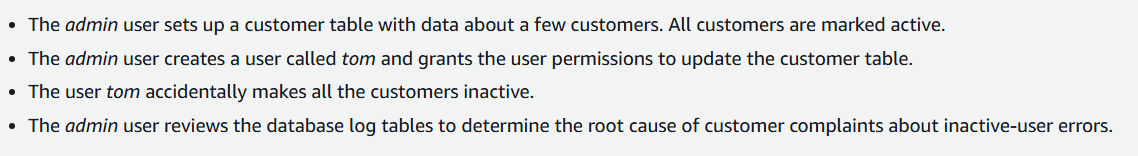
# Lab 8 - Auditing and Monitoring Amazon Redshift Clusters

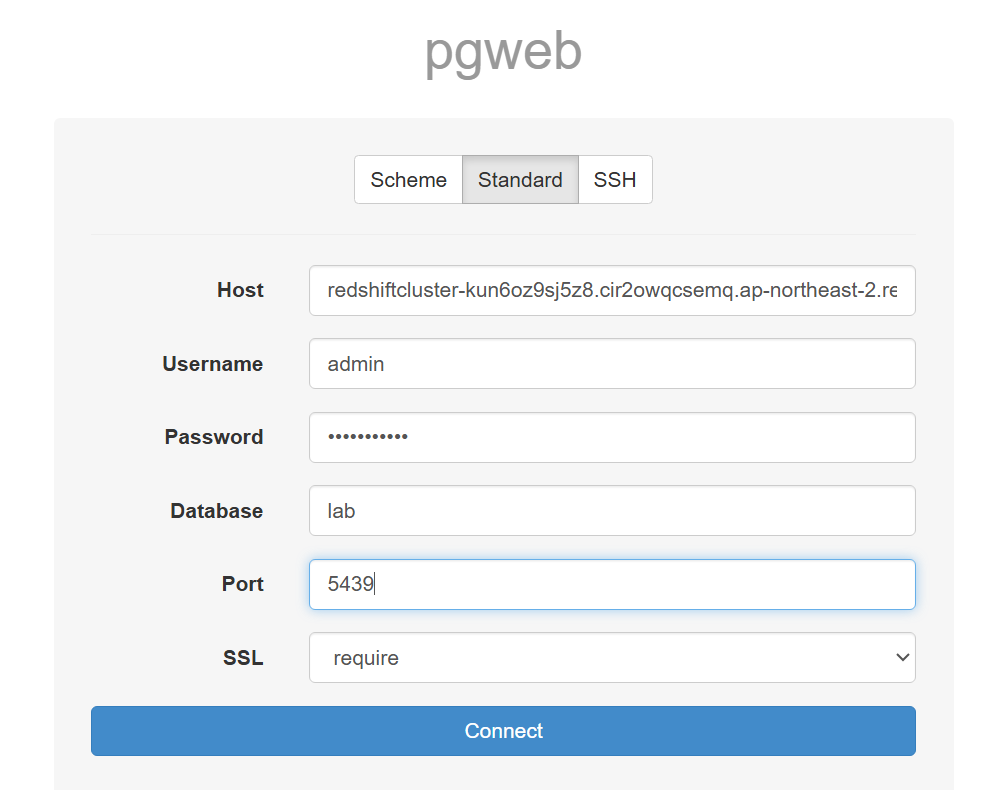
1. Use database audit logs.
2. Configure Amazon CloudWatch metrics to monitor performance.
3. Configure Amazon Simple Notification Service (Amazon SNS) to send notifications for events that occur in Amazon Redshift.

**Scenario -**



**Task 1: Connecting to Amazon Redshift**

1.1 use pgweb credentials to login as admin



1.2 query to create a table called CUSTOMER and a user named tom and grant all permissions to this user:

*CREATE TABLE customer (*

*id INT PRIMARY KEY,*

*name VARCHAR(25),*

*active BOOLEAN*

*);*

*INSERT INTO customer (id, name, active) VALUES*

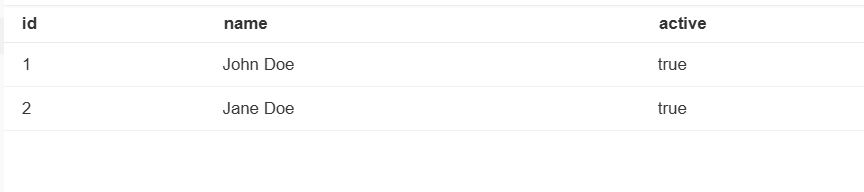
*(1, 'John Doe', TRUE),*

*(2, 'Jane Doe', TRUE);*

*CREATE USER tom WITH PASSWORD 'Redshift123' NOCREATEDB NOCREATEUSER;*

*GRANT ALL ON TABLE customer TO tom;*

1.3 table-



1.4 again connect to pgweb, this time as user - tom

Diff creds



1.5 run this to make users inactive

*UPDATE customer*

*SET active = FALSE;*

The Amazon Redshift cluster keeps an audit of which user ran each command. The UPDATE command you just ran will be associated with the user tom.

**Task 2: Checking Database Audit Logs**

2.1 use admin pgweb to run

*SELECT \**

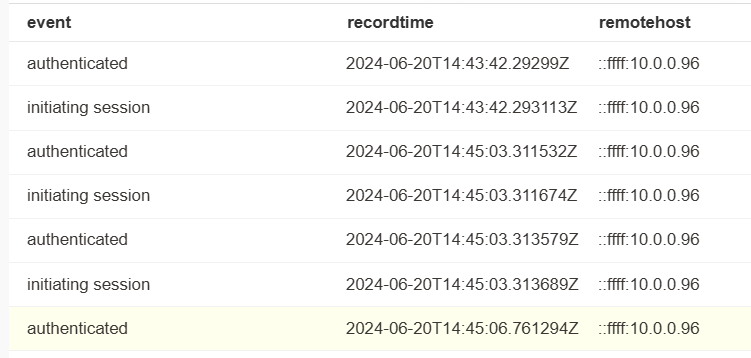
*FROM stl\_connection\_log*

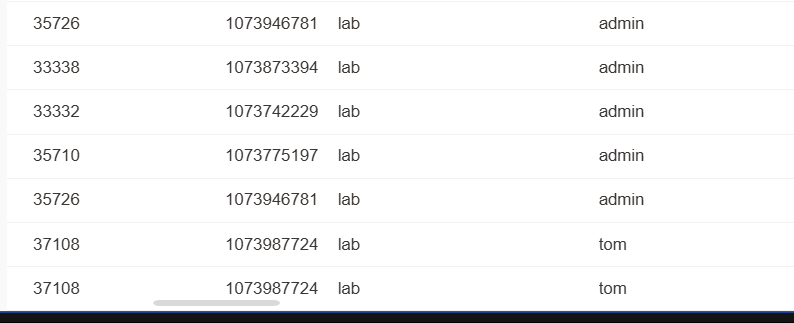
*WHERE dbname='lab'*

*AND username in ('admin', 'tom')*

*ORDER BY recordtime;*

Shows all things done



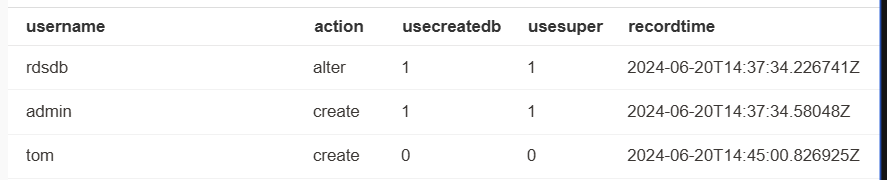


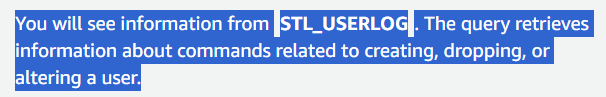
2.2 to view commands run

*SELECT username, action, usecreatedb, usesuper, recordtime*

*FROM stl\_userlog*

*ORDER BY recordtime;*





2.3 to view queries run

*SELECT q.userid, u.username, q.query, q.database, q.querytxt*

*FROM stl\_query q, stl\_userlog u*

*WHERE querytxt ILIKE '%UPDATE%'*

*AND q.userid = u.userid;*



**Task 3: Configuring Event Notifications and Alarms**

1. **\*\*Email Alerts:\*\***

- Create an Amazon SNS topic.

- Subscribe your email to the SNS topic.

2. **\*\*Storage Alarm:\*\***

- Set up a CloudWatch alarm for storage usage.

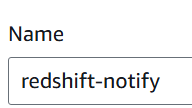
- Link the alarm to the SNS topic for notifications.

3. **\*\*Connection Threshold:\*\***

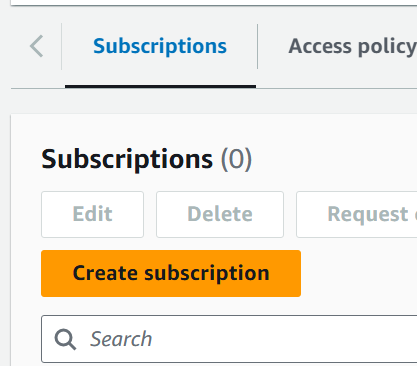
- Configure an alert for exceeding database connections.

- Use CloudWatch for similar metrics like CPU and network.

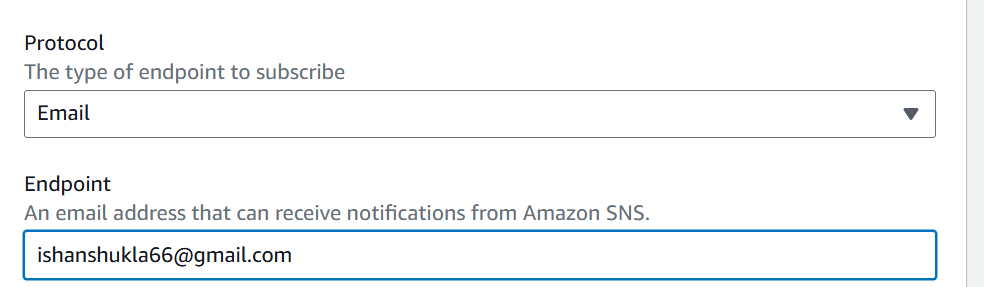
3.1 open SNS on console and create a new topic



3.2 create a sub



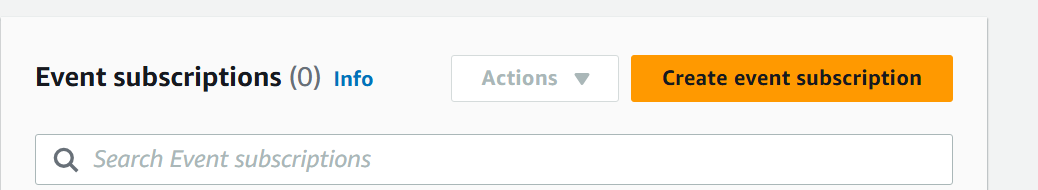
3.3 provide email deets



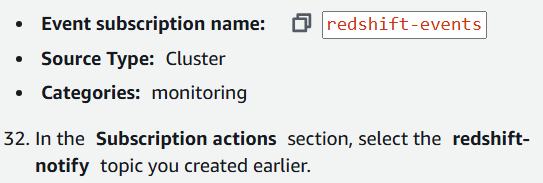
3.4 confirm on email

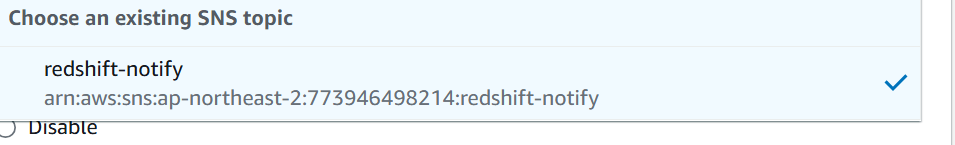
3.5 configure redshift to use SNS

3.6 go to events, then event subs

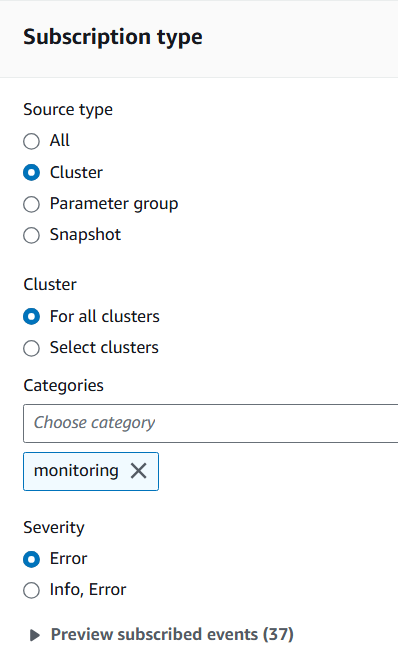


3.7 these values connect it to SNS

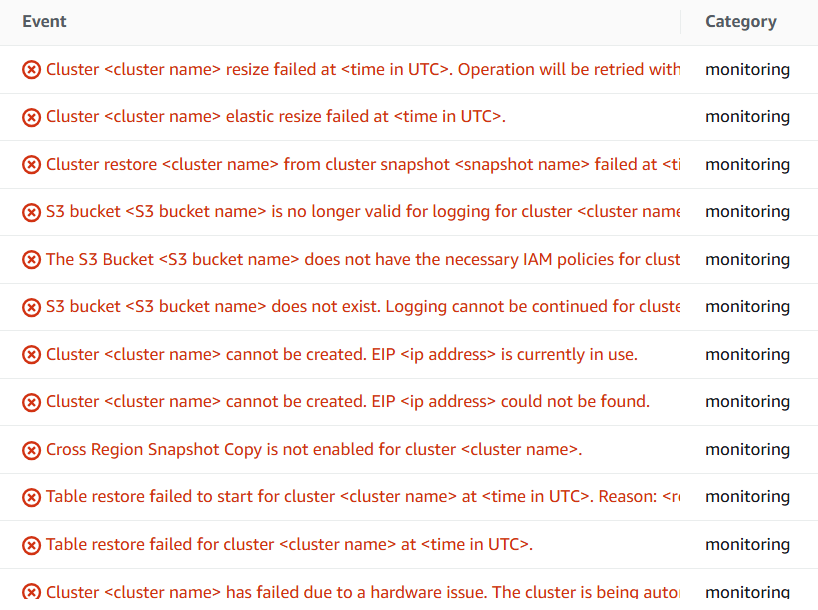




3.8 how?

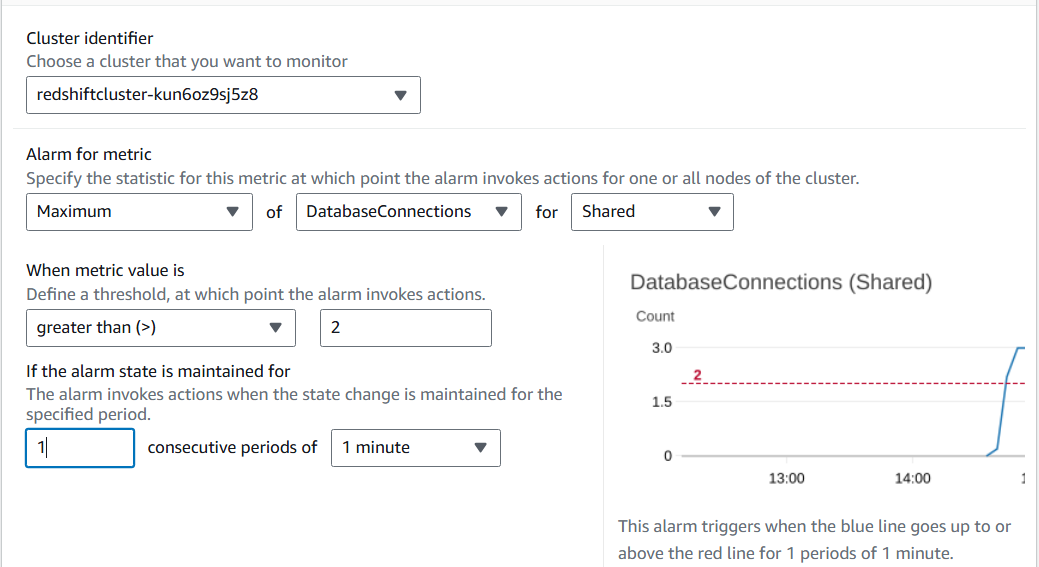


3.9 these error

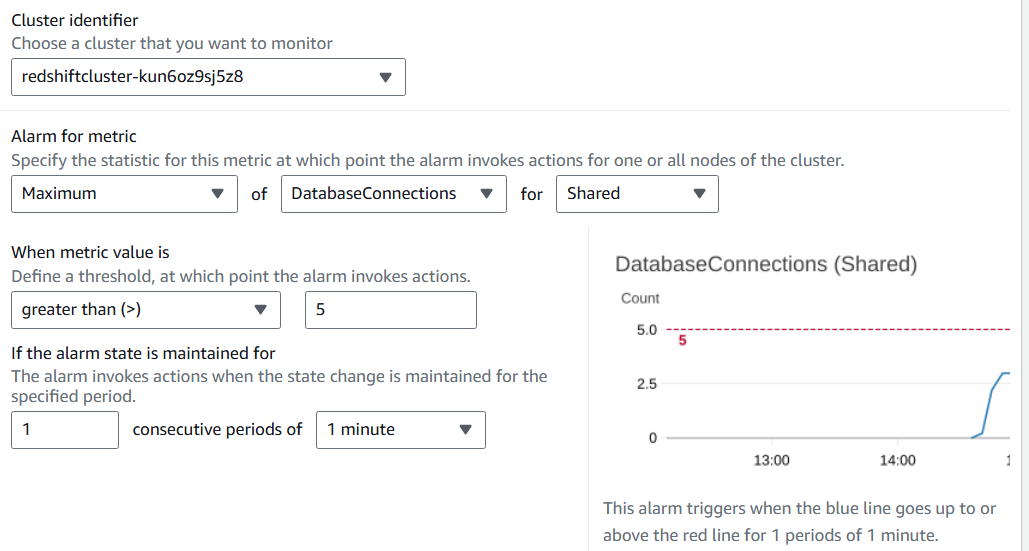


**Task 4 - CONFIGURING AN AMAZON CLOUDWATCH ALARM**

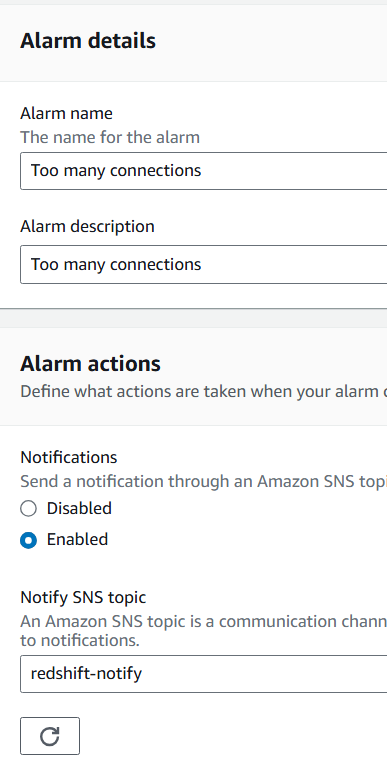
4.1 open alarms in RS, and create one using these metrics



4.2 edit 2 to 5 to ensure starting state is not alarming

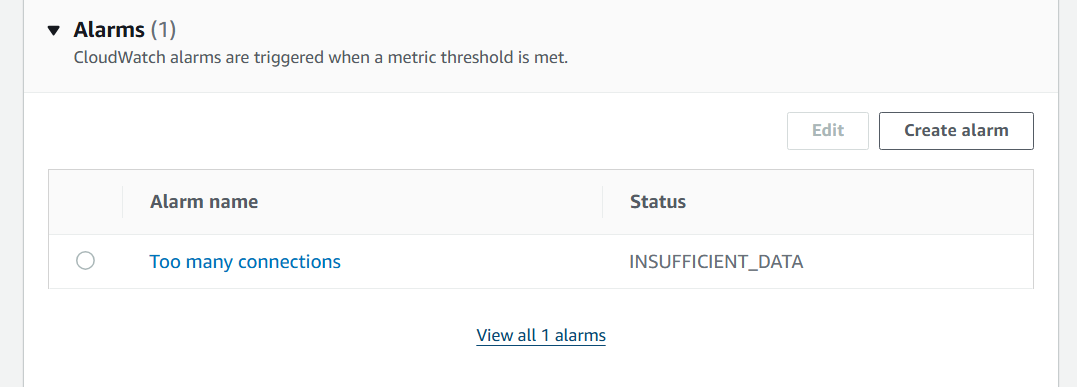


4.3 choose alarm name and connect to sns

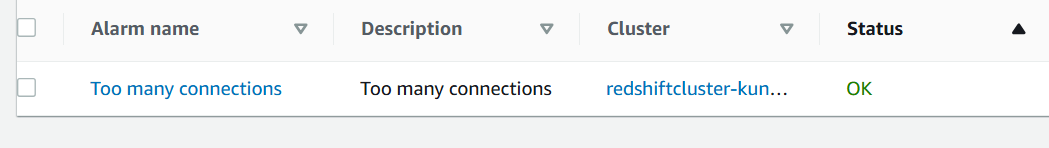


4.4 open you RS cluster

4.5 open the alarm



4.6 wait for 1 min for data to load

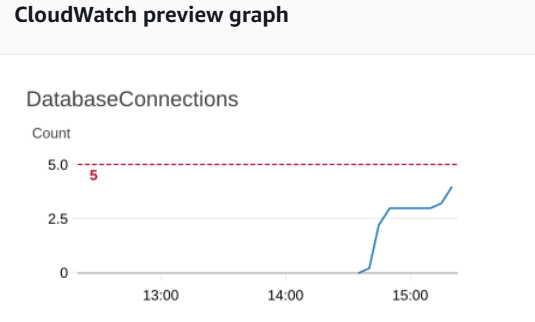


4.7 re-connect pgweb

4.8 re-connect total 5 times (4.2)



4.9 results -



Alarm received on email