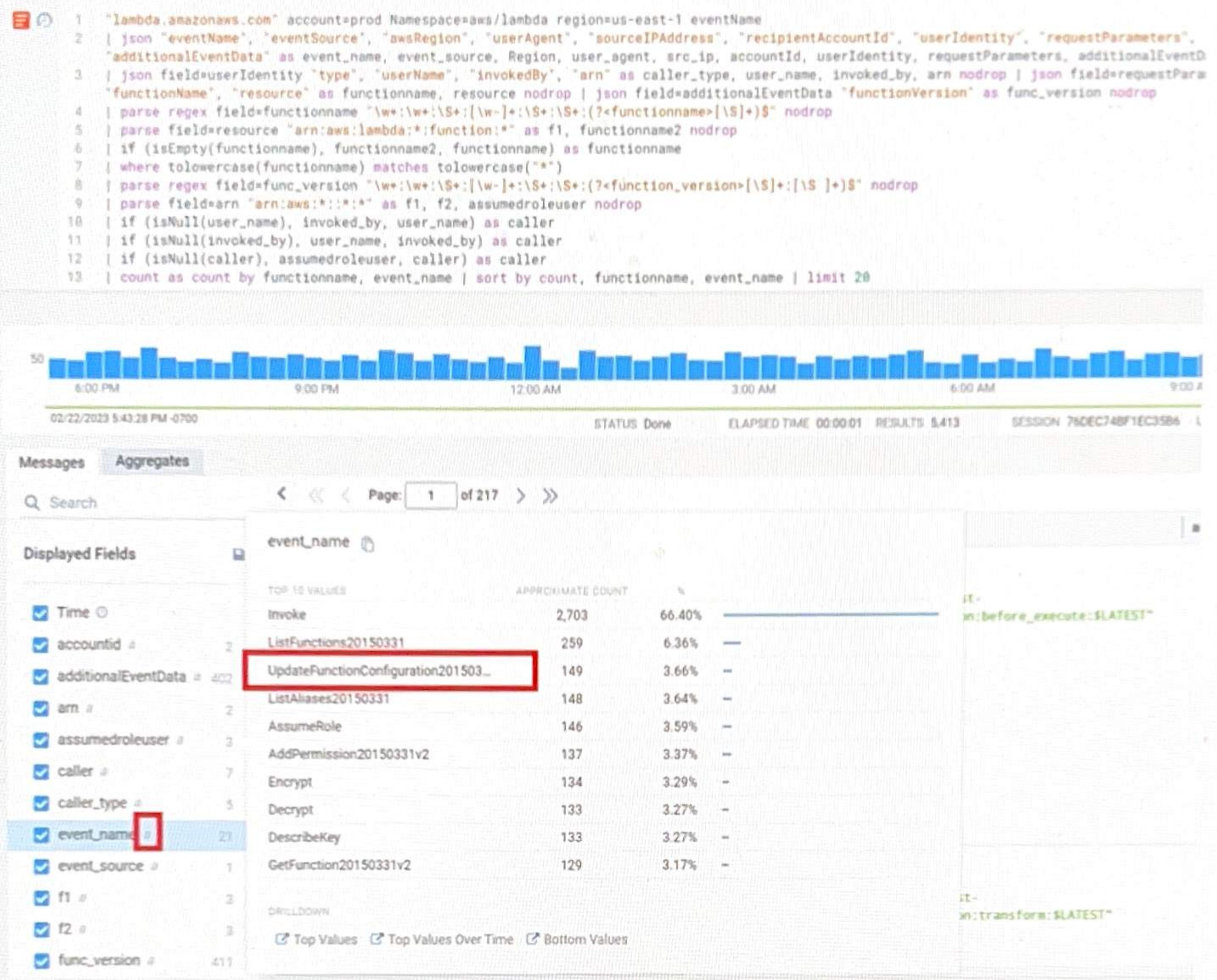
- In this case, we can see that there were 720 errors related to the 'get_billing_info' function just in the last 24 hours. (NOTE: your exact numbers may vary.)
- Locate the dashboard panel entitled "Successful Vs Failed Requests". Note here that it appears that 100% of the attempts to call 'get_billing_info' failed, indicating it is likely a fundamental configuration issue, not an occasional communication problem.
- Find the dashboard directly below called "Most Frequent Function Operations". Note that there are a few calls to "UpdateFunctionConfiguration" that could be the cause of our consistent errors due to misconfiguration. Click the 'triple dot' button on the top right of this panel, then select 'Open in Log Search' to open the related log events in the search mode.



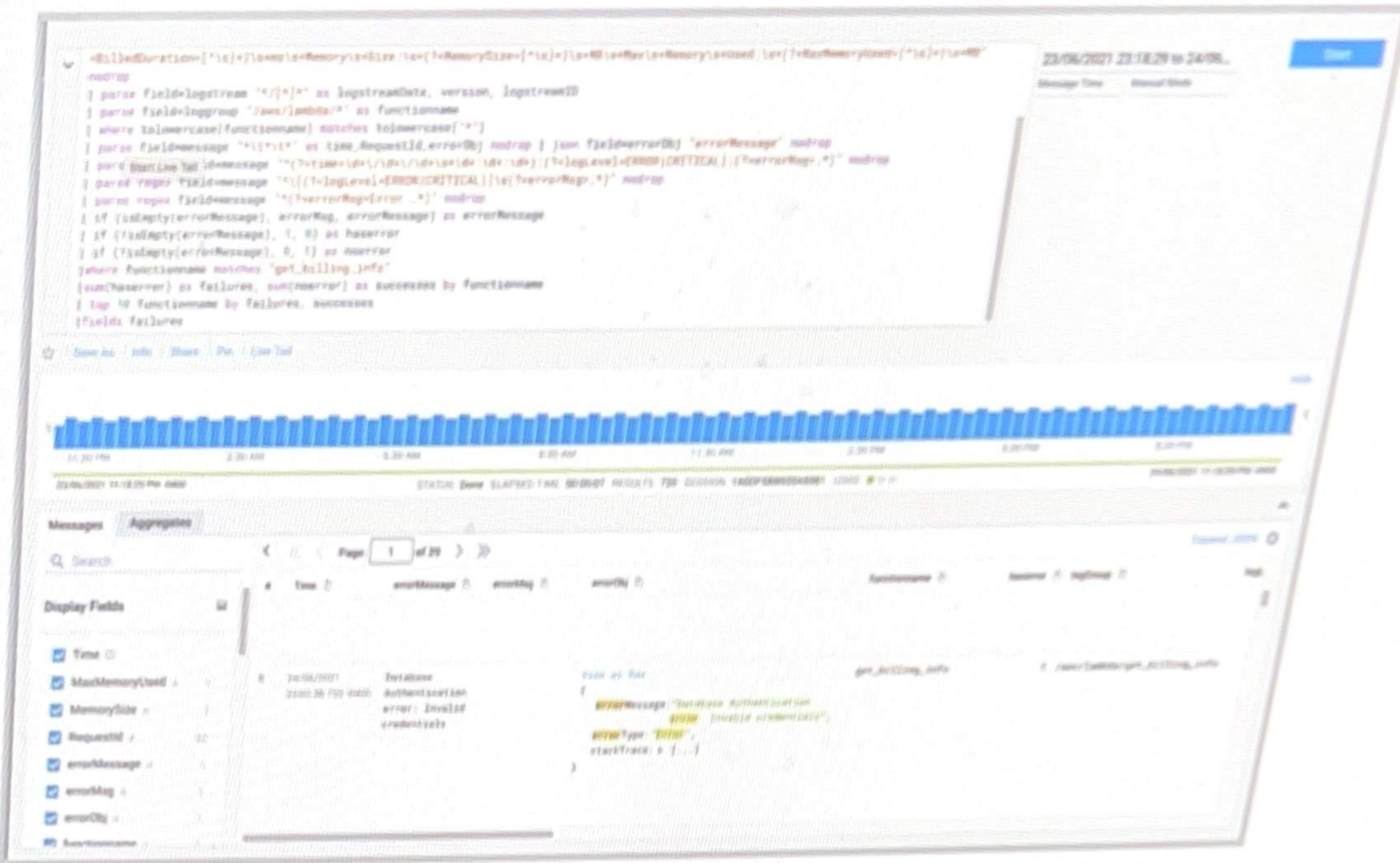
8. Click **Messages** to take a look at the logs for the frequent function operations.



9. Use a filter to view the specific entries for the `UpdateFunctionConfiguration` commands. In the **Displayed Fields** list, find the entry for "event_name" then click the small "a" next to the name. On the popup window, select the entry that starts with "UpdateFunctionConfiguration".



- Examine the resulting logs to find that one of the developers has been making the changes in the function configuration that likely caused the constant failures. You'll see in the filtered log messages that it was **Mark Smith** who was trying to change the configuration (click on the "a" next to the "user_name" to see the recorded user names for these log entries – note that training log names are randomized and you may see a different name here). We can now contact Mark to find out the reason why he was changing the details, and rectify the issue.



Lab 3: Working with Logs

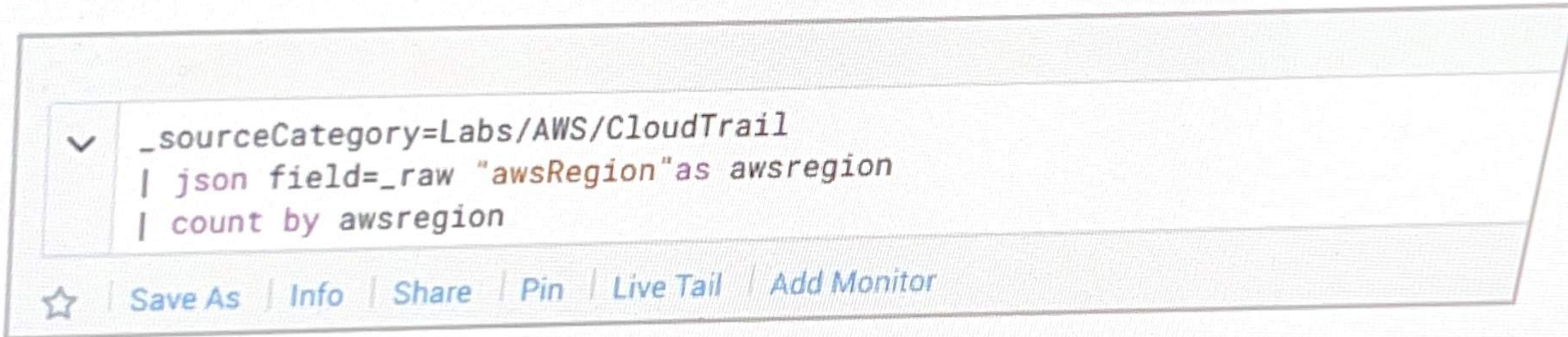
In this lab, you'll learn how to use some additional tools to filter and analyze logs, as well as explore the LogReduce® and LogCompare operators.

Note: All labs assume you're using a training+analyst###@sumologic.com account. The data you see may vary depending on your environment if you're using your own credentials or a Sumo Logic trial account instead of a training account. To access a training account, review [Lab 0](#).

The lab begins.

1. In the left tab menu of the Sumo UI, click **Logs**, then **Log Search**.
2. In the search query box, type or copy/paste the following query:

```
_sourceCategory=Labs/AWS/CloudTrail  
| json field=_raw "awsRegion"as awsregion  
| count by awsregion
```

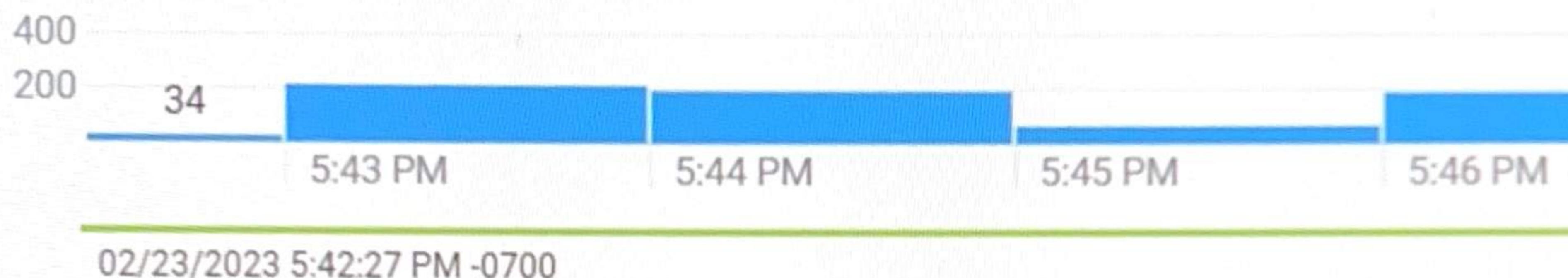


3. Click **Start** to execute the query.

The Aggregate panel shows the count by region.



```
1 _sourceCategory=Labs/AWS/CloudTrail  
2 | json field=_raw "awsRegion" as awsregion  
3 | count by awsregion
```

**Messages****Aggregates**

Page: 1 of 1



Time Compare



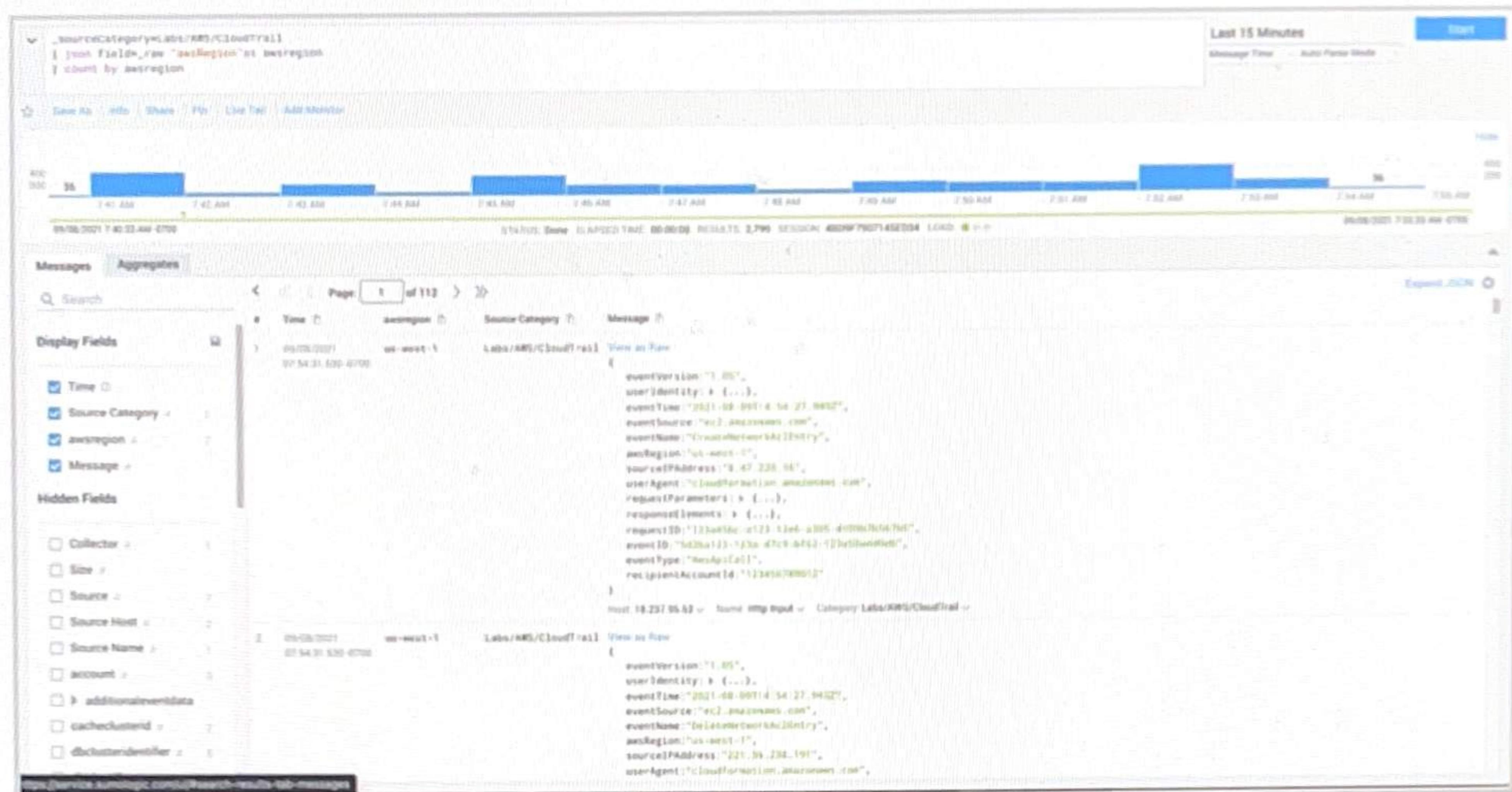
#

awsregion

_count

1	us-east-3	13
2	us-west-3	13
3	us-west-2	385
4	us-east-2	436
5	us-west-1	1,277
6	us-east-1	449
7	sa-east-1	58

4. Click **Messages**. The individual Log entries will be displayed. Use the page navigation arrows to move within the log pages.



Use the **field browser** to filter the log entries as per your requirement.

LogReduce® Operator:

The LogReduce® algorithm uses fuzzy logic to group messages together based on string and pattern similarity. You can use the “logreduce” operator in queries to quickly assess activity patterns for things like a range of devices or traffic on a website. Focus the LogReduce® algorithm on an area of interest by defining that area in the keyword expression.

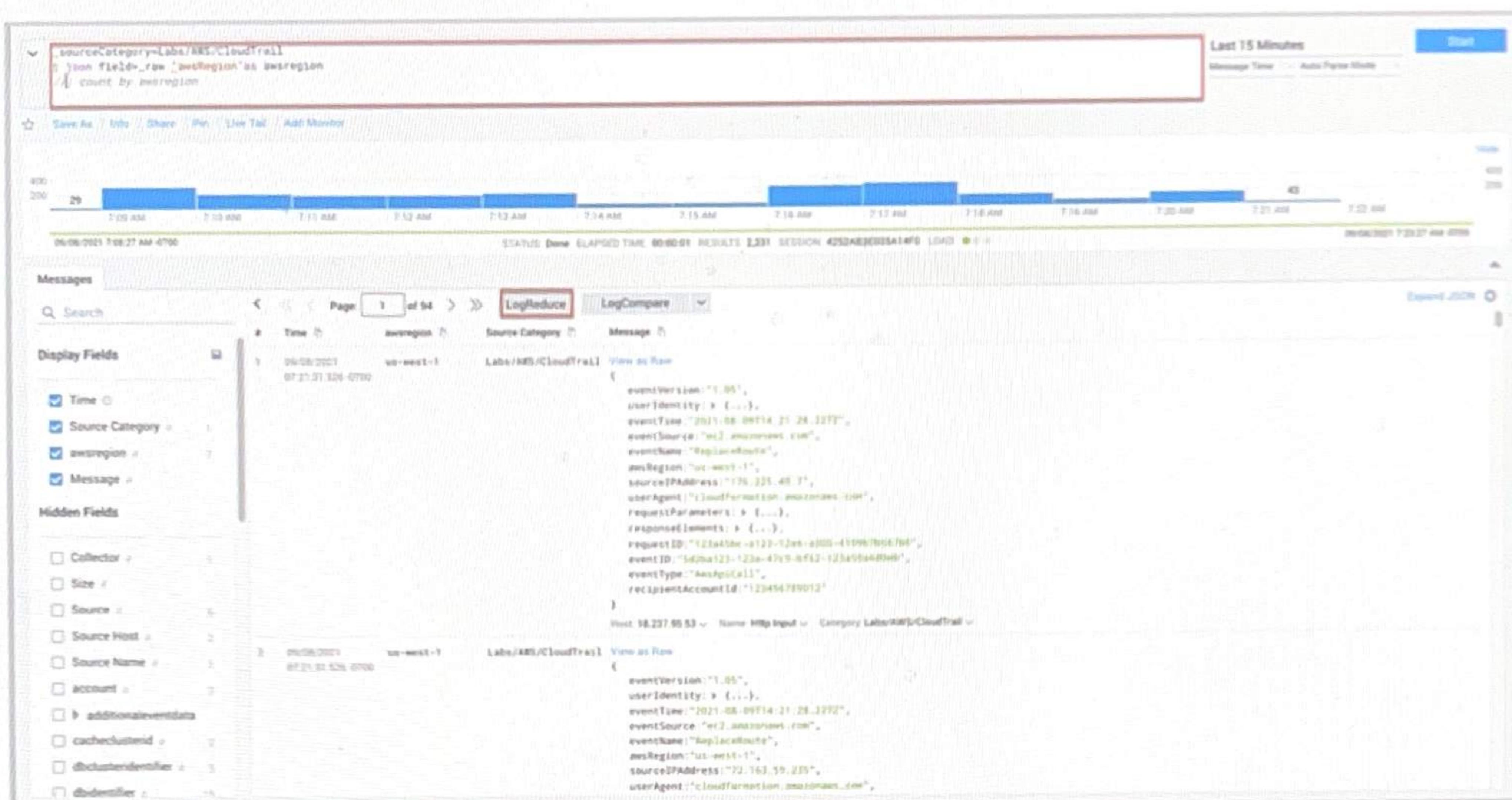
There are two ways to use the operator.

- Use the **LogReduce** button displayed on the results table after running a search.
- Manually add the “logreduce” operator to your query following its syntax.

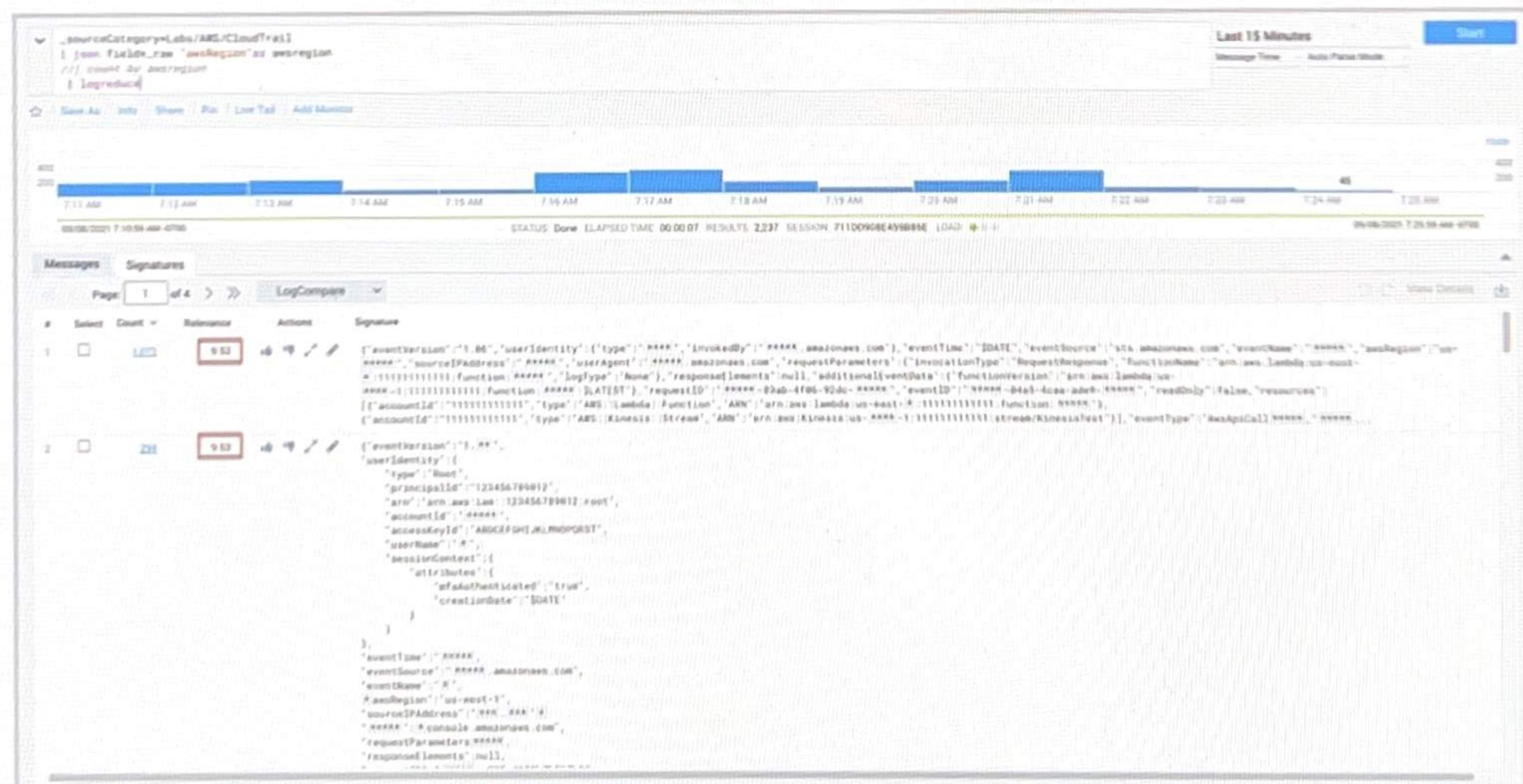
For our lab, we will use the LogReduce button.

1. First, modify our query to remove the ‘count by’ clause – the “logreduce” operator cannot be used with aggregation or group-by operators such as “count by field”.

```
_sourceCategory=Labs/AWS/CloudTrail
| json field=_raw "awsRegion" as awsregion
```



2. Click the LogReduce button.
3. The Signatures tab is displayed with the results. Note the "Relevance" column which indicates the category of log messages that are judged to be most relevant to the query.



We can further explore the functionality of LogReduce® which allows you to distill unique messages from the noise by identifying recurring Signatures in your data.

4. Search for all messages that have the word "error" across all your data, use:

```
error | logreduce
```

5. Then click on **LogReduce** to get a summarized view of all messages.

6. Next, in a new search, run LogReduce® on your Snort security data to identify unusual activity (i.e. intrusions).

```
_sourceCategory=labs/snort  
| logreduce
```

7. Sort your results by count to identify those that happen only once.

8. Click on the count (1) to view the unusual message.

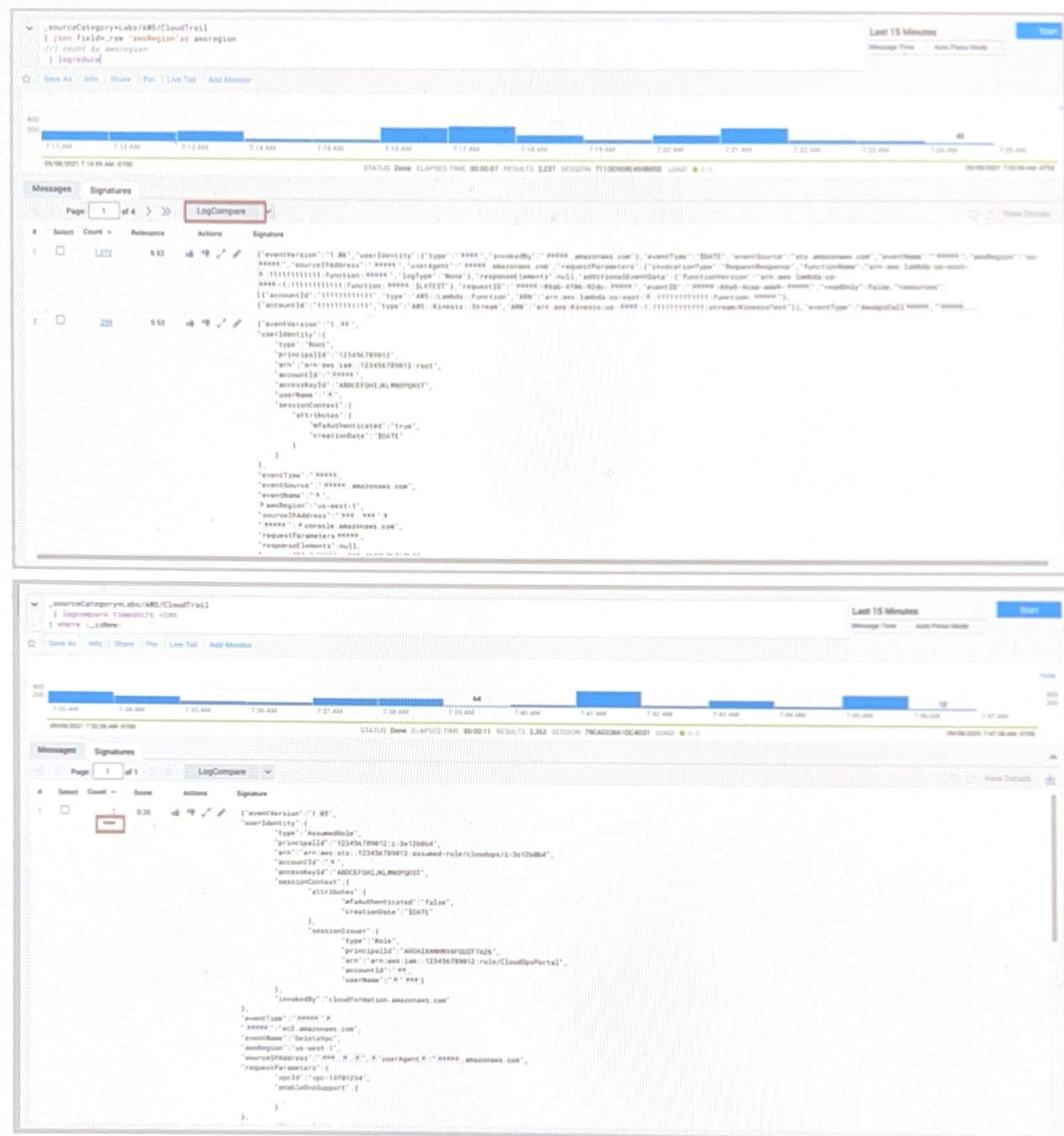
9. Now click on the host to view surrounding messages to identify the context of the intrusion.

"logcompare" Operator:

The "logcompare" operator allows you to compare log activity from two different time periods, providing you insight on how your current time compares to a baseline. In this case, we will use the logcompare operator to identify when signature messages deviate by more than 25% from the baseline.

1. Use logcompare to run a summarized query for a baseline 24 hours ago.

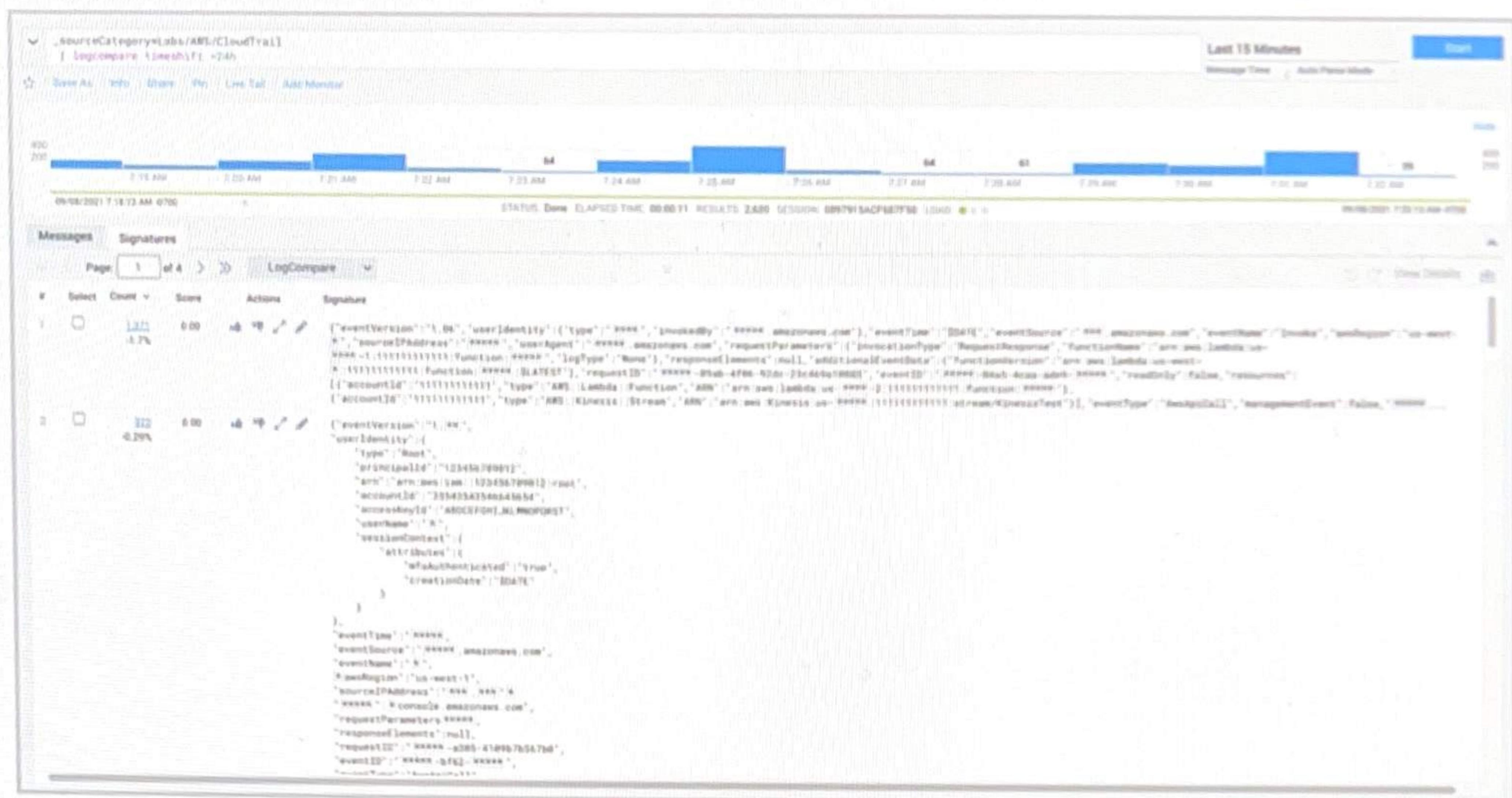
```
_sourceCategory=Labs/AWS/CloudTrail  
| logcompare timeshift -24h
```



2. Note that in some cases the delta (percentage change) is very small and not significant. To view only those results where the delta percentage is significant (let's say: greater than 25%), let's add a 'where' clause for `_deltaPercentage`.

```

    _sourceCategory=Labs/AWS/CloudTrail
    | logcompare timeshift -24h
    | where abs(_deltaPercentage) > 25
  
```



3. We can also look for entirely new messages that did not appear at all during the previous baseline time period. To view results for new Signature messages in the current time period, add a where clause for `_isNew`:

```
_sourceCategory=Labs/AWS/CloudTrail
| logcompare timeshift -24h
| where (_isNew)
```

