sumo logic
Academy

Logs For Security

Student Lab Guide

Rev 10.29.25.K

Table of Contents

Login to the training environment

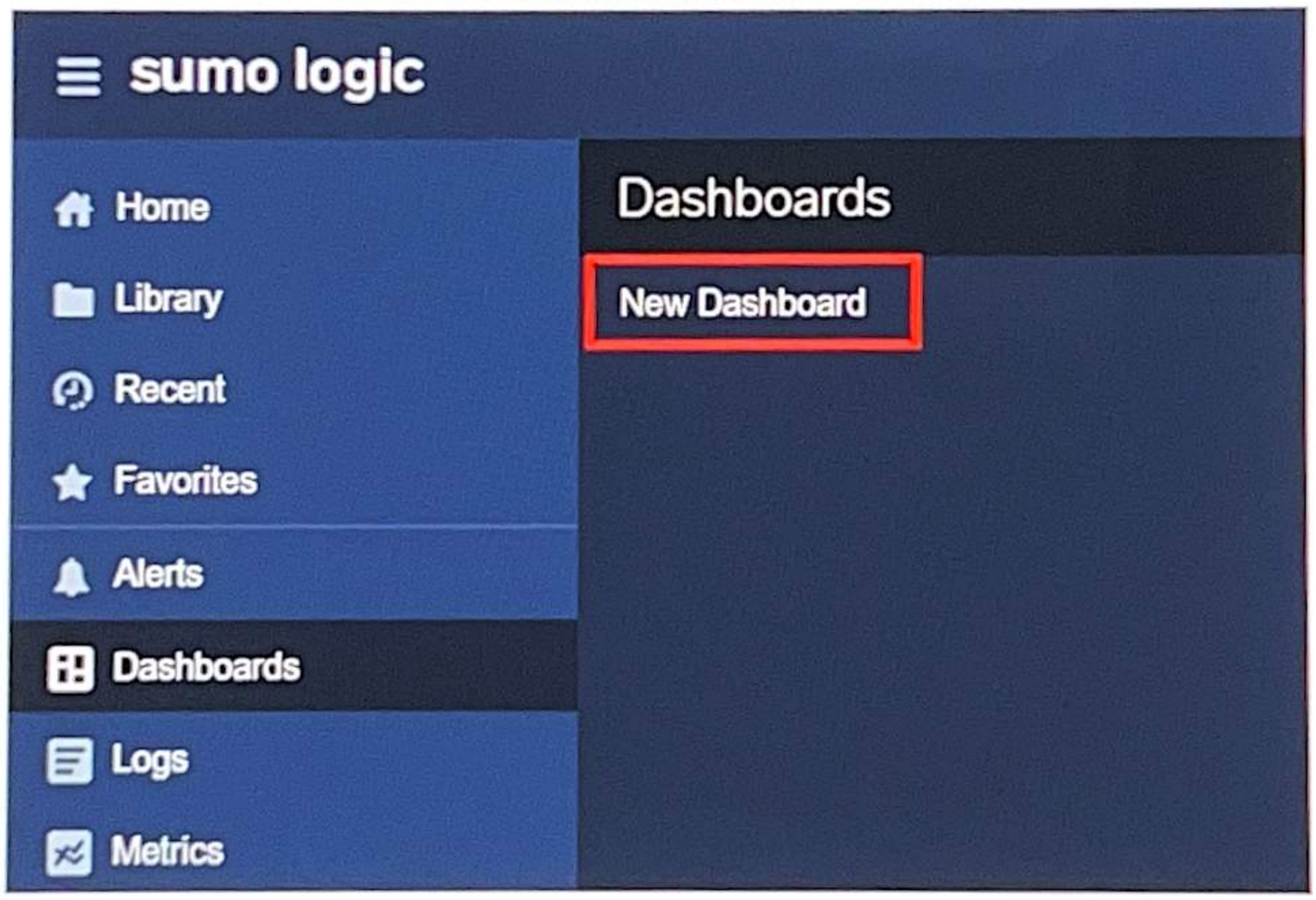
These labs are meant to be done in our Training environment using curated sample data. However, you are welcome to use your own environment by editing the query samples to fit your data and metadata.

1) Using Google Chrome or Firefox browsers, go to **https://service.sumologic.com**
2) Select a three-digit number from 001- 999
3) For the email enter **training+analyst###@sumologic.com** where **###** is the number you selected. For example, if you select 123 then your email would be training+analyst123@sumologic.com
4) For the password - check the landing page of the learning portal or email us at: training@sumologic.com

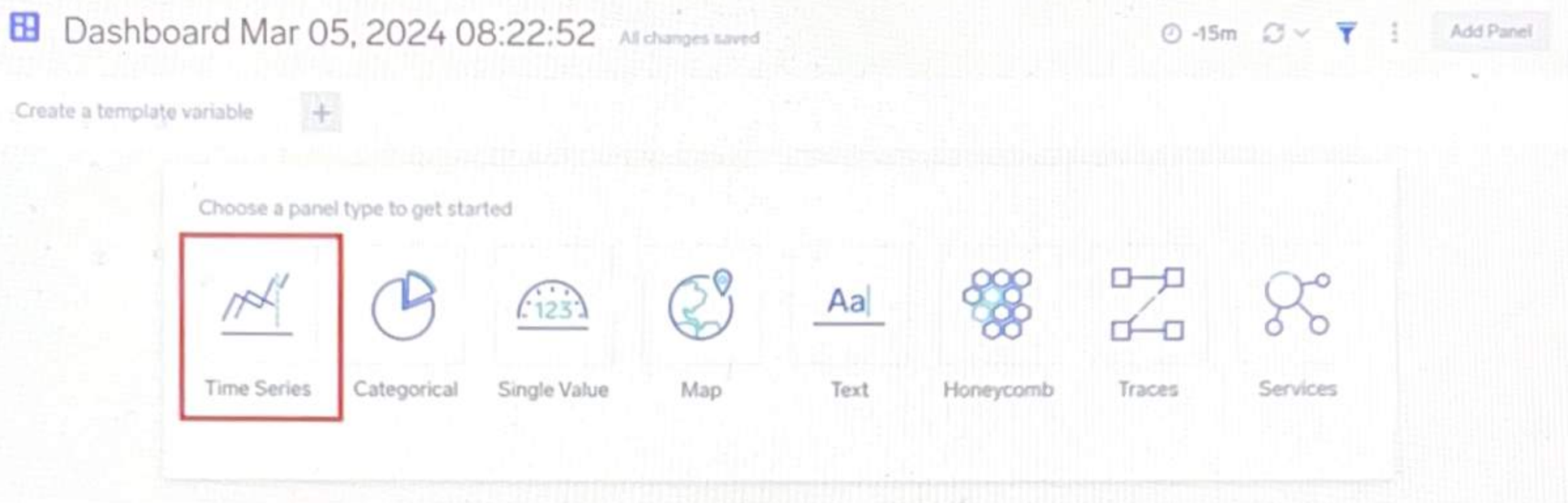
Lab Activity 1 - Create Top 10 User Activity Panel

Through these lab exercises, we are going to create a dashboard to look at our security activity in several different ways. In this step you will create a Top 10 User Activity query, turn it into a bar chart, and add it to your dashboard.

1. From the left tab menu in the Sumo Logic interface, select **Dashboards > New Dashboard** as shown below:



2. Select a **Time Series** panel as shown below:



3. A CloudTrail log is a record in JSON format. The log contains information about requests for resources in your account, such as who made the request, the services used, the actions performed, and parameters for the action. First, you can search AWS CloudTrail logs to extract the user, event, IP addresses, and event names as metadata using a parse json command, so you can use the extracted values to monitor user activity. Copy or type the following query to the query window:

```
_sourceCategory=Labs/AWS/CloudTrail
| json field=_raw "userIdentity.userName" as actor
| json field=_raw "eventType" as event_type
| json field=_raw "sourceIPAddress" as src_ip
| json field=_raw "eventName" as event_name
```

4. Note that if you try to run the query now, you'll receive an error saying you are missing an aggregate operator – an operator that groups your data together according to your desired criteria. Let's add some aggregate operators to the query. You can add additional lines to your query text at any time by typing **shift + enter**.

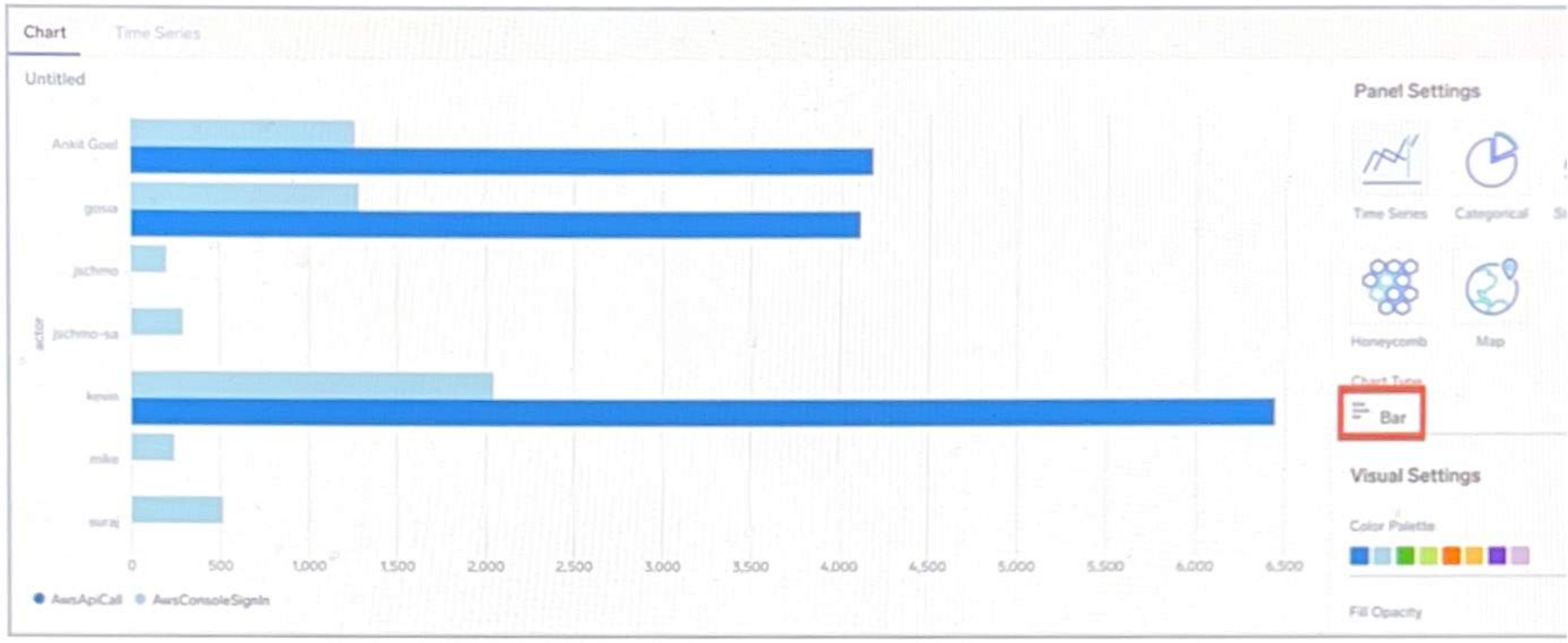
5. Now add the following code to the end of your query:

```
| count by actor,event_type,event_name
| top 10 actor by event_type,_count
| transpose row actor column event_type
```

6. Change the timeframe for the query to be the last 24 hours:

7. Now your query is complete: run it by hitting **Enter** or clicking on the "magnifying glass" icon on the right side.

8. In the Panel Settings area change the chart type to Bar:



9. In the upper left corner of your bar chart - change the panel title to **Top 10 User Activity**

You can also click on the General tab on the far right of the panel controls (with the gear symbols) and add the panel title there.

10. Click the **Add to Dashboard** button in the top right corner.

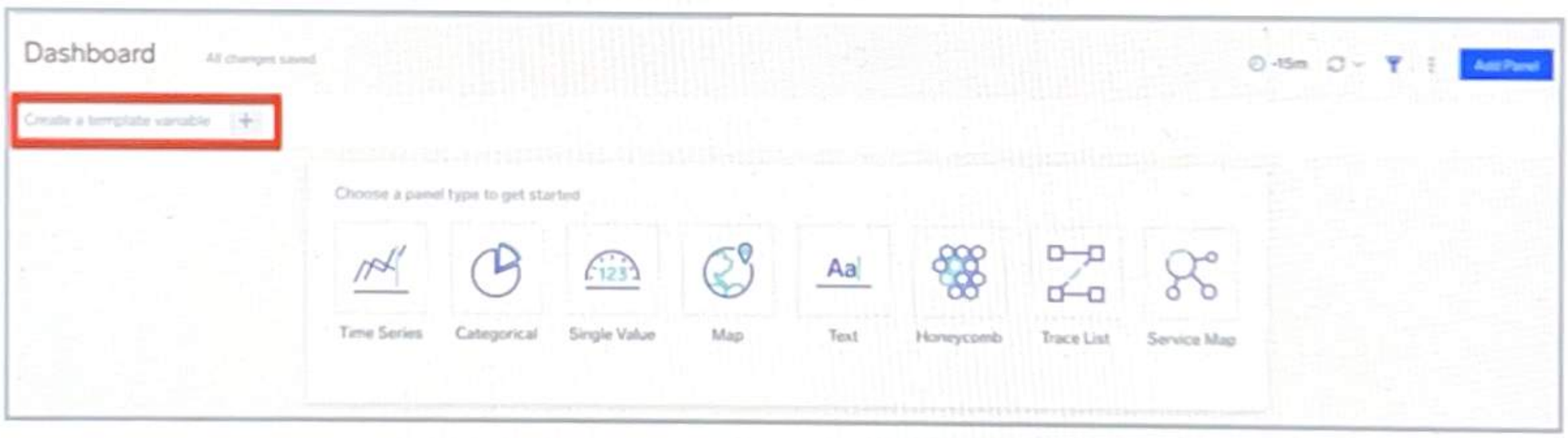
CONGRATULATIONS!! You have made your first Dashboard panel. Now we will enhance it by adding Dashboard Template Variables and modifying the query to include these new parameters.

Lab Activity 2 - Create Dashboard Template Variables

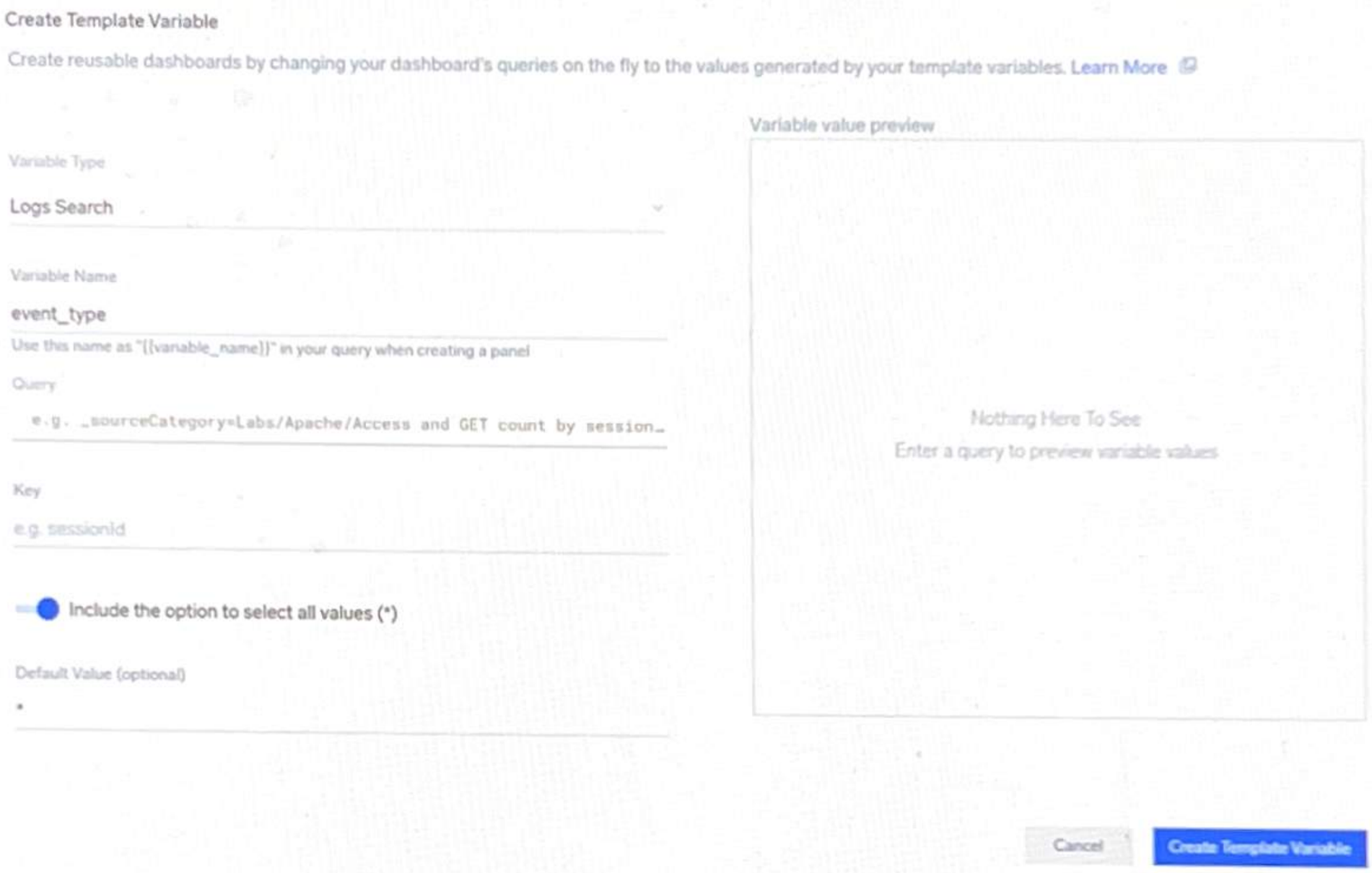
We can add more flexibility to our queries and dashboard outputs by using **template variables**. Let's create some to build out our dashboard.

1. In the upper right corner of the dashboard, change the time range to **24 hours**.
2. Click on the **filter** icon to display the template variable bar.

3. In the template variable bar, select **Create a Template Variable**.



4. In this panel for the Variable Type, select **Logs Search**, and for the Variable Name enter **event_type.**

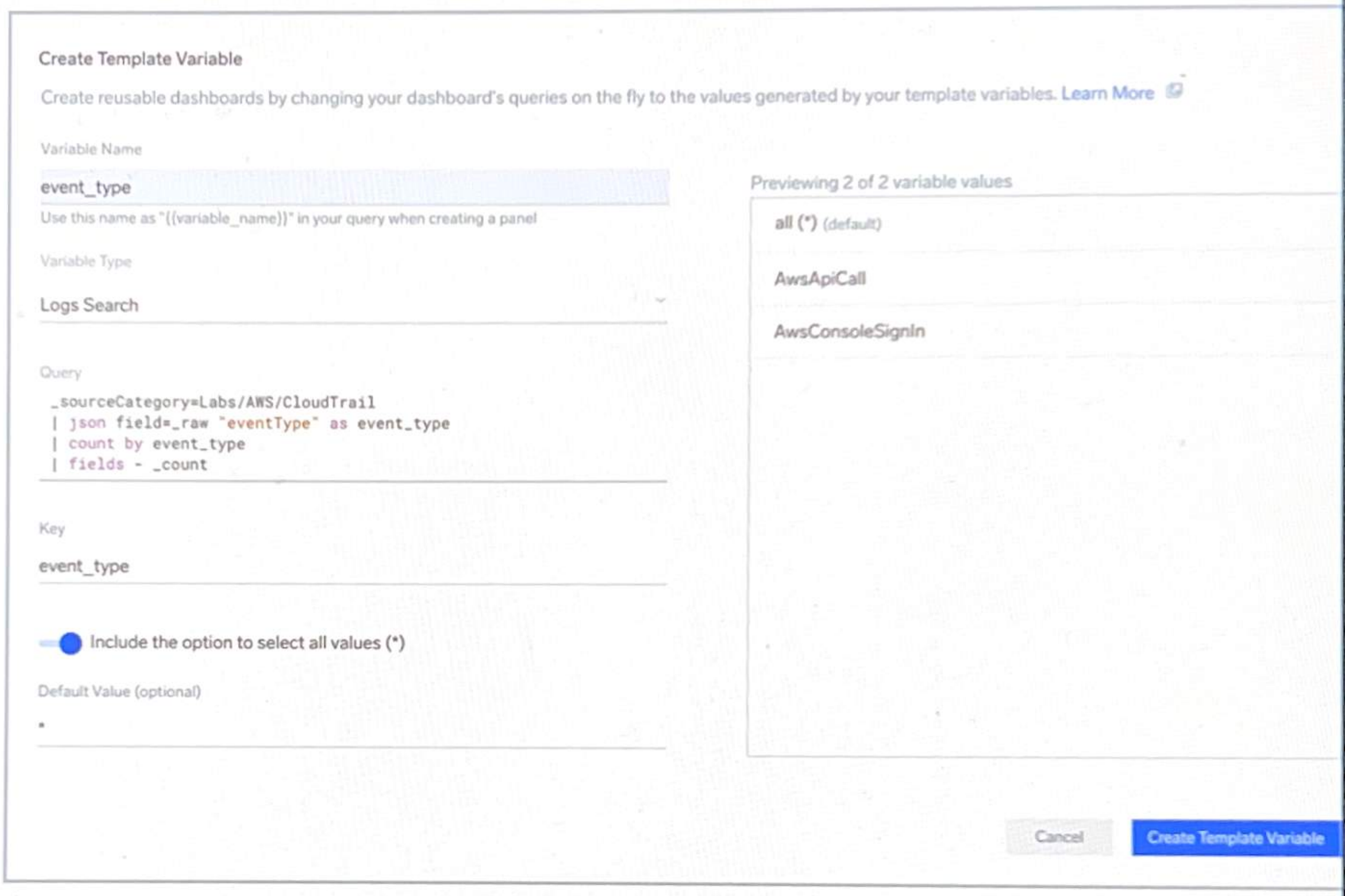


5. For the **Query** field, copy and paste the following:

```
_sourceCategory=Labs/AWS/CloudTrail
| json field=_raw "eventType" as event_type
| count by event_type
| fields - _count
```

6. Under the **Key** field, select "event_type" – you will notice that the right side of the panel will now populate showing you the values created by the query.

7. Make sure the "**Include the option to select all values (*)**" selector is active.



8. Click the **Create Template Variable** on the lower right button. A variable selector called "event_type" is now visible on the dashboard bar.

9. Let's create another variable called actor. Use this table to create the elements that are needed for this template variable.

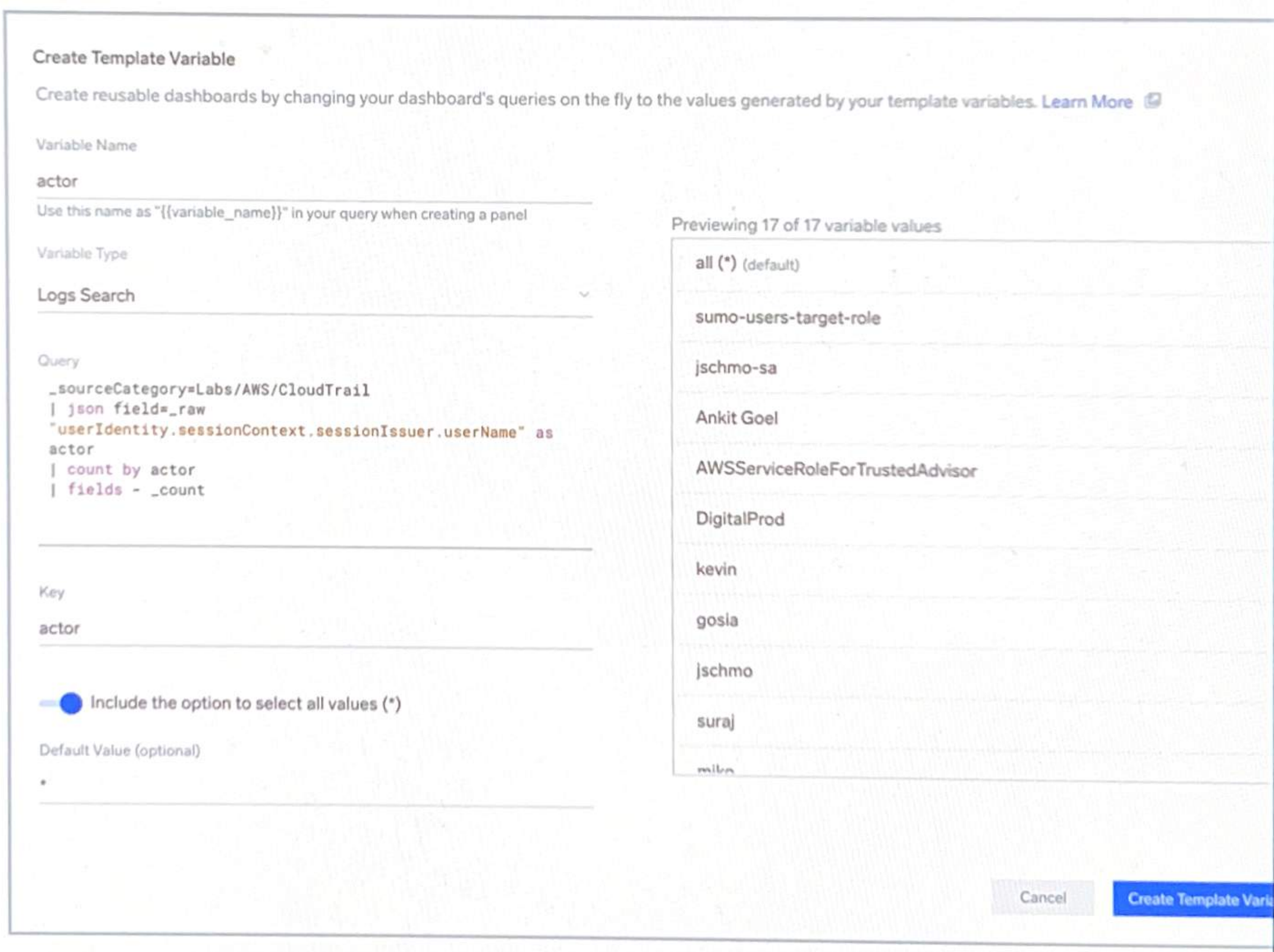
Variable Name	actor
Variable Type	Logs Search

Query	`_sourceCategory=Labs/AWS/CloudTrail`
	`| json field=_raw`
	`"userIdentity.sessionContext.sessionIssue`
	`r.userName" as actor`
	`| count by actor`

```
| fields - _count
```

Key	actor
Include the option to select all values (*)	Yes

Your results should look like:



10. Click the **Create Template Variable** button.

11. In the upper left of the dashboard, rename your dashboard to **Cloud Security Dashboard**

Lab Activity 3 - Modify your query to use these variables

We now want to modify our query to take advantage of the variables we have created. This will require us to add a where clause and reference the parameter by its name.

1. Click on your dashboard panel and click on the three vertical dots in the upper right corner of the **Top 10 User Activity** panel
2. Select the **Edit** option so you can modify your query.
3. We want to filter by event_type in our panel, but allow the dashboard viewer options to pick the event type. Add a new line to the query by hitting **Shift + Enter** and type or copy the following code:

```
| where event_type matches "{{event_type}}"
```

4. Now add this line after the line you just added

```
| where actor matches "{{actor}}"
```

5. Your query should now look like this:

```
_sourceCategory=Labs/AWS/CloudTrail
| json field=_raw "userIdentity.userName" as actor
| json field=_raw "eventType" as event_type
| json field=_raw "sourceIPAddress" as src_ip
| json field=_raw "eventName" as event_name
| where event_type matches "{{event_type}}"
| where actor matches "{{actor}}"
| count by actor,event_type,event_name
| top 10 actor by event_type,_count
| transpose row actor column event_type
```

6. Now click the **Update Dashboard** button

You can test these changes by selecting different event_types and actors from the template fields in the dashboard bar. As you select different values you will see the dashboard panel change automatically. Select the * option for a variable to see all options. When you are done testing, ensure both event_type and actor template variables are set to * before continuing.

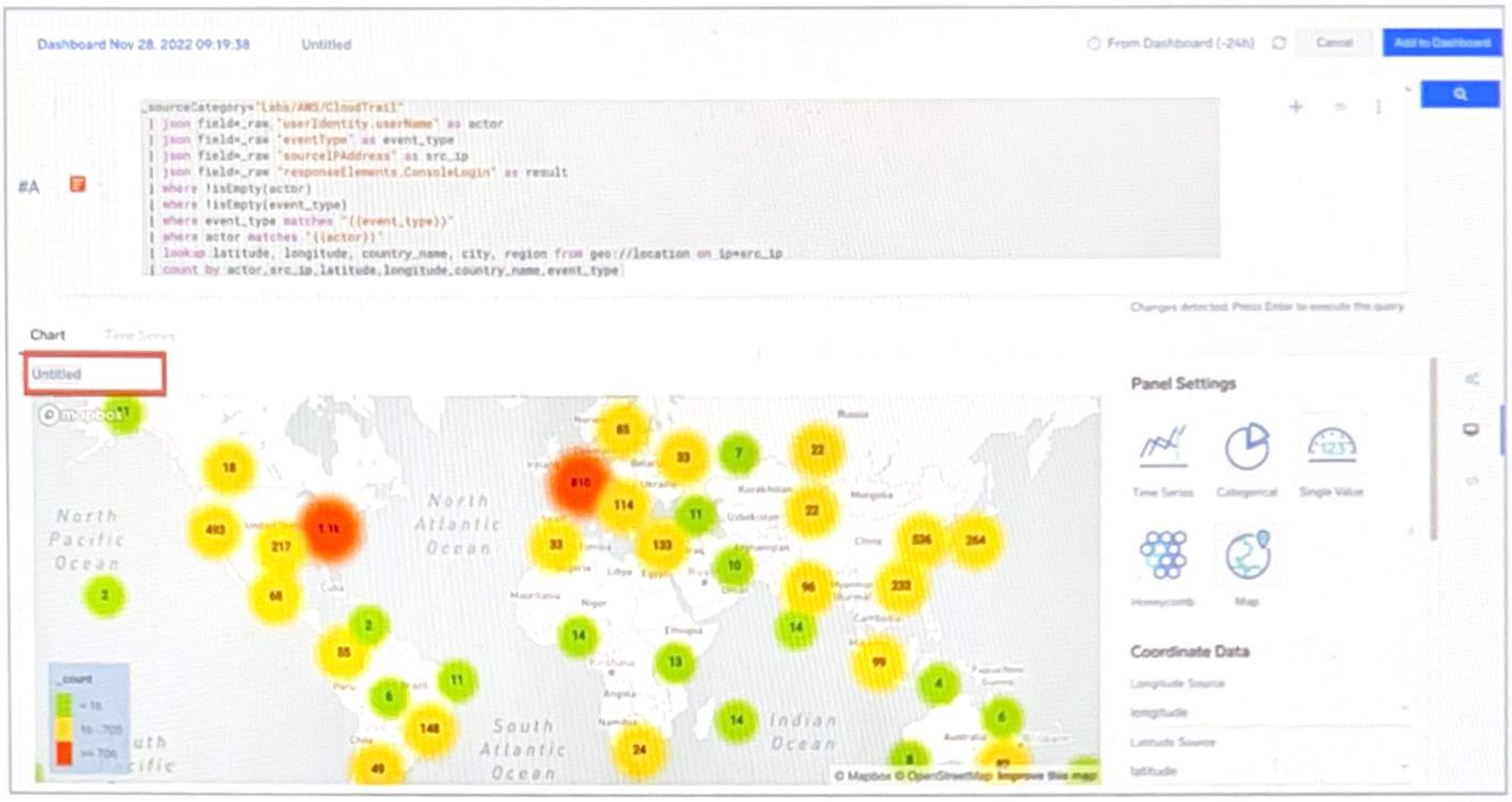
Lab Activity 4 - Geo Location of Console Logins

We want to see where our users are logging in from around the globe. So we are going to create a query to get the IP address and use the Lookup function to get the latitude and longitude of where that IP address is located.

1. Click **Add Panel** - select **Map**
2. Copy or type this code to the query window.

```
_sourceCategory="Labs/AWS/CloudTrail"
 | json field=_raw "userIdentity.userName" as actor
 | json field=_raw "eventType" as event_type
 | json field=_raw "sourceIPAddress" as src_ip
 | json field=_raw "responseElements.ConsoleLogin" as
result
 | where !isEmpty(actor)
 | where !isEmpty(event_type)
 | where event_type matches "{{event_type}}"
 | where actor matches "{{actor}}"
 | lookup latitude, longitude, country_name, city,
region from geo://location on ip=src_ip
 | count by
actor,src_ip,latitude,longitude,country_name,event_type
```

3. Name the panel **Geo Location of Console Logins**



4. Click the **Add to Dashboard** button.

NOTE: Change the values of the Dashboard Template Variables and see the effect on your dashboard panels

Lab Activity 5 - Top 10 number of failed login attempts

Now we want to see if users are failing to login, which might be an indication of users forgetting their passwords or someone else trying to login using stolen users' credentials

1. Click **Add Panel** - select **Time Series**
2. Add this code, your instructor will break down this code for you.

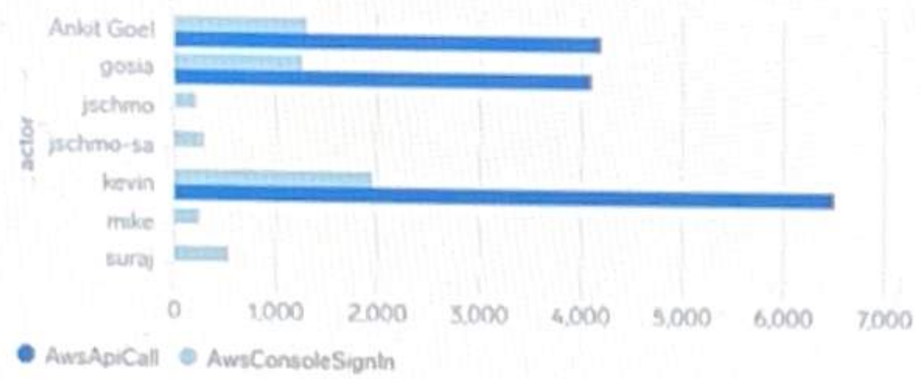
```
 _sourceCategory=Labs/AWS/CloudTrail
| json field=_raw "userIdentity.userName" as actor
| json field=_raw "eventType" as event_type
| json field=_raw "sourceIPAddress" as src_ip
| json field=_raw "responseElements.ConsoleLogin" as
result
| json field=_raw "eventName" as event_name
| where actor matches "{{actor}}"
| where !isEmpty(actor)
| where !isEmpty(event_type)
| where result = "Failure"
| where event_type = "AwsConsoleSignIn"
| timeslice 1h
| count by _timeslice,actor,event_type,event_name,result
| top 10 actor by _timeslice,event_type,result,_count
```

3. Under chart type, select **Table**
4. Rename this panel **Top 10 Number of Failed Login Attempts**
5. Click the **Add to Dashboard** button
6. Your dashboard will now look something like this:

event_type * actor * +

Top 10 User Activity

-24h



Geo Location of Console Logins

Top 10 Number of Failed Login Attempts

	actor	Time	event_type
1	gosia	02/28/2023 2:00:00 PM -0700	AwsConsoleSignIn
2	suraj	02/28/2023 2:00:00 PM -0700	AwsConsoleSignIn
3	kevin	02/28/2023 1:00:00 PM -0700	AwsConsoleSignIn
4	gosia	02/28/2023 1:00:00 PM -0700	AwsConsoleSignIn
5	kevin	02/28/2023 12:00:00 PM -0700	AwsConsoleSignIn
6	gosia	02/28/2023 12:00:00 PM -0700	AwsConsoleSignIn

Lab Activity 6 - Account compromise detection from a brute force attack

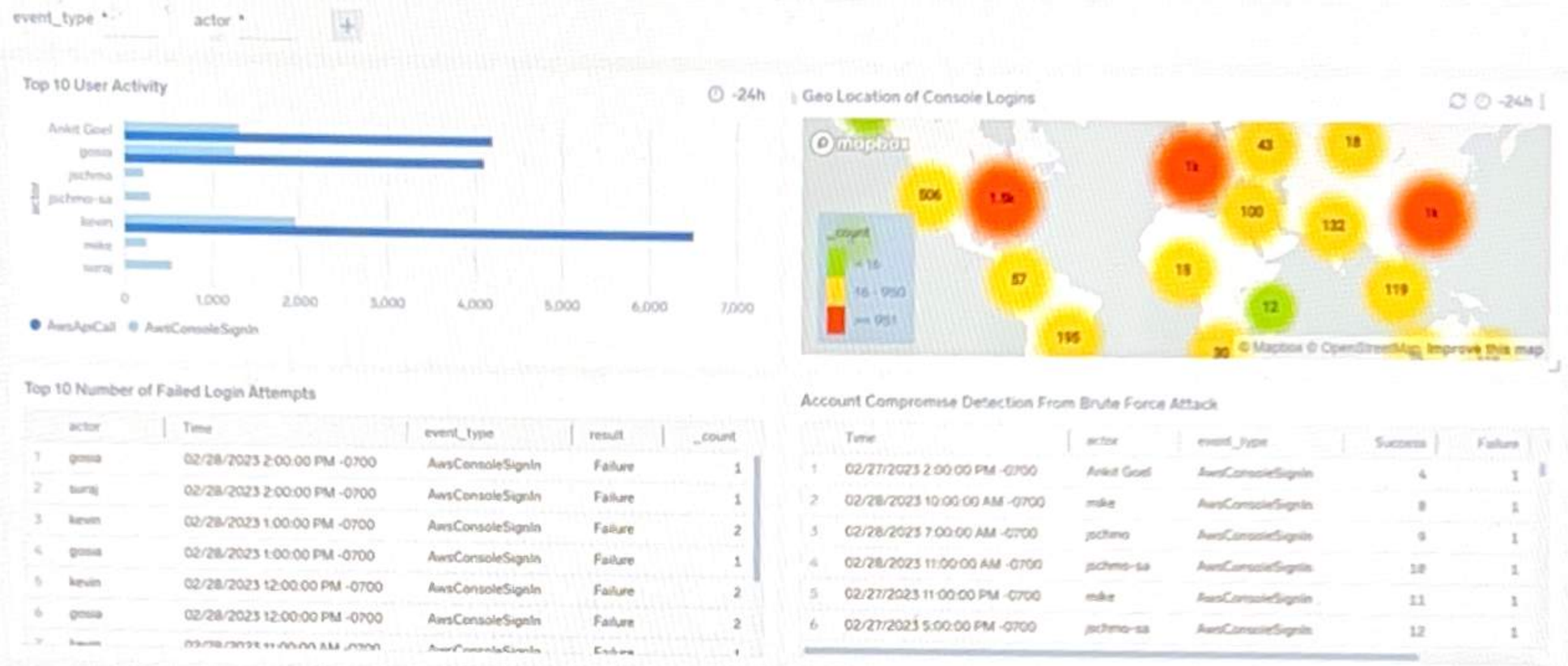
Brute force attacks are generally noted by a user failing to login a number of times, and then logging in successfully. Let's write a query to look for this and display the results.

1. Click **Add Panel** - select **Time Series**
2. Type or paste the following code into the query window:

```
_sourceCategory=Labs/AWS/CloudTrail
| json field=_raw "userIdentity.userName" as actor
| json field=_raw "eventType" as event_type
| json field=_raw "sourceIPAddress" as src_ip
| json field=_raw "responseElements.ConsoleLogin" as
result
| json field=_raw "eventName" as event_name
| where event_type matches "{{event_type}}"
| where actor matches "{{actor}}"
| timeslice 1h
| if(result = "Success",1,0) as success_count
| if(result = "Failure",1,0) as failure_count
|sum(success_count) as Success, sum(failure_count) as
Failure by _timeslice,actor,event_type
| where Failure>0
| Failure/Success*100 as Failure_percent
| order by Failure_percent desc
| format("%.2f", Failure_percent) as Failure_percent
```

3. Under chart type, select **Table**

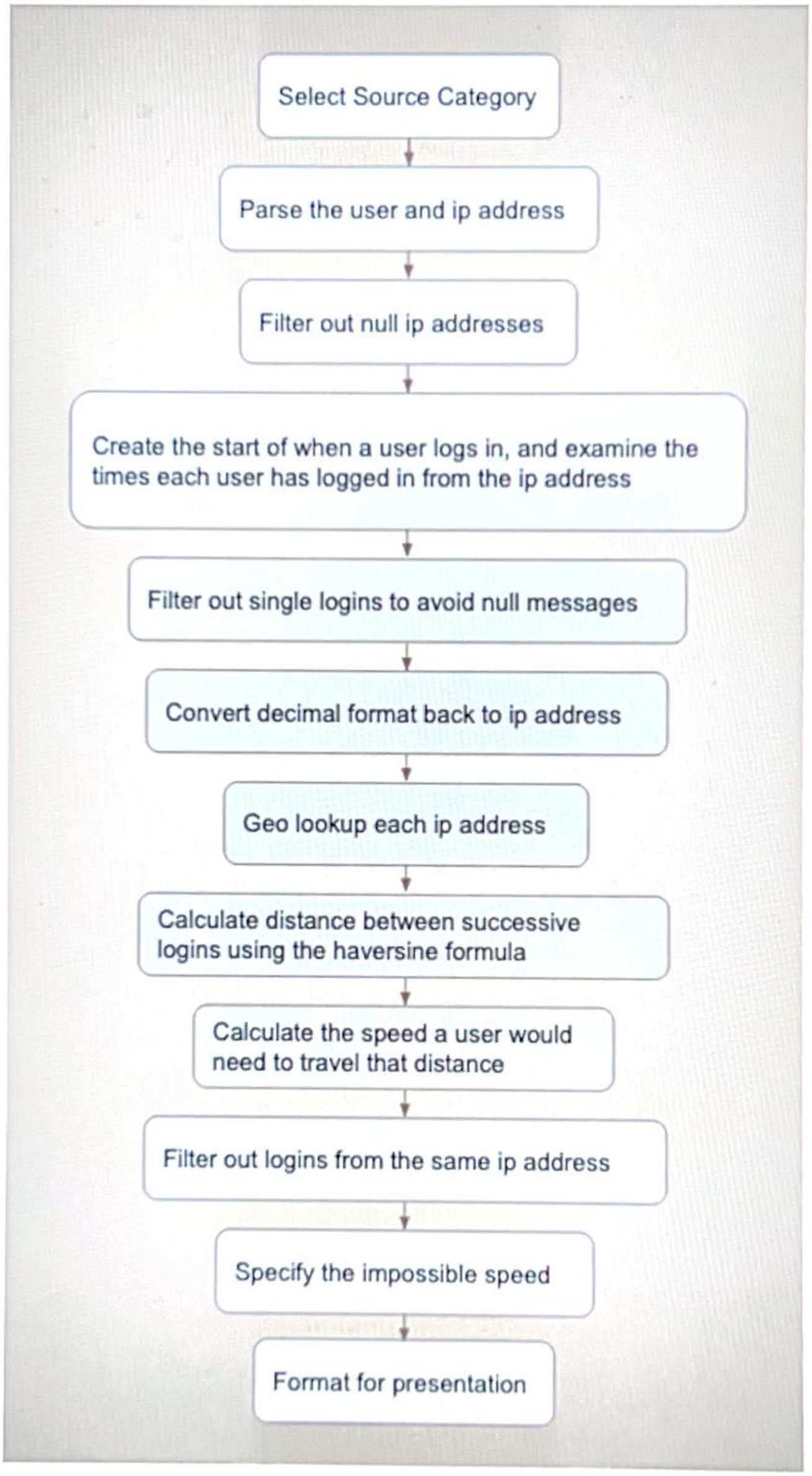
4. Rename this panel **Account Compromise Detection From Brute Force Attack**
5. Click the **Add to Dashboard** button. Your dashboard will now look something like this:



Lab Activity 7 - Detecting a Landspeed Violation

A "landspeed violation" occurs when a user logins from an IP address and then logs in a short time later from a different IP address where the location is a significant distance from the first location. For example - if someone logged in from New York City USA and then 4 hours later logged in using an IP address from Amsterdam in the Netherlands. It's impossible to get from New York City to Amsterdam in 4 hours, thus is an example of a landspeed violation that could indicate stolen or compromised credentials.

1. Click **Add Panel** - select **Time Series**
2. Because this query is fairly large we'll add it in sections so that we can explain what each section is doing. (Don't execute the query until the end when we've completed the entire query text.) The following flowchart represents the basic steps in summary:



3. In this first section, we are looking at the logs from our AWS Cloudtrail environment, parsing out the IP addresses and usernames. We will tie the username to our "actor" parameter and then filter for