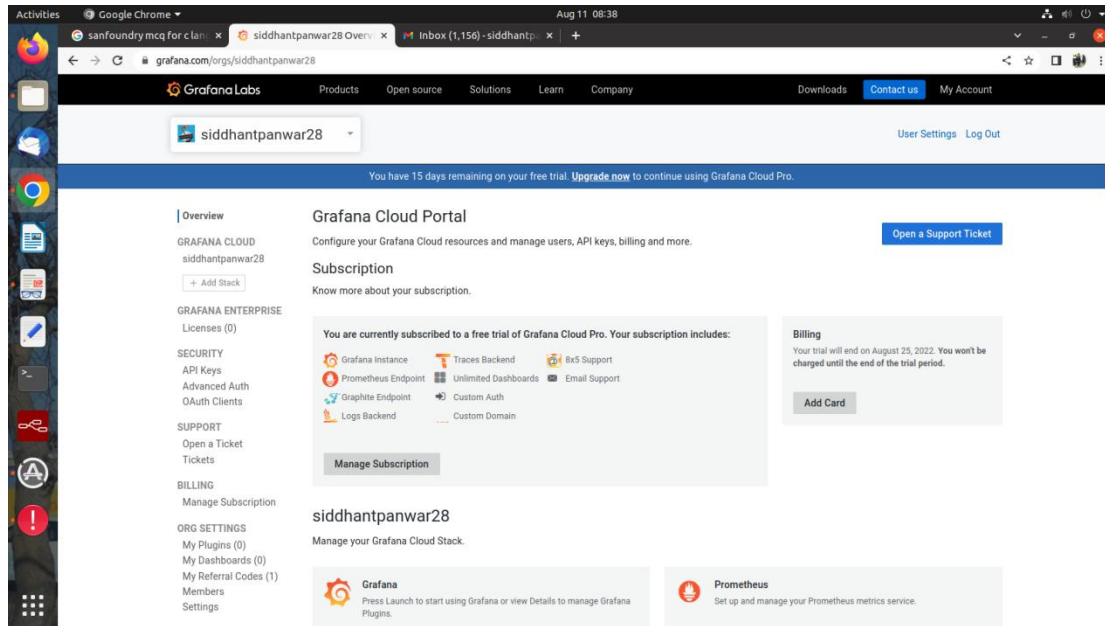
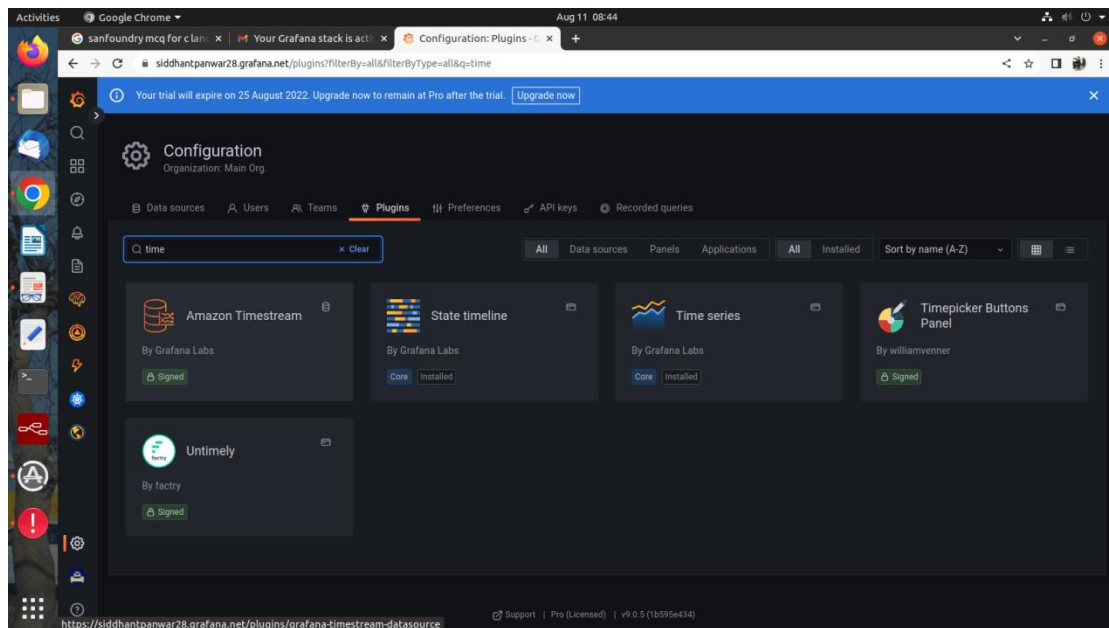


1) Make grafana Account and wait for the stack active e-mail from Grafana(2-3 mins wait)
-->Reopen tabs to refresh



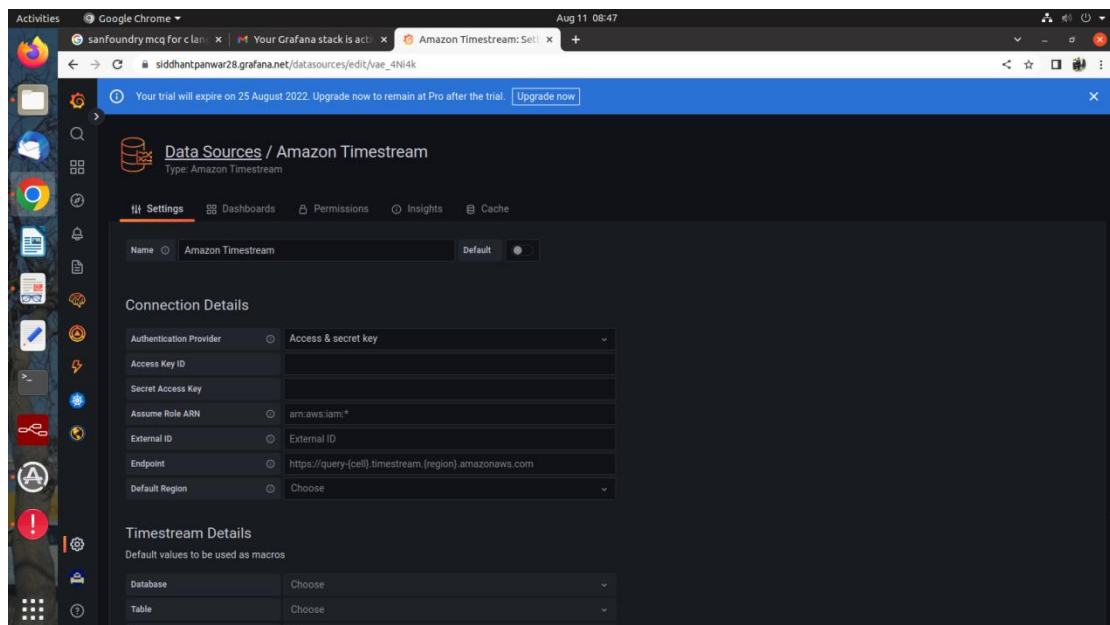
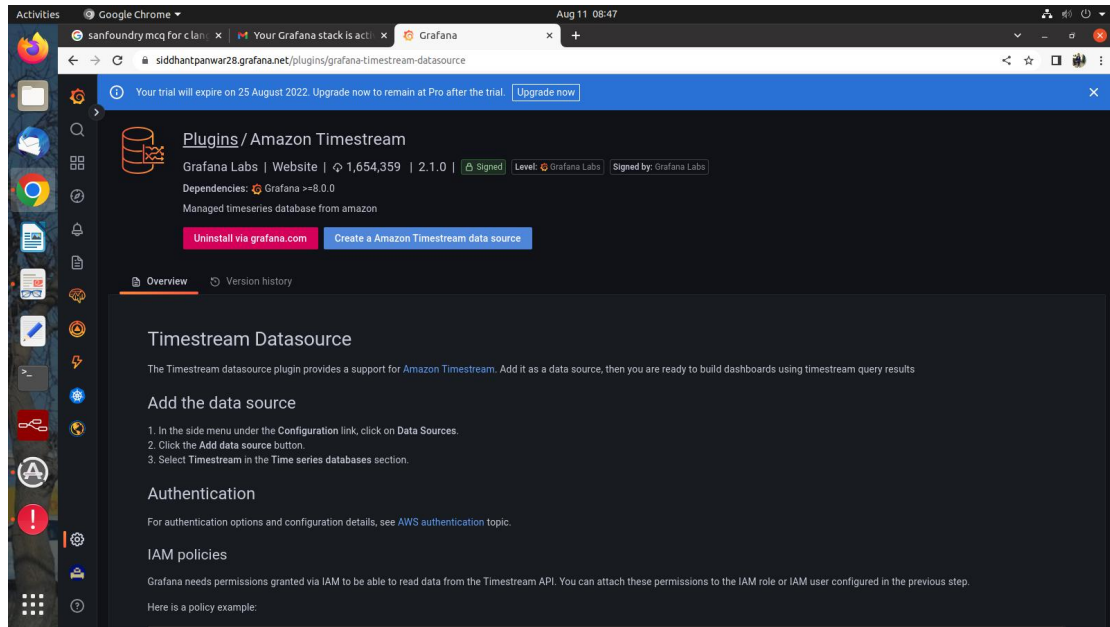
2)Goto -->Settings --> Configuration --> Plug-ins --> search 'Amazon Timestream' & then install it -->Reopen Grafana to refresh



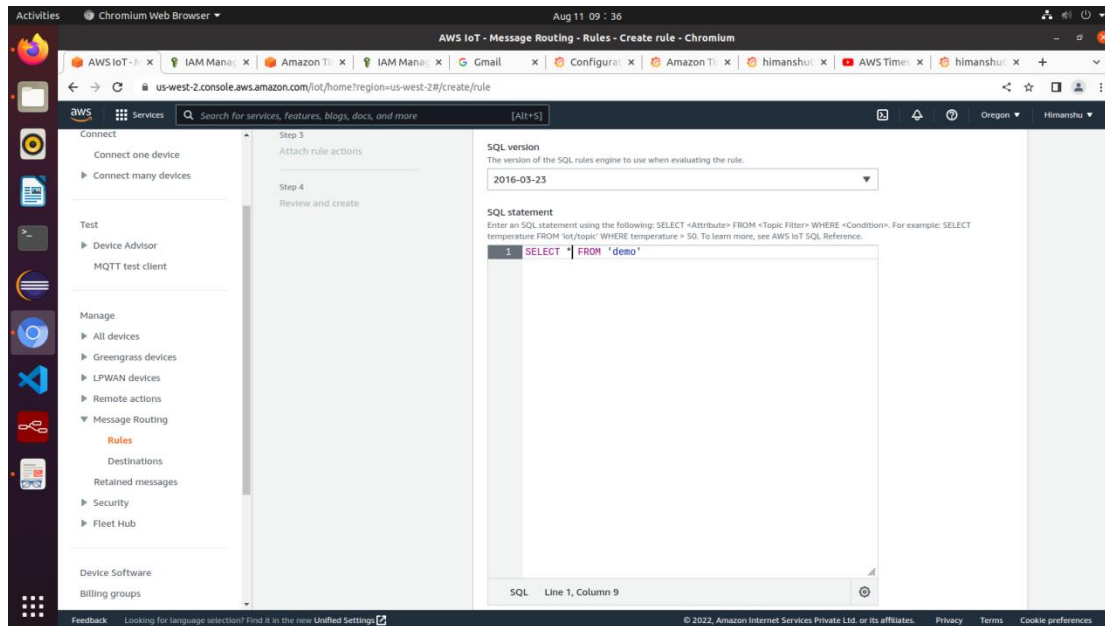
3)Goto -->Amazon Timestream again and then click 'Create a Amazon Timestream data source'

Now we are on Connection Details page.

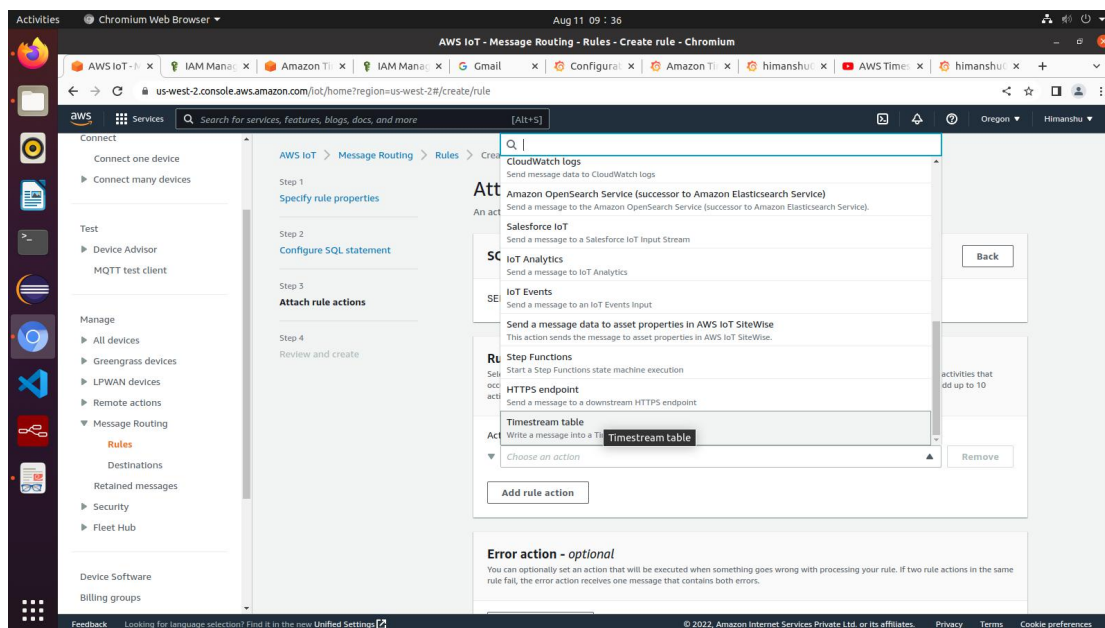
Need 'Access Key id' and 'Secret Access Key'

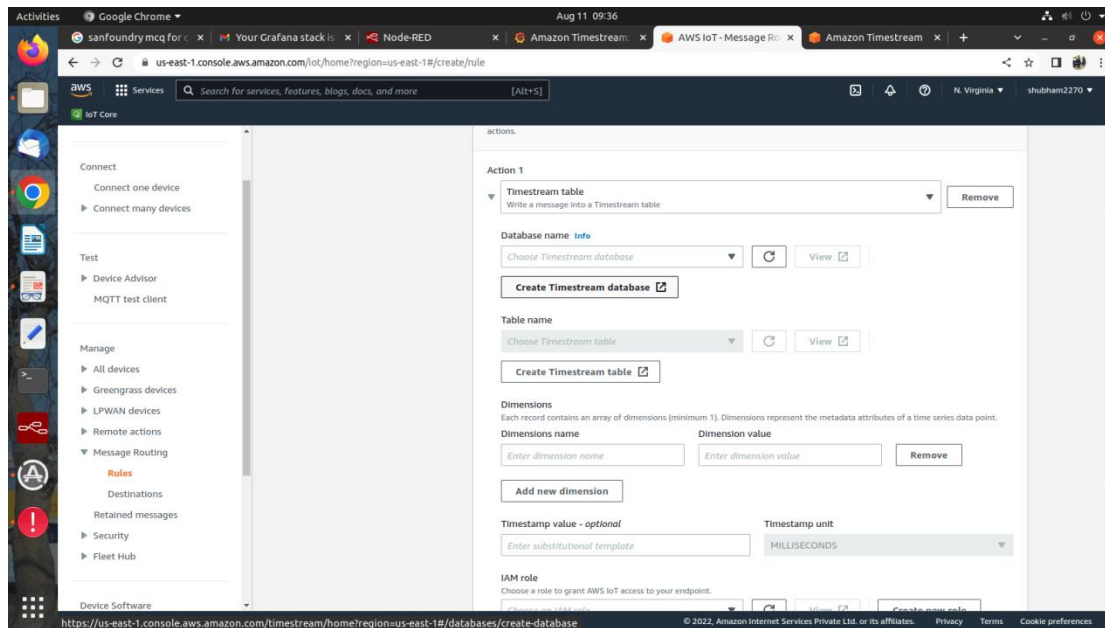


4) Now Goto --> AWS IoT --> Message Routing --> Rules --> Create Rule --> Name --> SQL Statement --> **SELECT * FROM 'demo'**



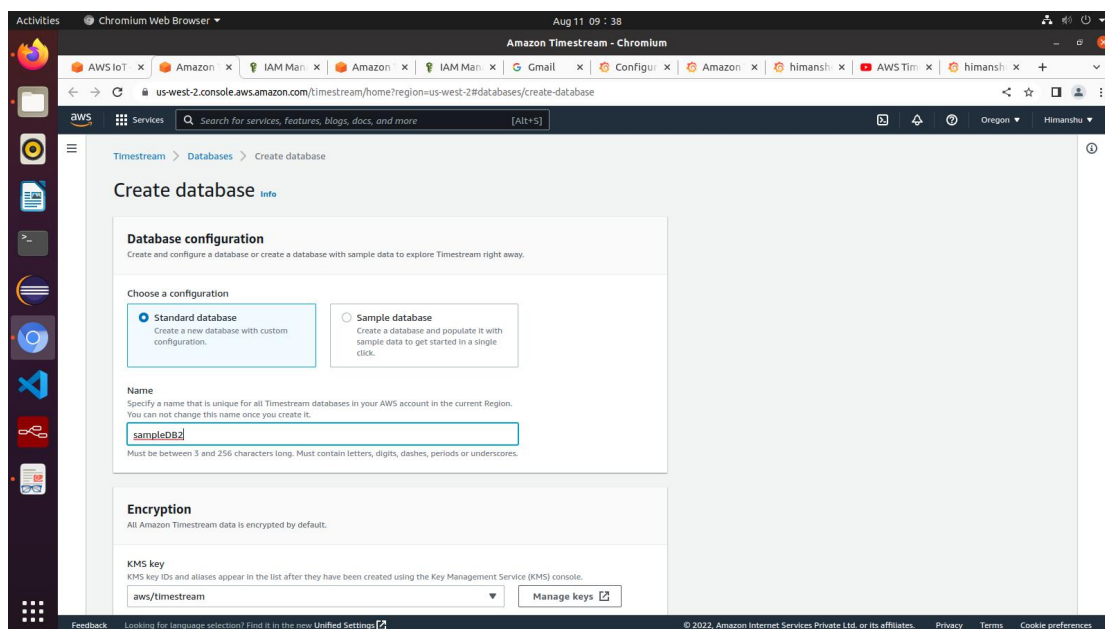
--> Rule Action --> Action1 --> Choose TimeStream Table.



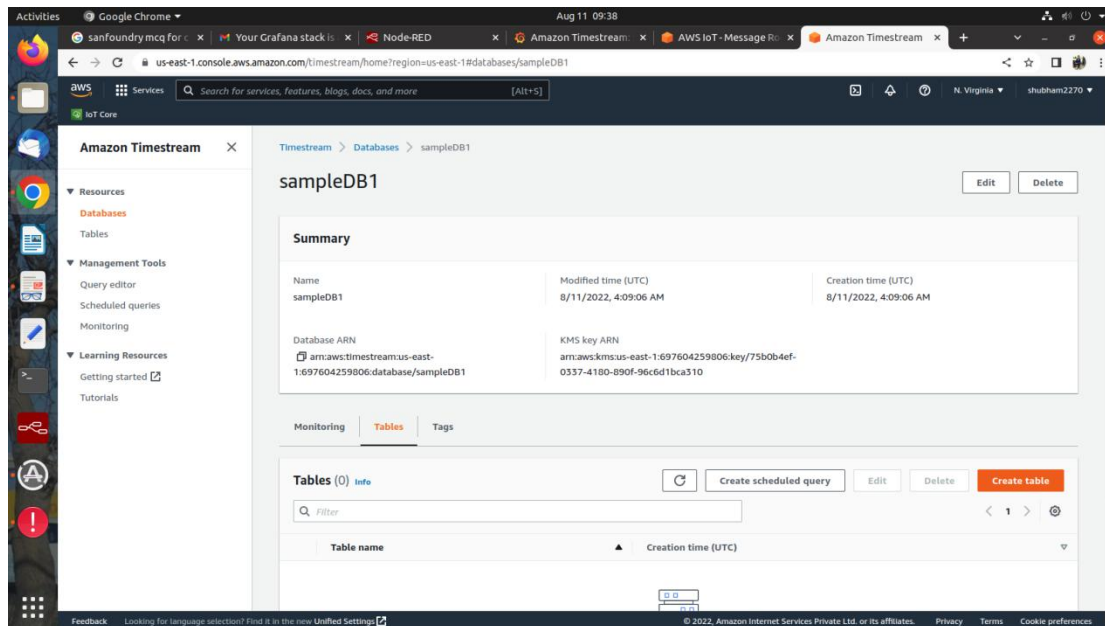


--> Click on Create Timestream Database --> Select Standard Database.

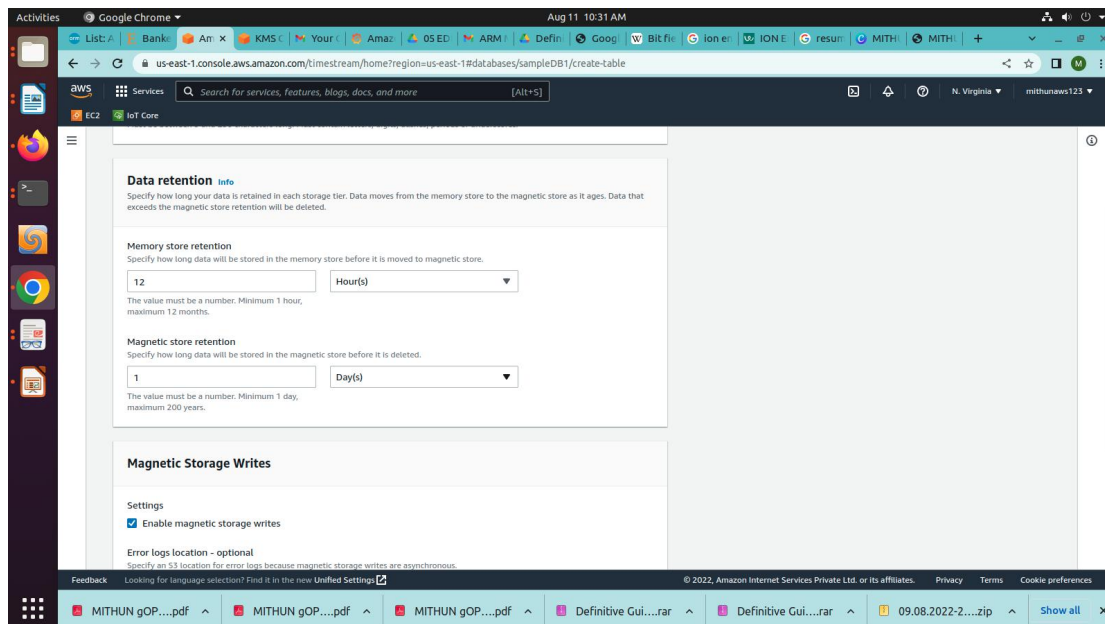
--> Give name(SampleDB) and then Click on Create.



--> Goto SampleDB tab --> Tables --> Create Table --> Give name (myTimestreamTable)



--> in Data Retention --> Select 12 Hours & 1 Day --> select Create Table.



5)Goto Previous tab (Rules) --> Refresh and Select Database name and Table name.

Step 4
Review and create

Rule actions
Select one or more actions to happen when the above rule is matched by an inbound message. Actions define additional activities that occur when messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. You can add up to 10 actions.

Action 1
Timestream table
Write a message into a Timestream table
Remove

Database name [info](#)
sampleDB1 [View](#)
[Create Timestream database](#)

Table name
MyTimestreamTable [View](#)
[Create Timestream table](#)

Dimensions
Each record contains an array of dimensions (minimum 1). Dimensions represent the metadata attributes of a time series data point.

Dimensions name
Enter dimension name

Dimension value
Enter dimension value
Remove

[Add new dimension](#)

Timestamp value - optional
Timestamp unit

--> in Dimensions --> Give Name(any) and Dimension value --> 'enode_id'(should be same as in NodeRed json msg)

Write a message into a Timestream table
Remove

Database name [info](#)
sampleDB1 [View](#)
[Create Timestream database](#)

Table name
MyTimestreamTable [View](#)
[Create Timestream table](#)

Dimensions
Each record contains an array of dimensions (minimum 1). Dimensions represent the metadata attributes of a time series data point.

Dimensions name
ID

Dimension value
enode_id
Remove

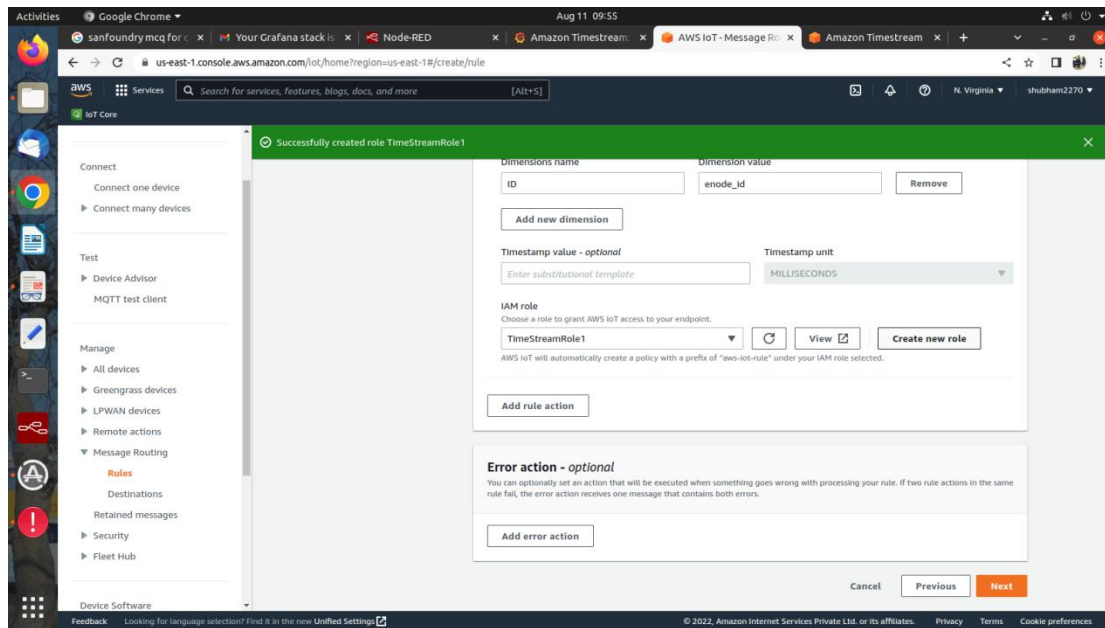
[Add new dimension](#)

Timestamp value - optional
Enter substitution template

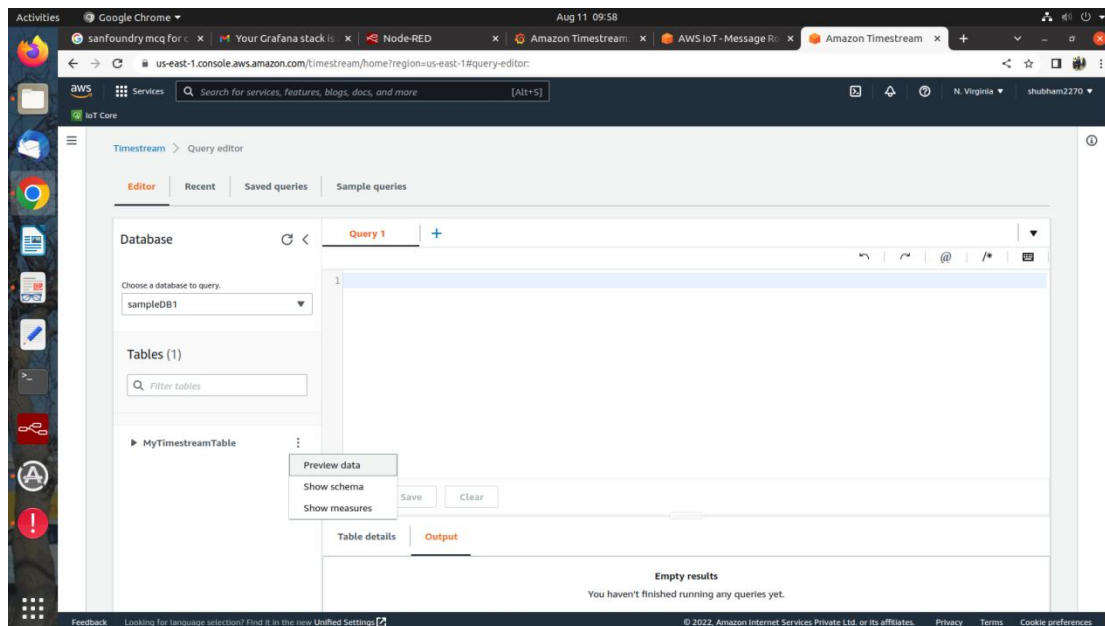
Timestamp unit
MILLISECONDS

IAM role
Choose a role to grant AWS IoT access to your endpoint.
Choose an IAM role [View](#) [Create new role](#)
AWS IoT will automatically create a policy with a prefix of "aws-iot-rule" under your IAM role selected.

[Add rule action](#)



**7)Goto TimeStream tab --> Query Editor -->(In tables)
On Timestream --> select Three dots --> Preview Data
--> New Query Will come**



Activities Chromium Web Browser Aug 11 10:02

Amazon TimeStream - Chromium

us-west-2.console.aws.amazon.com/timestream/home?region=us-west-2#query-editor:

Database

Choose a database to query:

sampleDB2

Tables (1)

Filter tables

mynewtimestreamtable

Query 1

```
1 -- Get the 10 most recently added data points in the past 15 minutes. You can change the time period if you're not continuously ingesting data
2 SELECT * FROM "sampleDB2"."mynewtimestreamtable" WHERE time between ago(15m) and now() ORDER BY time DESC LIMIT 10
```

Run Save Clear

Table details Query results Output

Rows returned (10)

Filter

enode_id	measure_name	time	measure_value:bigint
enodeid	temperature	2022-08-11 04:32:12.148000000	92

Feedback Looking for language selection? Find it in the new Unified Settings.

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

--> Run and Send Data From NodeRed and Check again.

Activities Google Chrome Aug 11 09:59

us-east-1.console.aws.amazon.com/timestream/home?region=us-east-1#query-editor:

MyTimeStreamTable

Run Save Clear

Table details Query results Output

Rows returned (10)

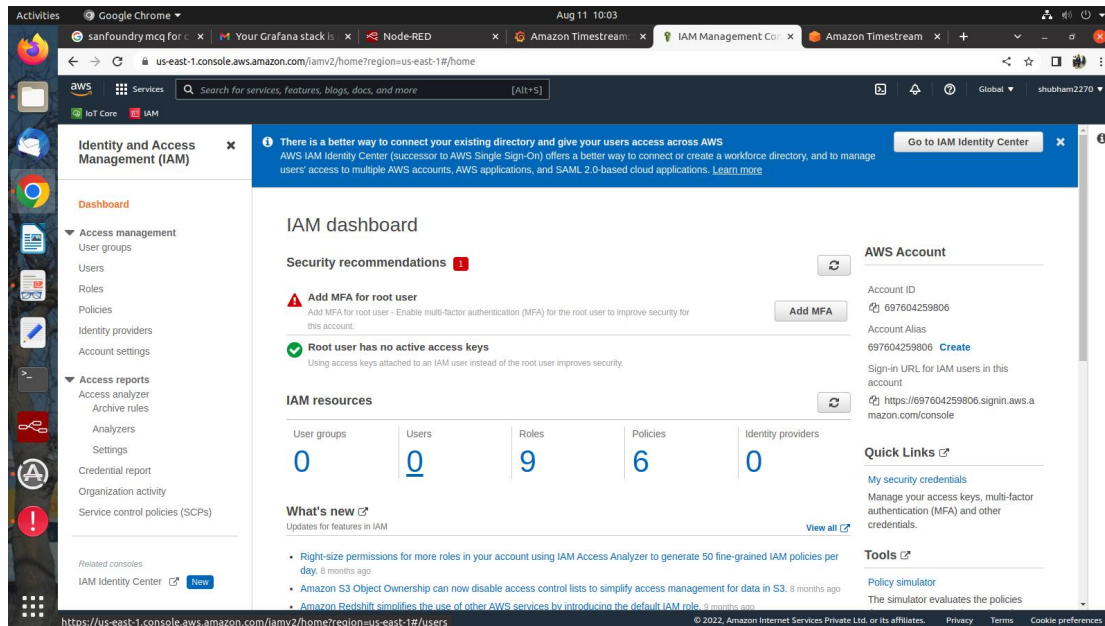
Filter

ID	measure_name	time	measure_value:bigint
enode_id	temperature	2022-08-11 04:31:26.670000000	50
enode_id	enode_id	2022-08-11 04:31:26.670000000	2
enode_id	humidity	2022-08-11 04:31:26.670000000	67
enode_id	timestamp	2022-08-11 04:31:26.670000000	1660192185796
enode_id	timestamp	2022-08-11 04:31:26.310000000	1660192185431
enode_id	enode_id	2022-08-11 04:31:26.310000000	3
enode_id	humidity	2022-08-11 04:31:26.310000000	73
enode_id	temperature	2022-08-11 04:31:26.310000000	68
enode_id	humidity	2022-08-11 04:31:25.996000000	77

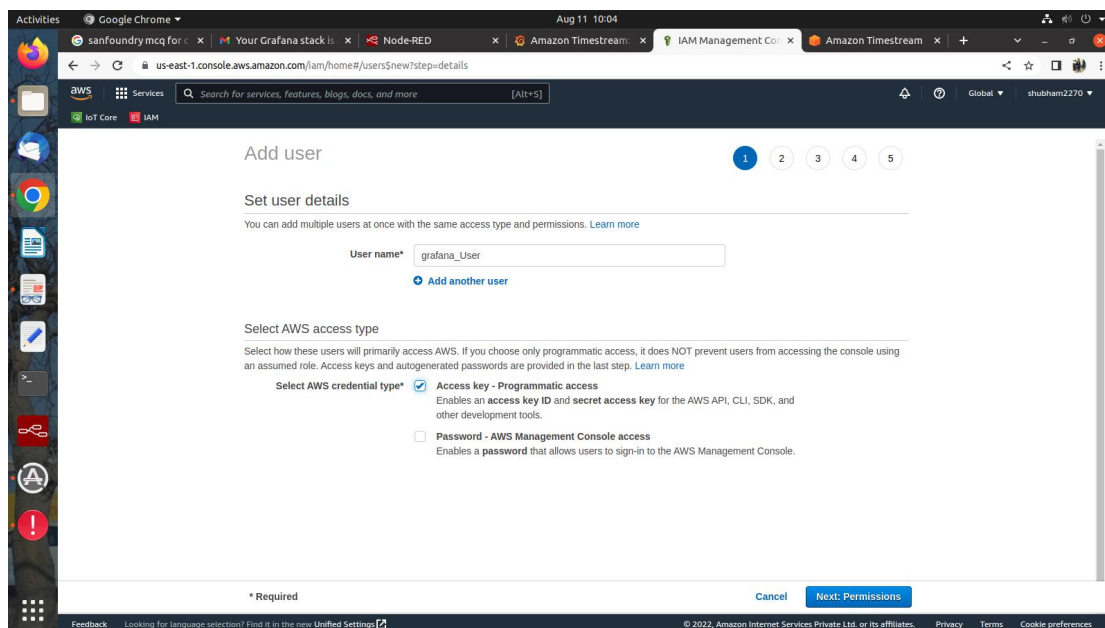
Feedback Looking for language selection? Find it in the new Unified Settings.

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

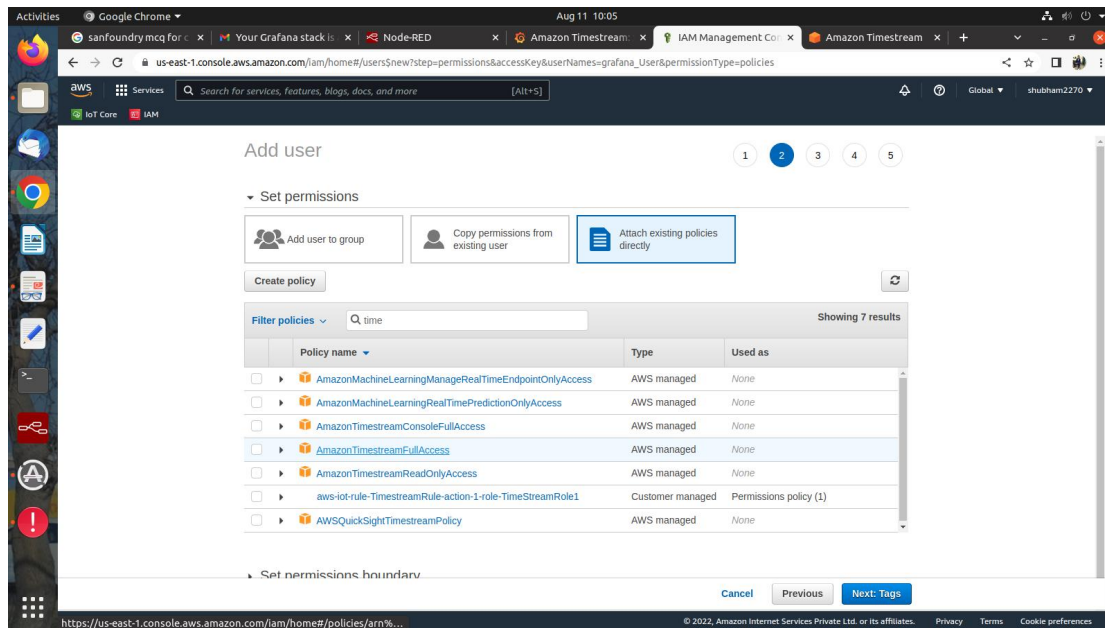
8) To give Access to Grafana --> Goto IAM service --> Users --> Add Users



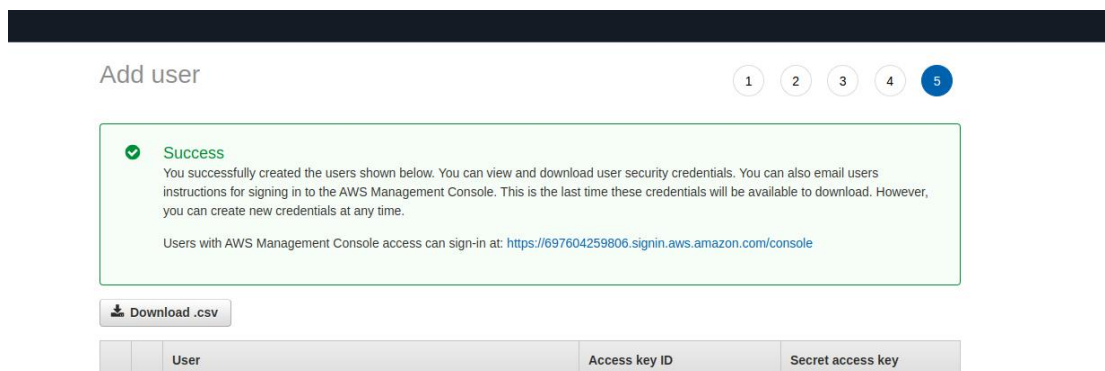
--> userName(any) & select(tick on) Access Key and click next



--> click on Attach existing Policies --> select 'Amazon Timestream Full Access' --> next and Create.



9) Now Copy Access Key id & Secret Access key(from AWS) to Respective fields(In Grafana --> Connection Details).
-->then Select Region.



10) After Connection Success DatabaseTable should appear

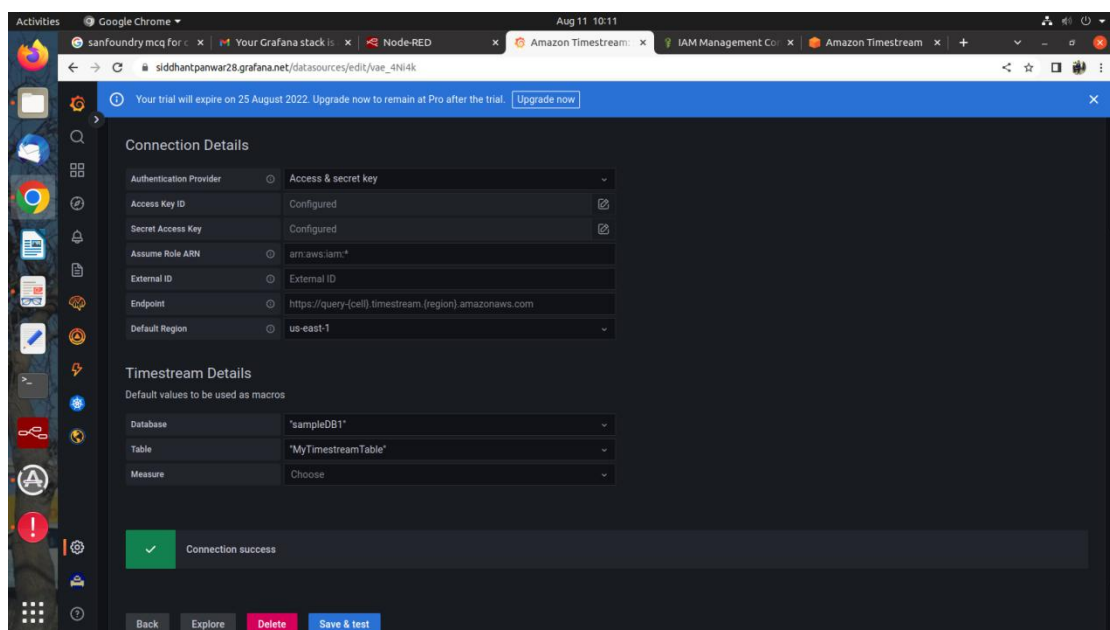
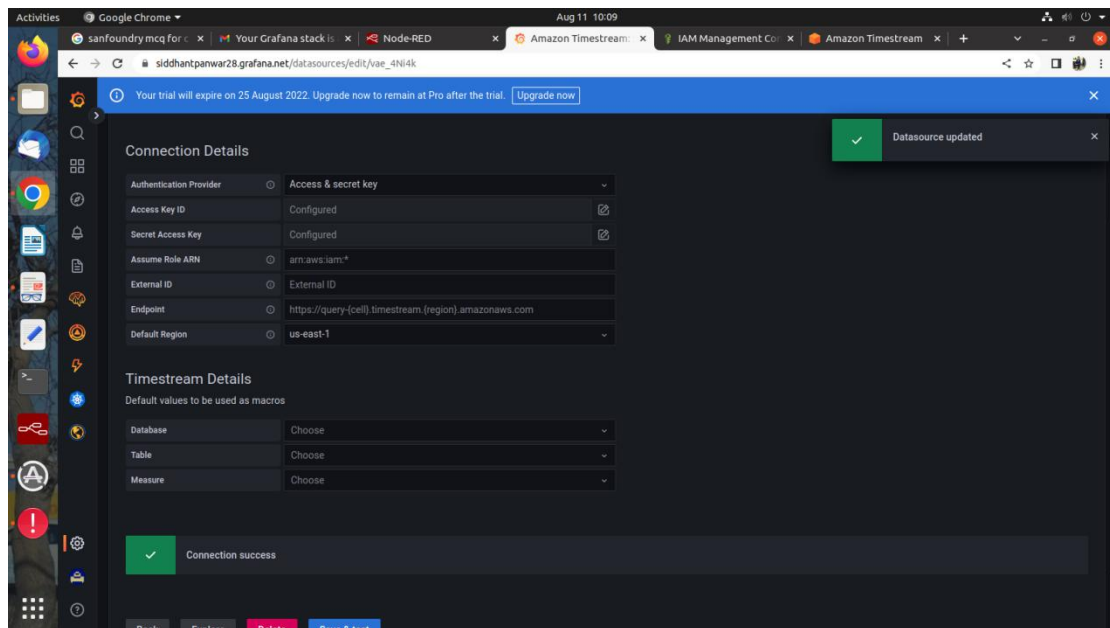
--> in Timestream Details

--> Database = 'SampleDB'

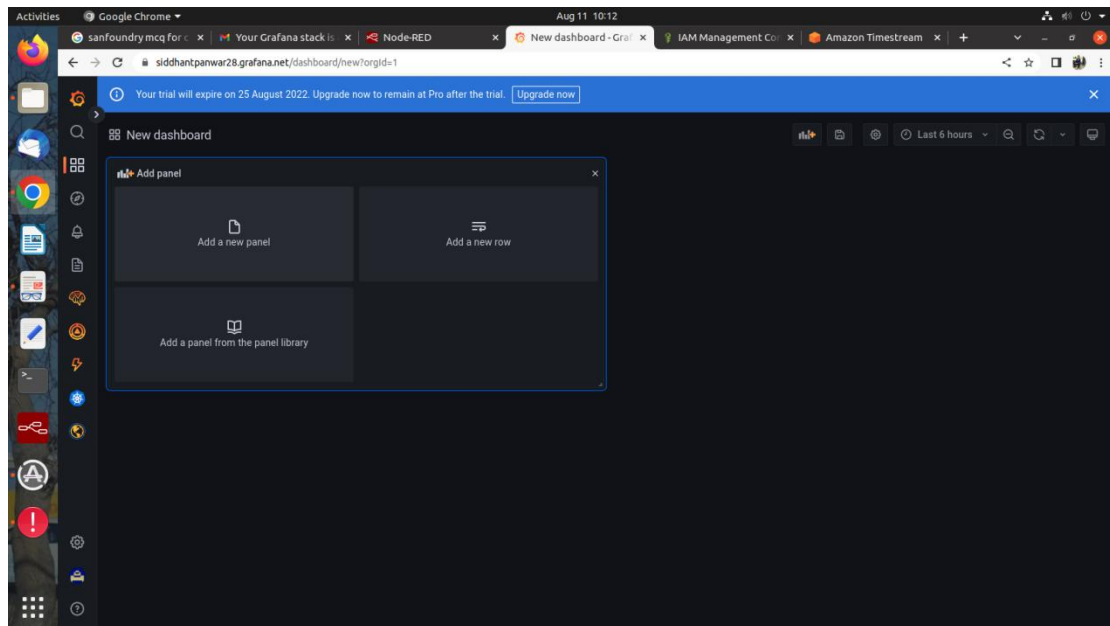
--> Table = 'myStramTable'

--> Measure = can be left

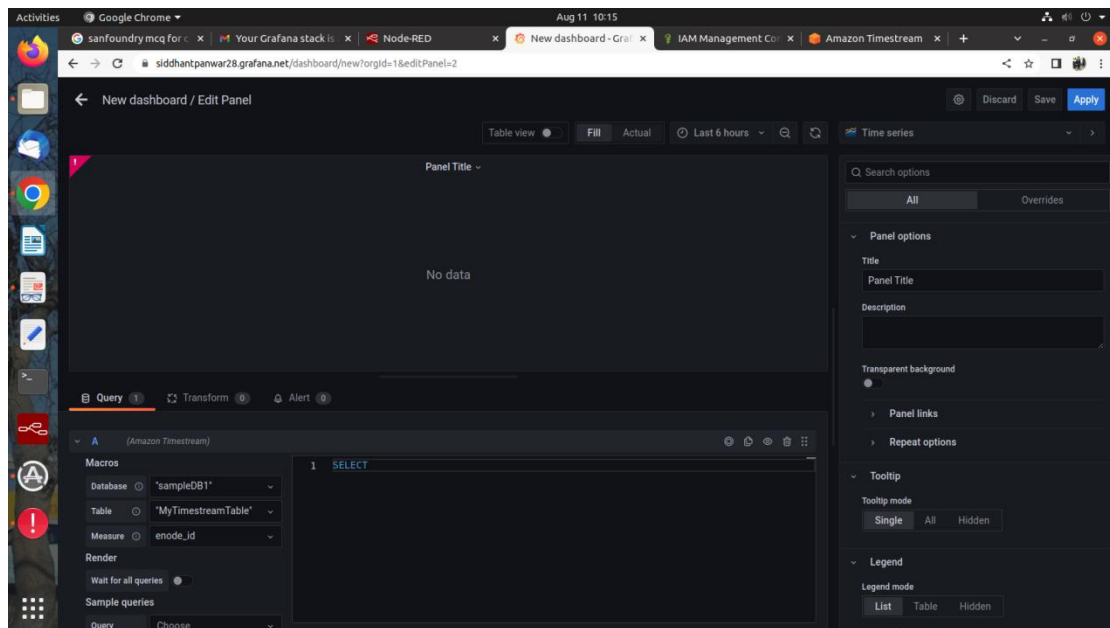
--> click Save & test below.



11) Now Goto Dashboards --> Create New Dashboard --> Add a new Panel



--> in 'Data source' --> select 'Amazon Timestream'



--> **Right Side Name the Table in 'Title'**

--> **in Sample Queries --> Select 'First 10 Rows'.**

--> **Now in Terminal(at line 1) --> Add SQL Query --> (different for different plots)**

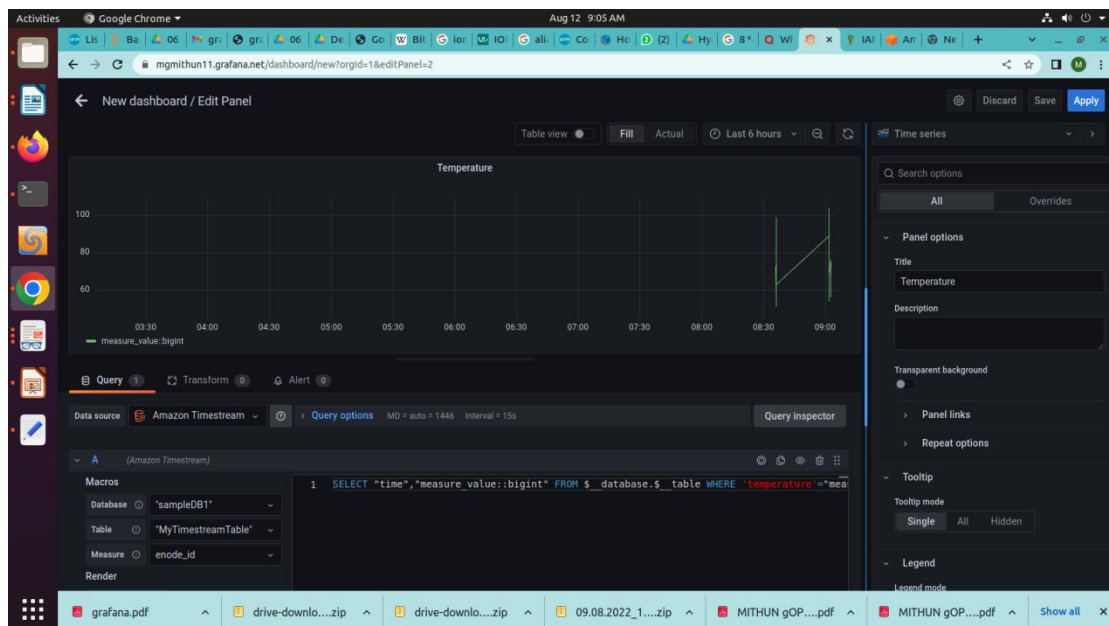
--> **For Humidity Plot -->**

**SELECT "time","measure_value::bigint " FROM
\$__database.\$__table WHERE 'humidity' = "measure_name"**

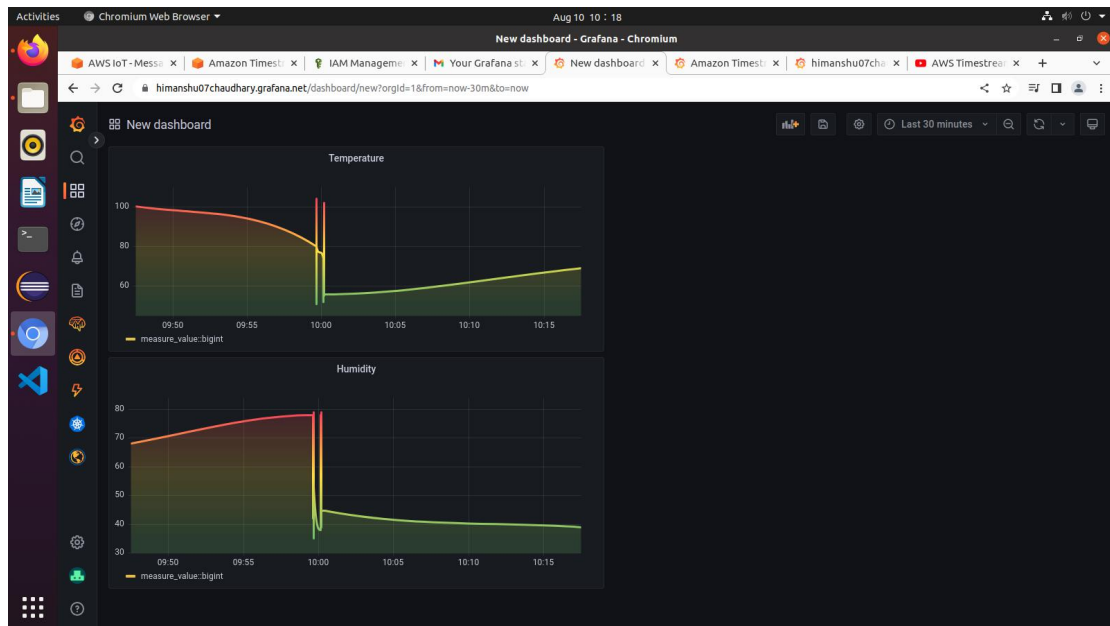
--> **For Temperature Plot -->**

**SELECT "time","measure_value::bigint" FROM
\$__database.\$__table WHERE 'temperature' = "measure_name"**

--> **then Click Apply.**



12) Now Send Data from NodeRed and Refresh For graph plot.



==== THANK YOU!!!! ==