# CS 524

# *COURSE PROJECT*

*Guided by : Prof. Igor Faynberg*

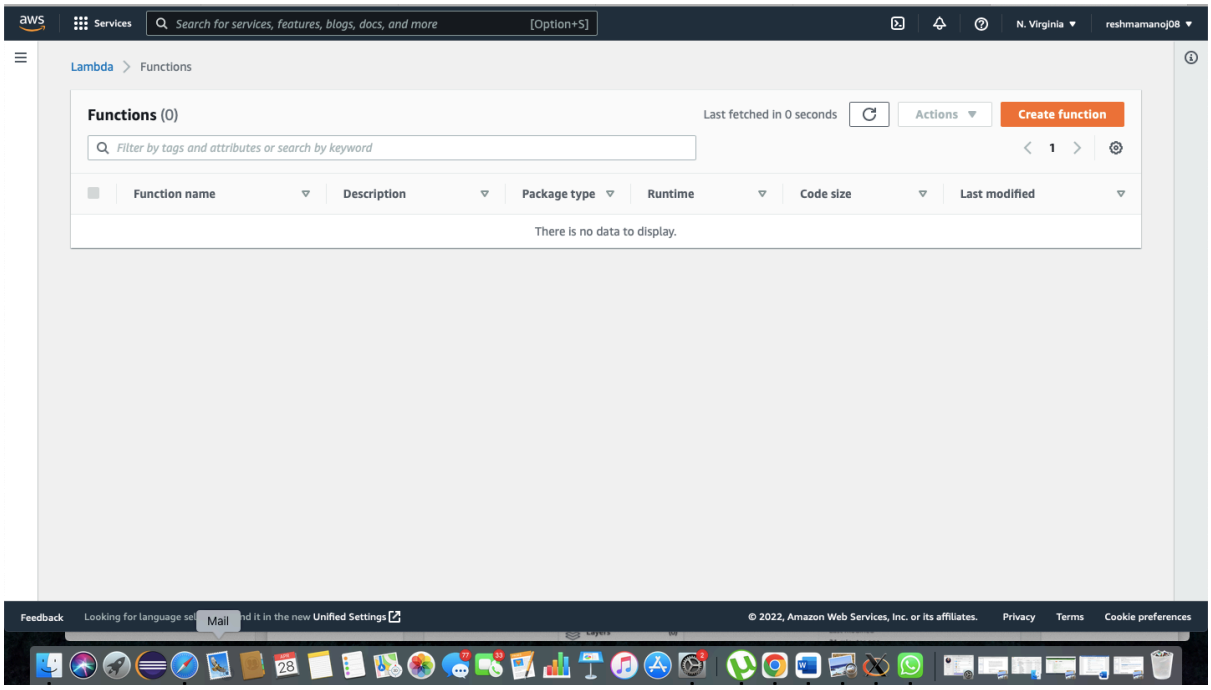
**Name : Reshma Manoj Kumar**
**CWID : 20007266**
**Course Section : CS 524 – A**

## Part – A - AWS

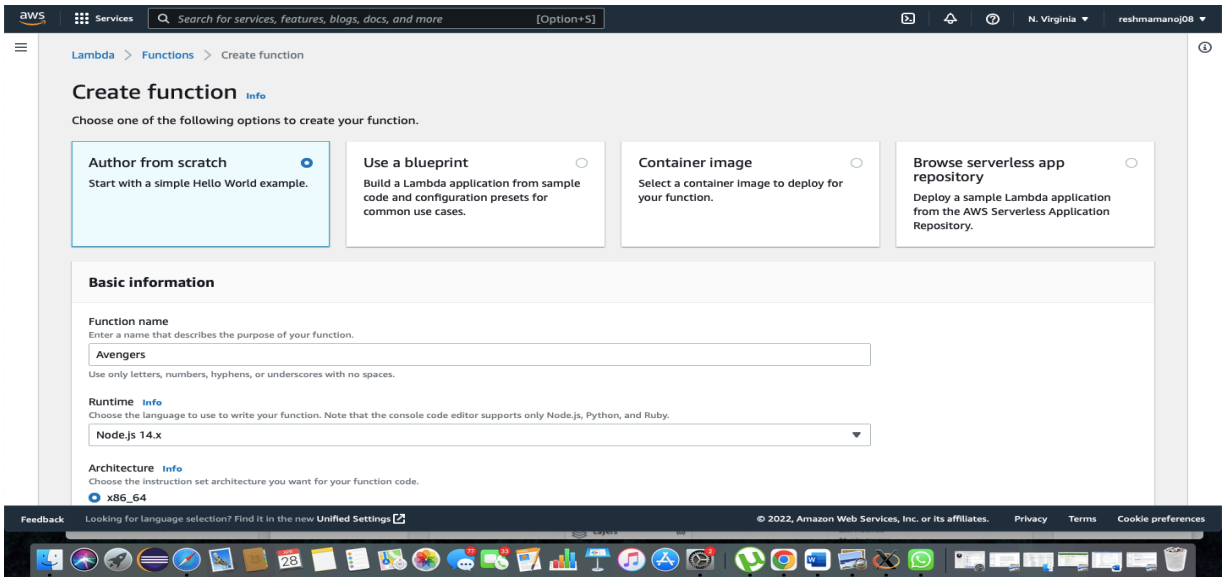
AWS Lambda is a serverless computing platform provided by AWS. We don't require a server, operating system, memory, or anything else in order to achieve this. Simply build our code and publish it using HTTP or REST API.
It indicates how many resources were used and how much time was billed. It's also reasonably priced.
Cloudwatch logs can also be used to view logs. It displays how many times the code has been called.

1) We need to sign into our aws account

2) Now, select lambda from services.

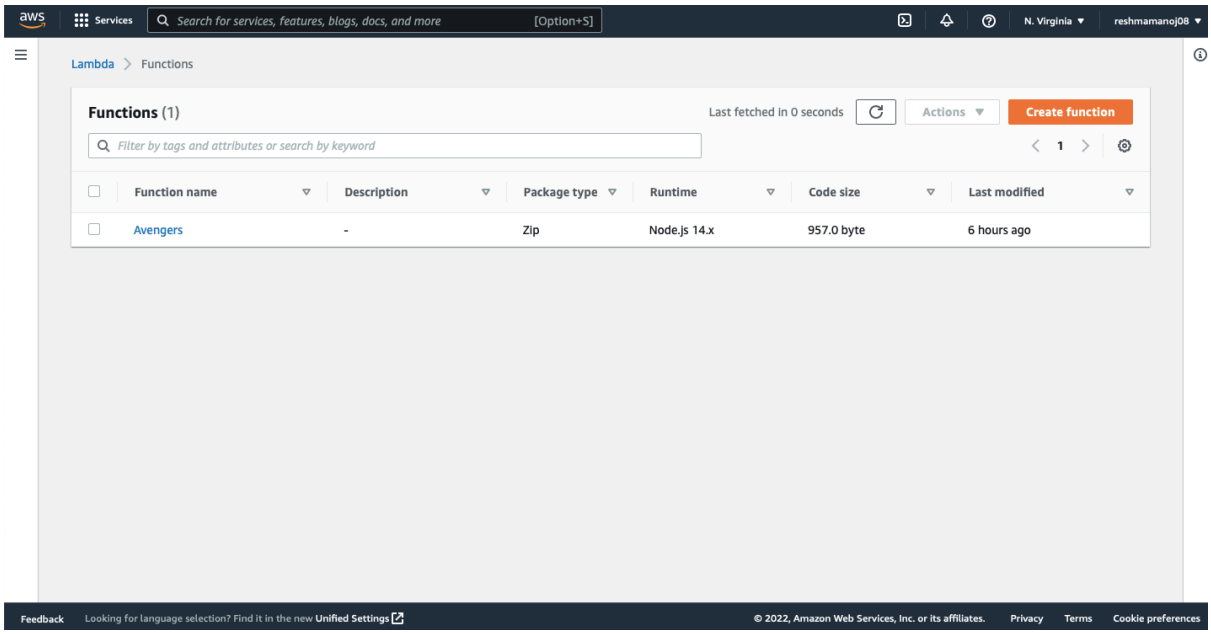



Now we have to create function for lambda. Choose **Author from Scratch**.

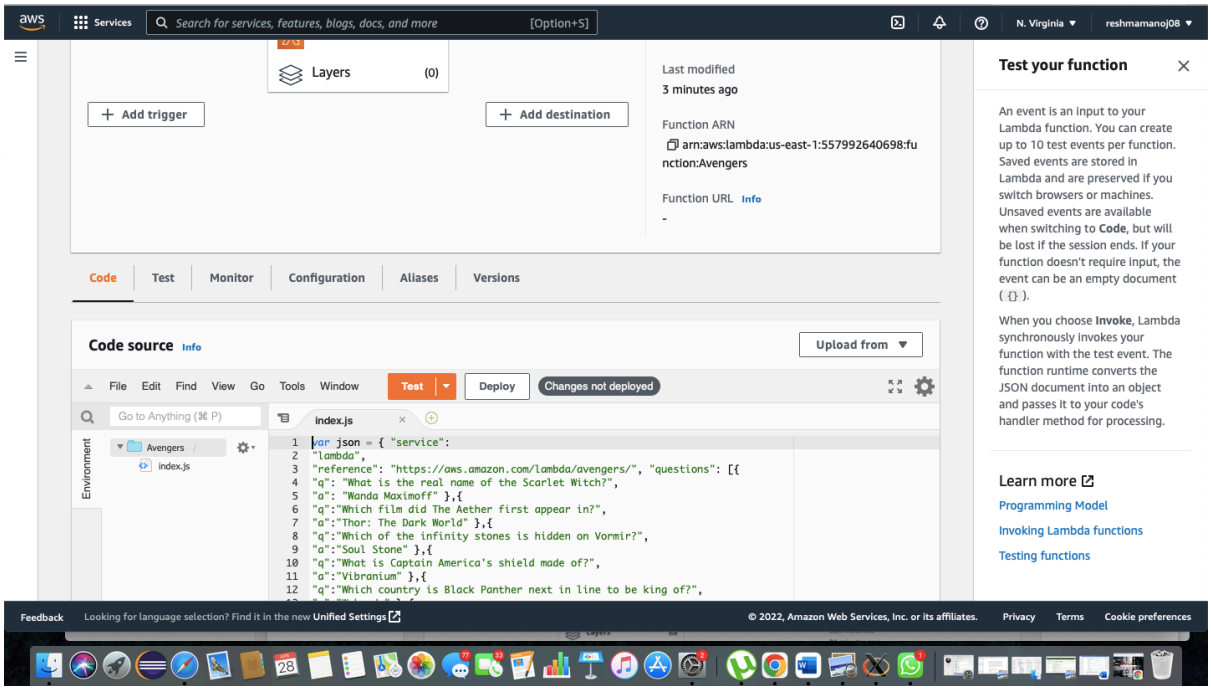


We are choosing **Author from scratch** to deploy our code.

The Lambda function "**Avengers**" has been successfully created.

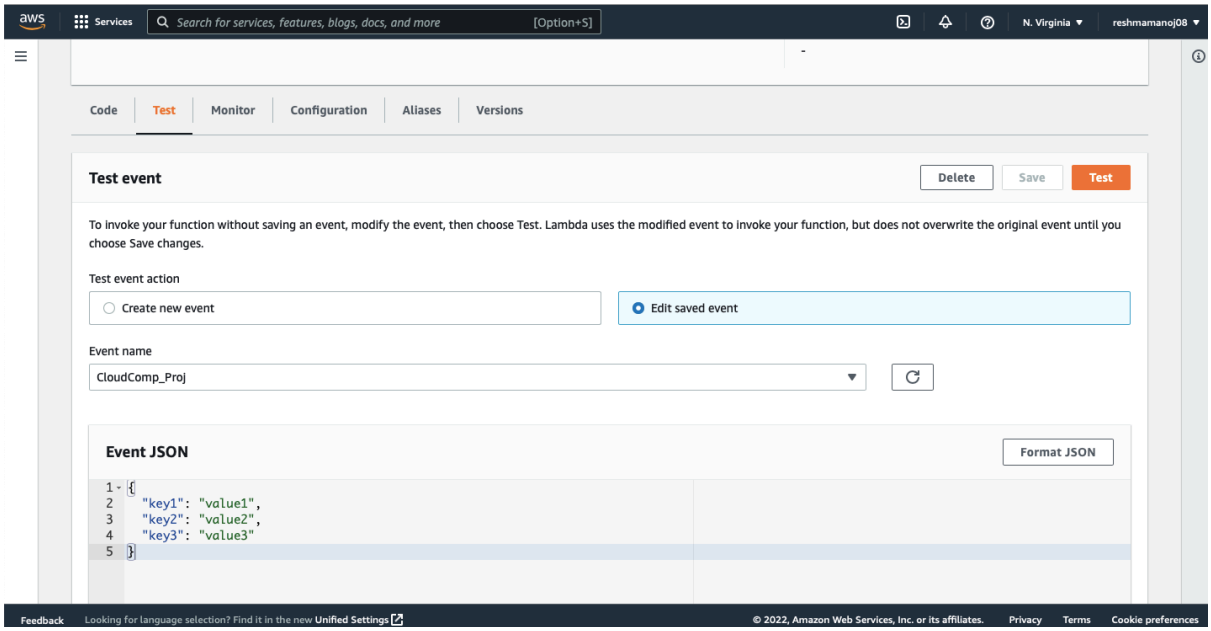


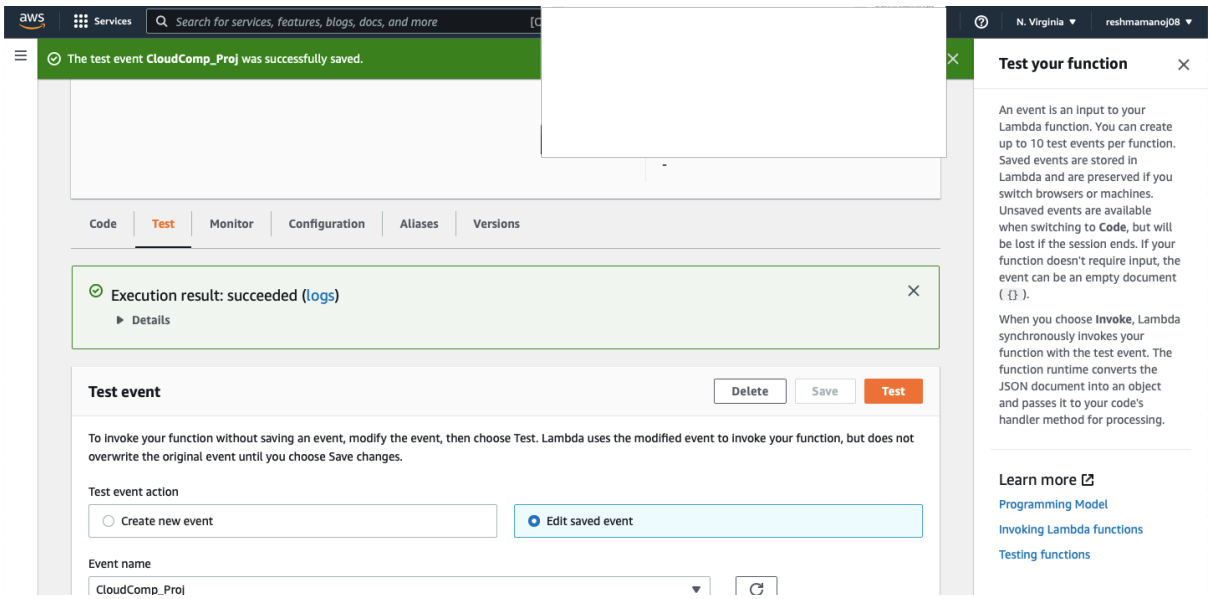

Source code provided should be added now under the Code Source section.



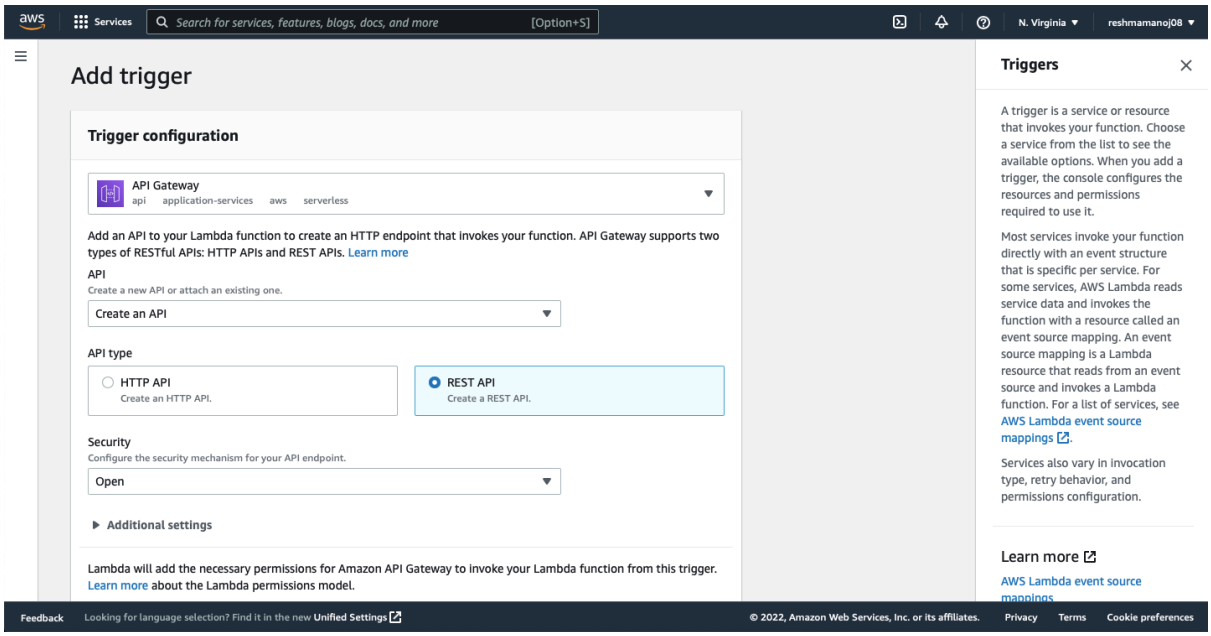

After adding the source code. Configure the test event and create a test.

Next step : Configure test event. Here I am naming the event name as CloudComp_Proj. hello-world is the event template used.

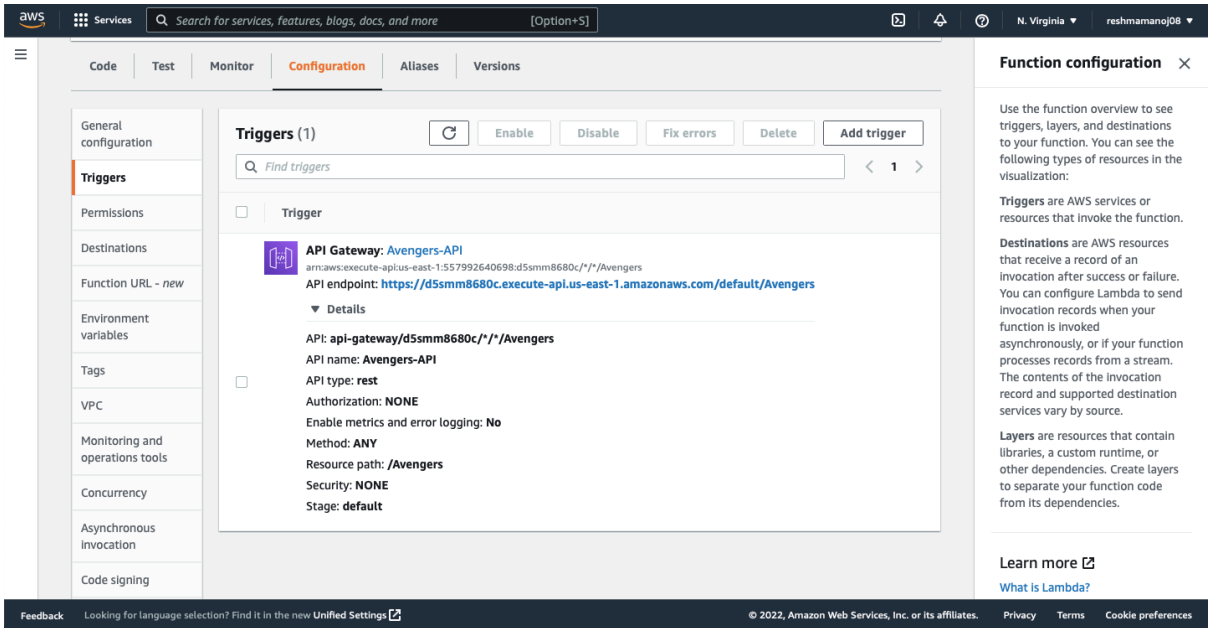



After deploying our lambda function, we must add an API Gateway as the trigger to the Lambda service, which should be implemented as **REST API**.




Go to Configuration. Copy and paste the URL of API endpoint in a new browser tab.

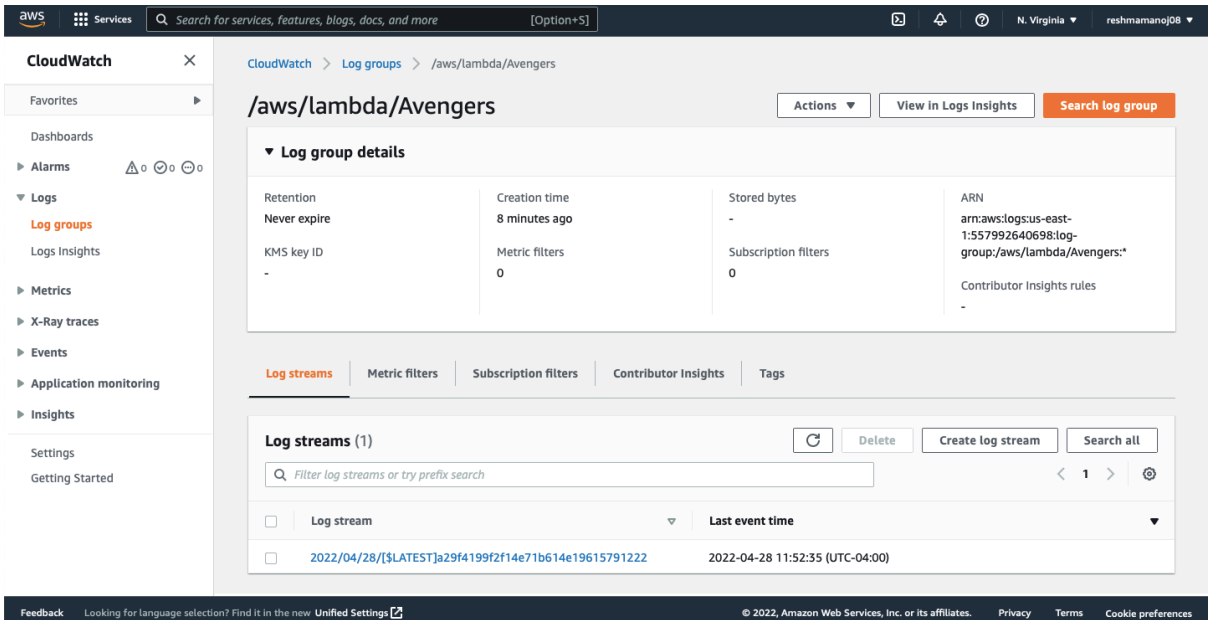


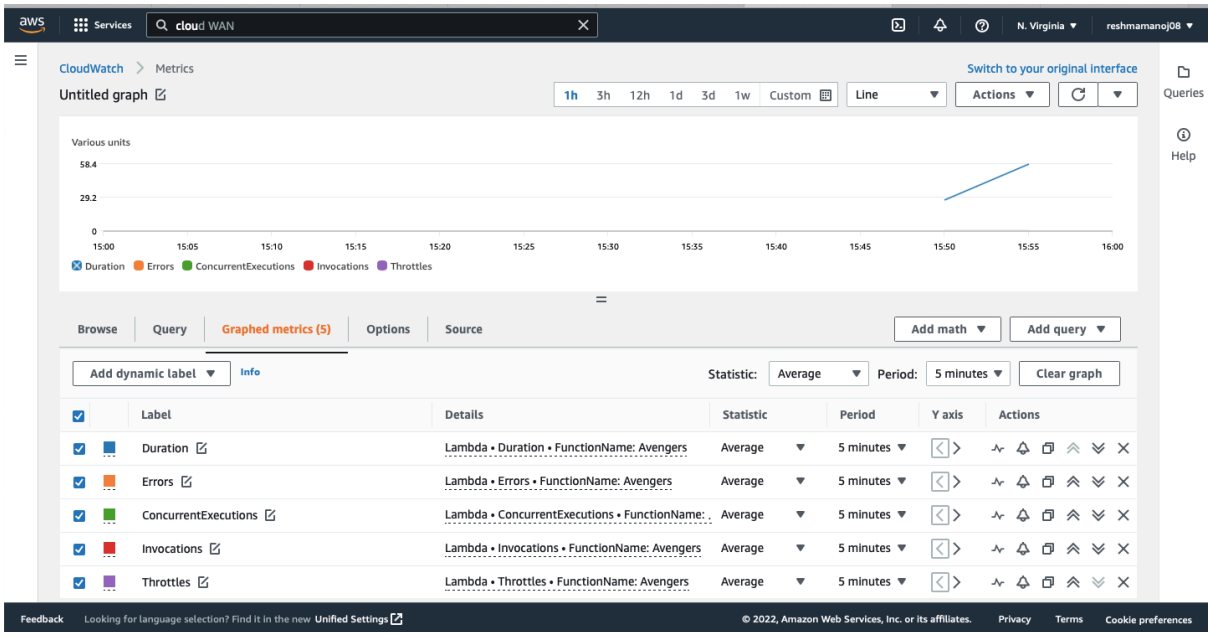

We get results. Each time we reload the browser tab with the URL, we get different results.

```
{"q":"What is Loki's title?","a":"God of Mischief"}
```

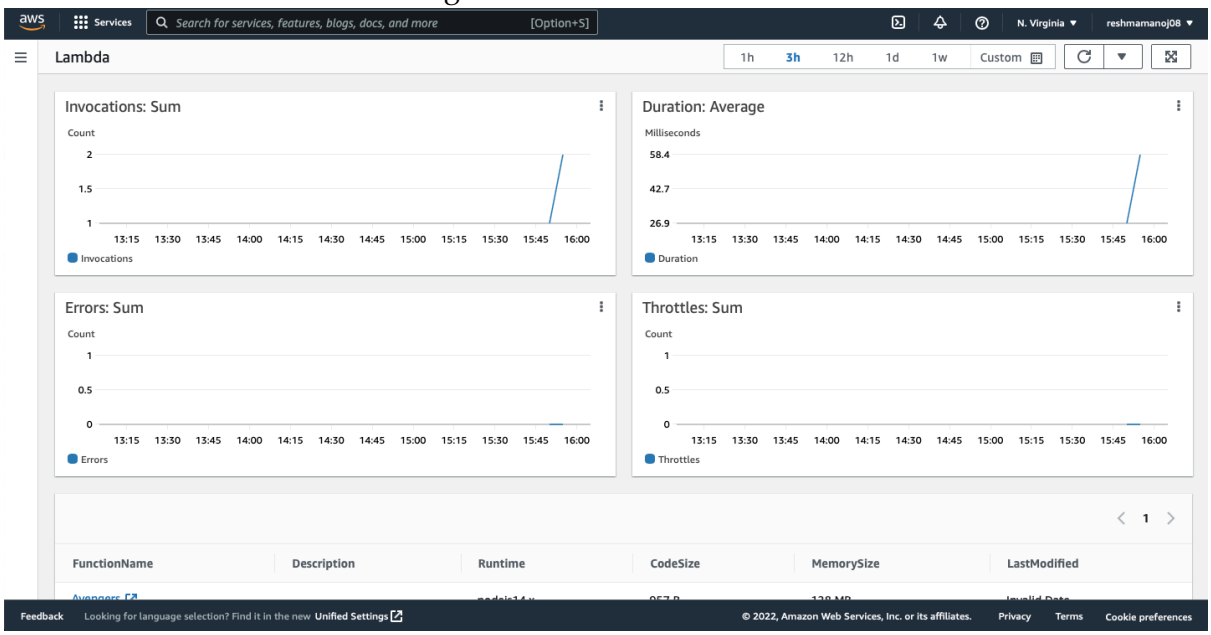
{"q":"What is the name of the organisation which is revealed to have infiltrated S.H.I.E.L.D. in Captain America: The Winter Soldier?","a":"Hydra"}

Open CloudWatch from the list of services. It is used to see logs and metrics.
Invoke the Lambda function and view **CloudWatch** logs for this function.



These are the metrics seen from logs of Lambda.

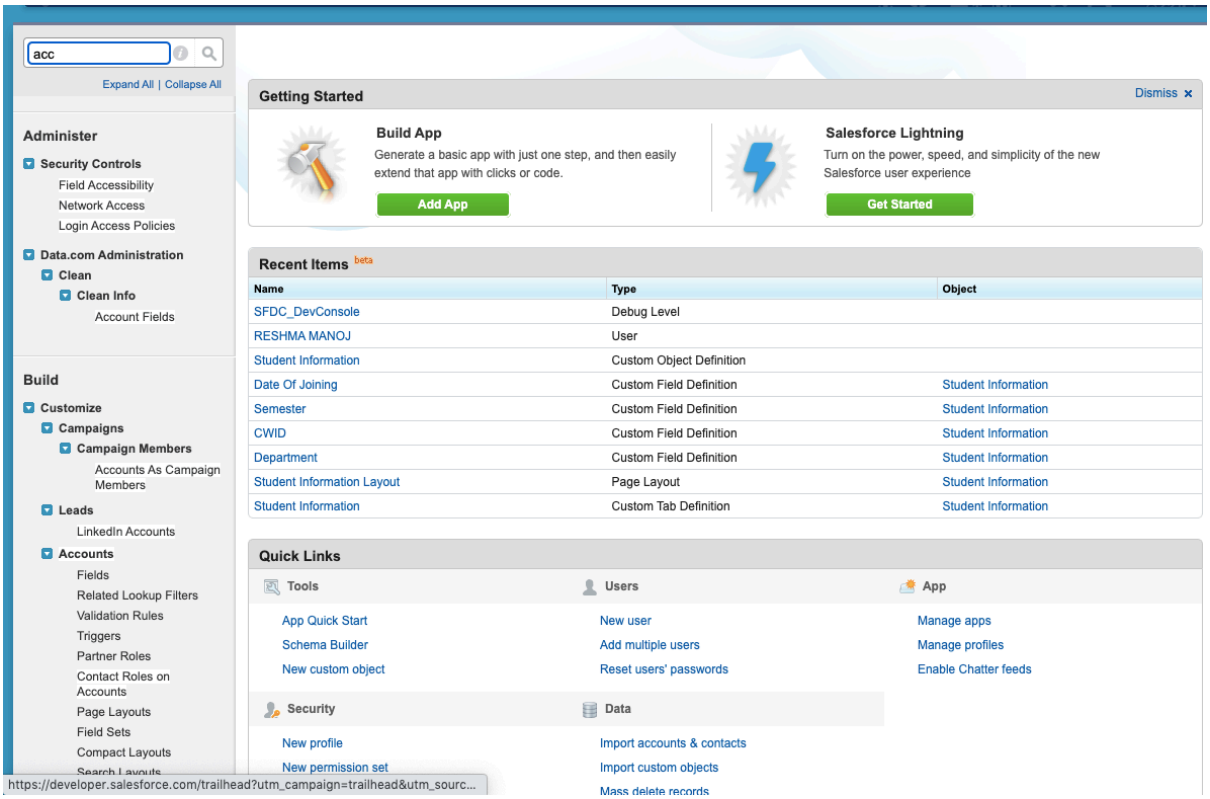



## Part – B – Salesforce

In this part we will be creating a custom field on the *Standard Object* **Account** and creating Contact record and the Apex trigger on ***Contact Object***. When we create a new record on the Contact object the custom field on the Account object needs to be updated with the help of the Apex trigger.

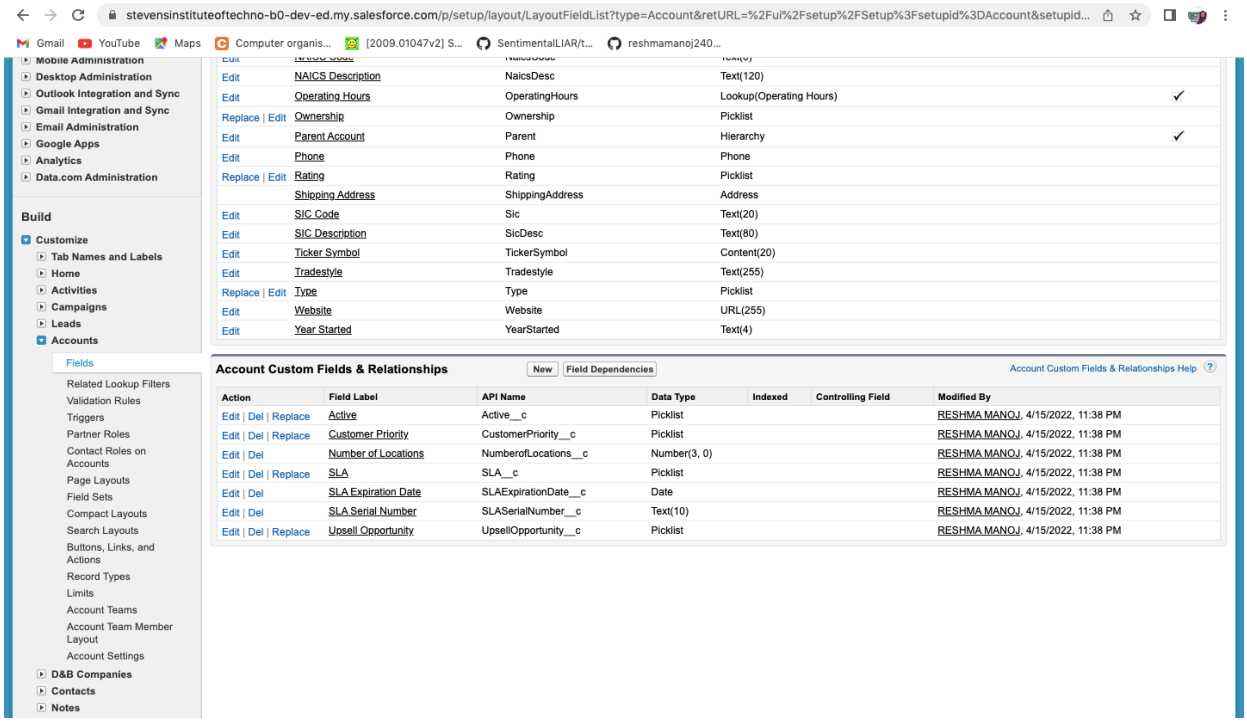
1) **Creating a custom field on Account Object :**
Now, we will be creating a custom field name **Field Update** on the **Account** object.

   a) Go to **Quick Find/ Search** box and Search Accounts and click on Fields under Accounts.




Choose **Fields** under Accounts as we need to create objects on Accounts of the user.

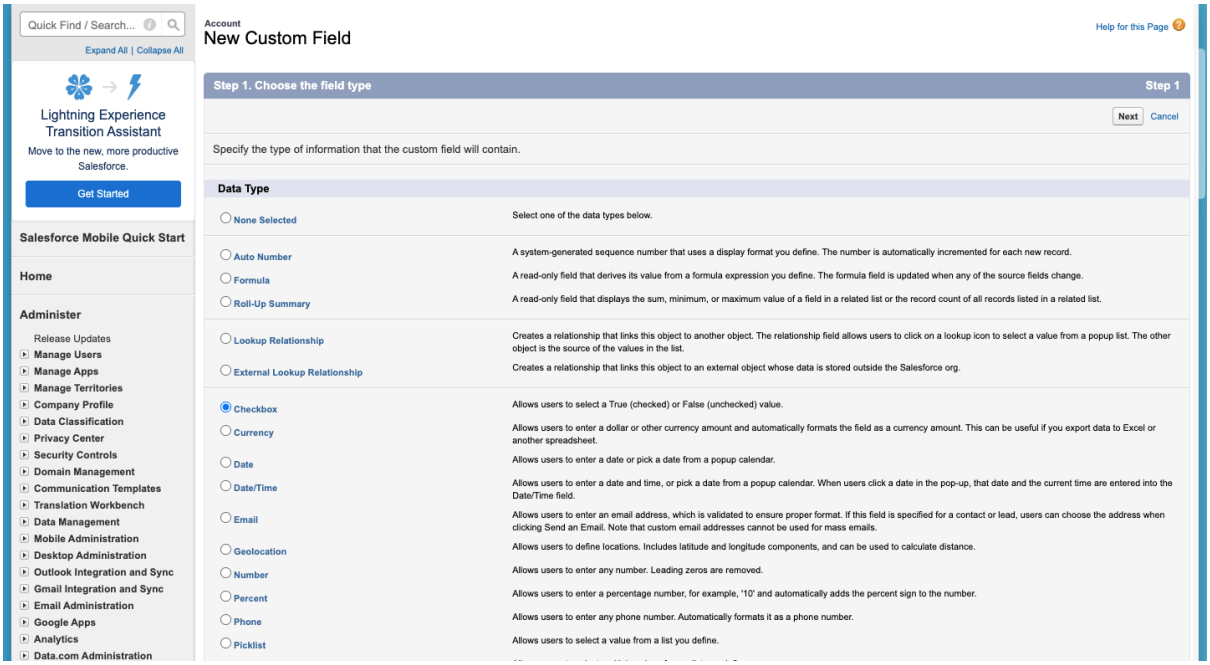
b) All the fields on the Account object will open Go to **Account Custom Fields & Relationships** and Click **New.**



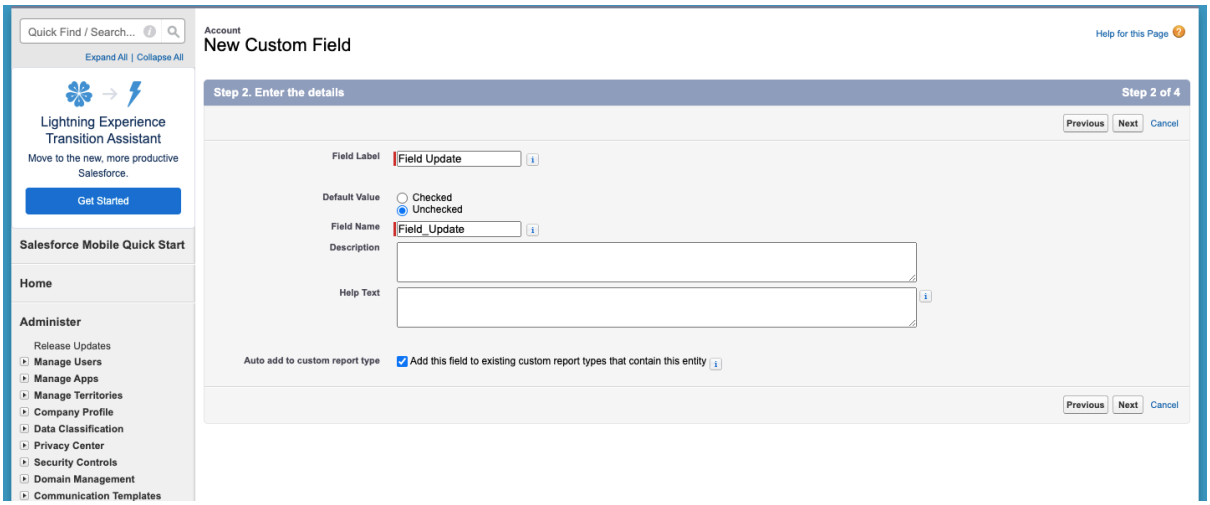
c) Now you will create custom field Field Update on the Account Object with the following information.

**Field Update :**
**Field Type:** Checkbox
**Field Label:** Field Update
**Default Value:** Unchecked
**Field Name:** Field_Update (This will be populated automatically) **Visible:** Checked
**Add to page layouts:** Select all the checkboxes

***Step 1 - Field type : Checkbox***



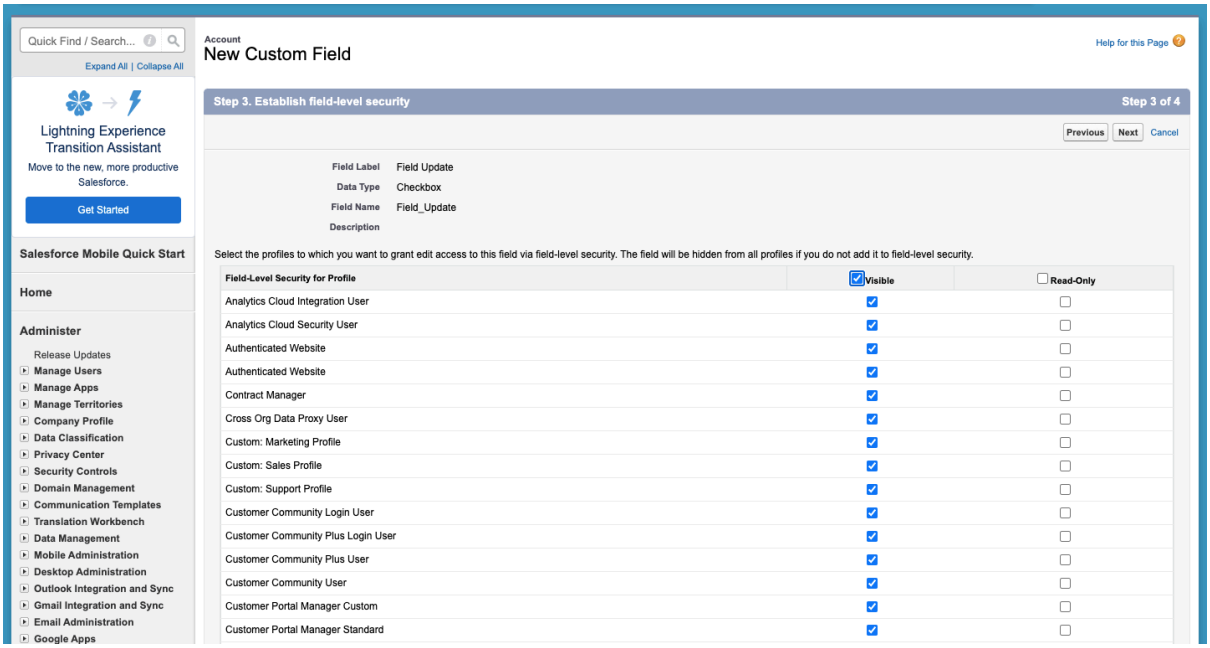

***Step 2 – Field label : Field Update***



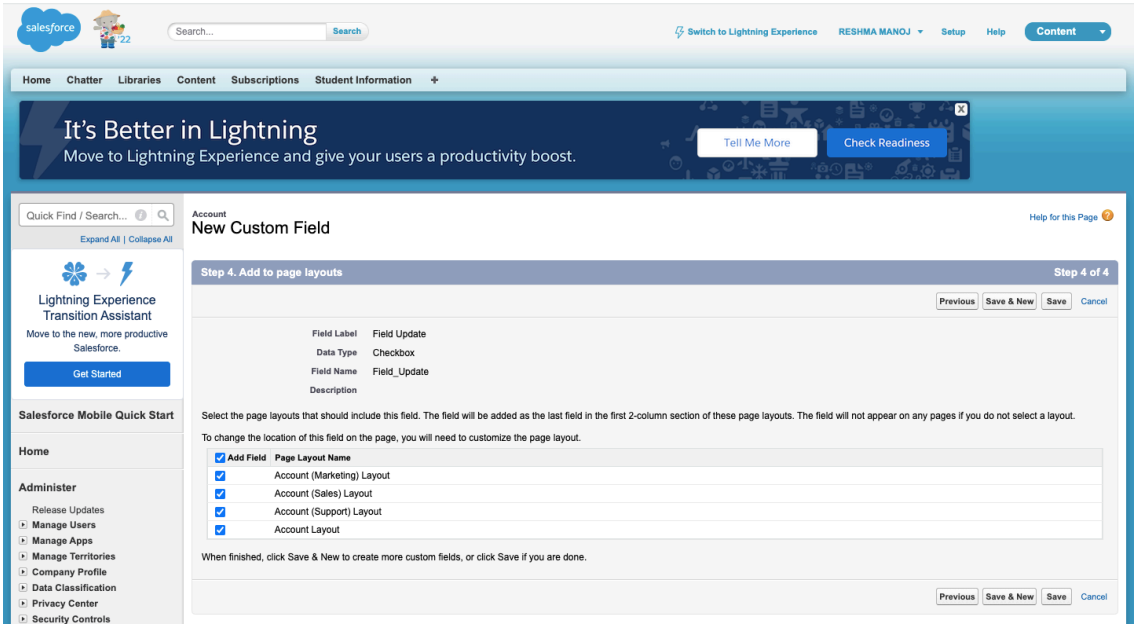

***Step 3 - Default Value: Unchecked ;***

***Field Name:*** *Field_Update (This will be populated automatically)*

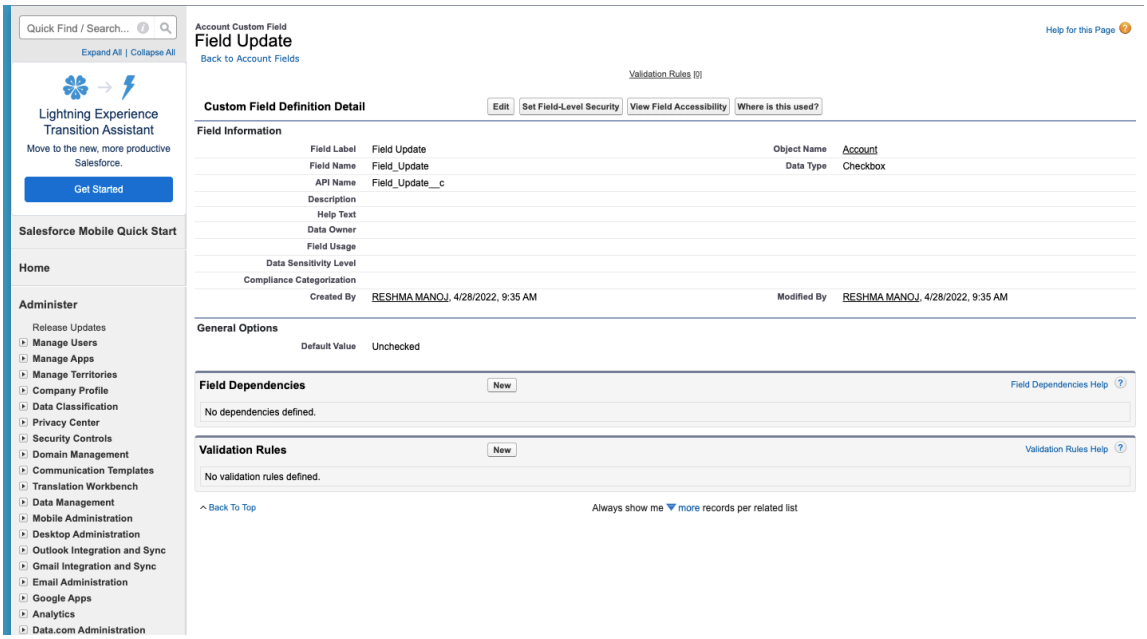
***Security : Visible***



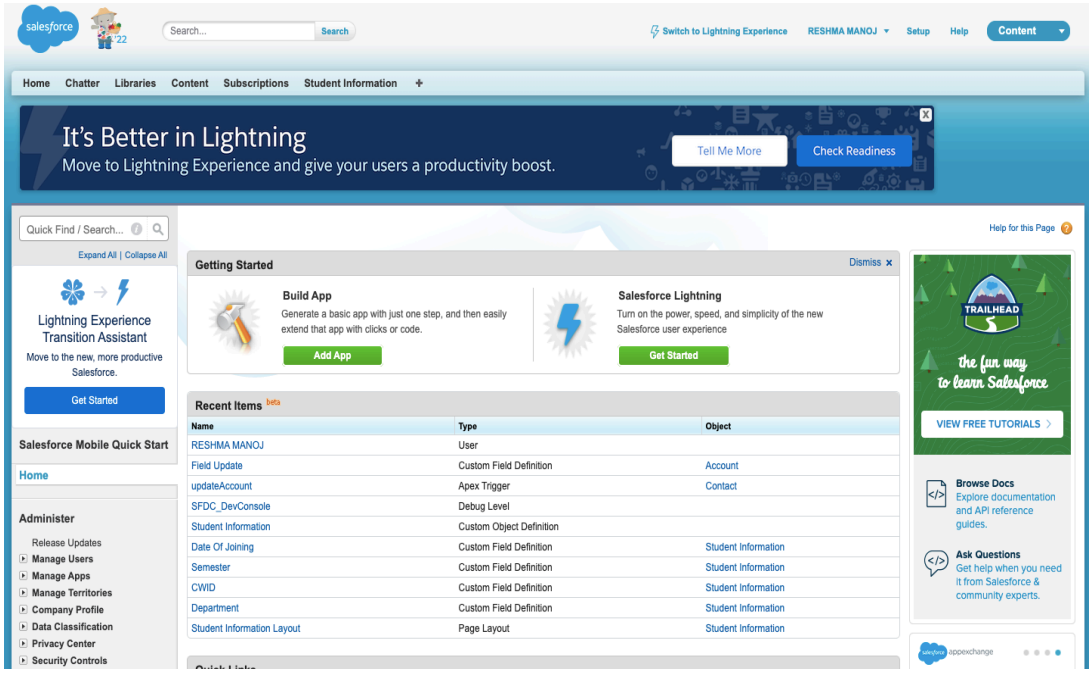

***Step – 4 – Add to page layouts. Select all the checkboxes.***




d) Click on **Save**



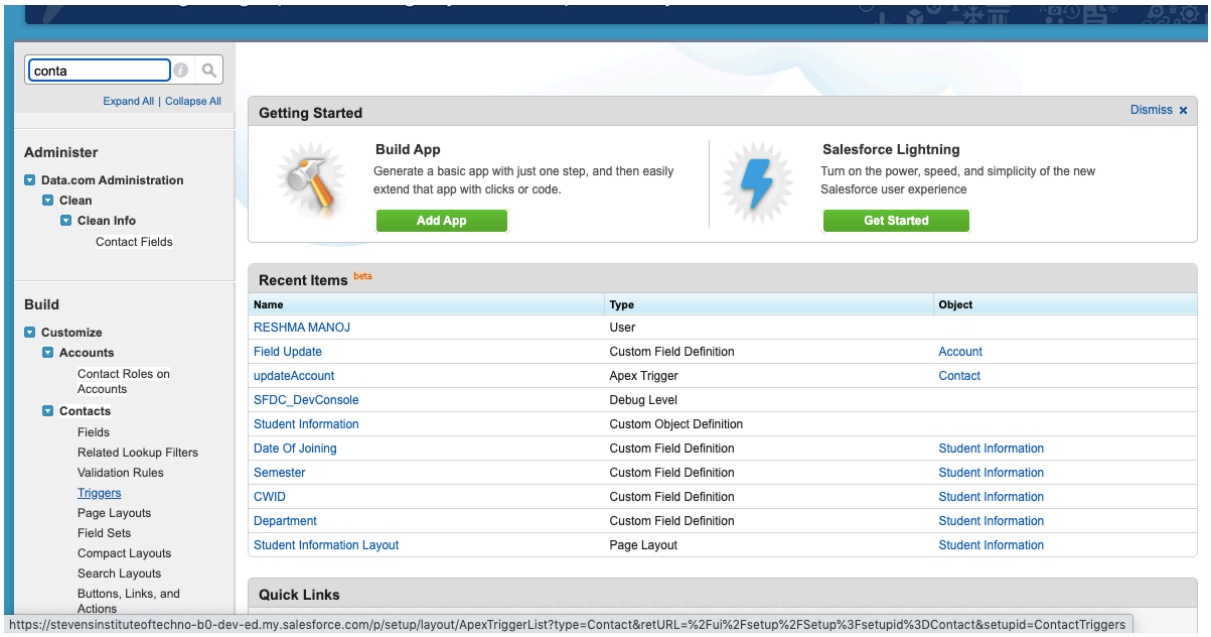

At last, the custom field Field Update is created. Now it is visible among the other previously created fields.



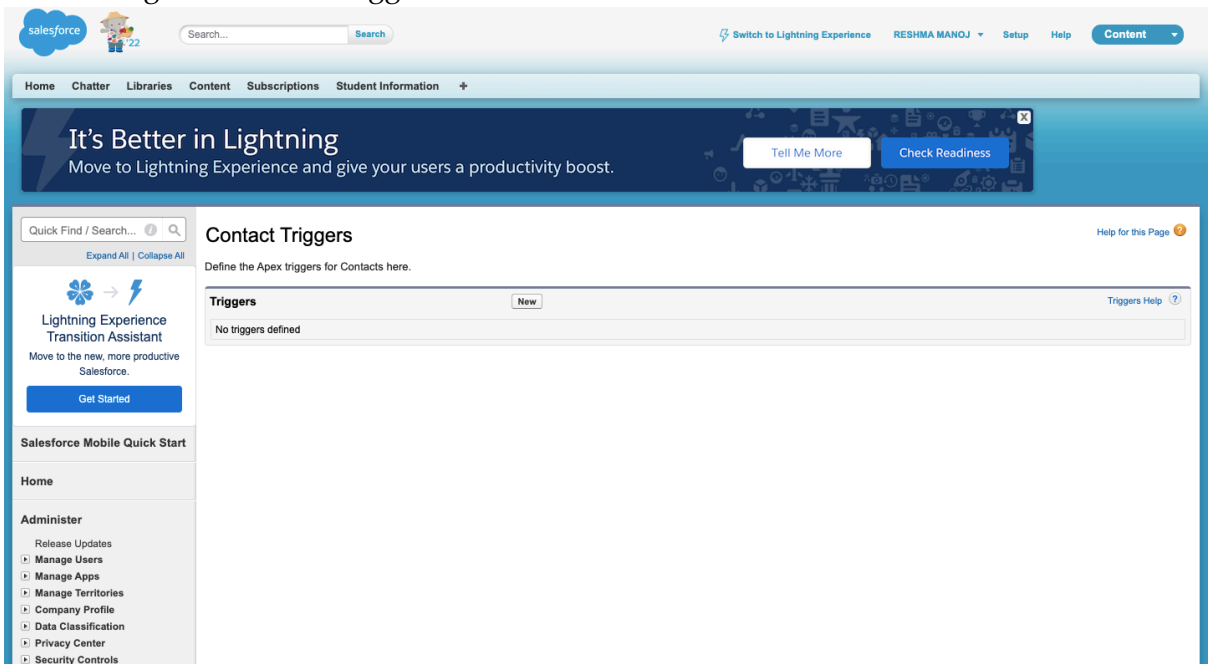

**2) Creating Apex trigger on Contact Object :**

Now we will be creating Apex trigger on the Contact Object.
Steps for are as follows:

a) Go to **Quick Find/ Search** box and Search **Contacts** and Click on **Triggers** which is under Contacts.




It will navigate to contact triggers.

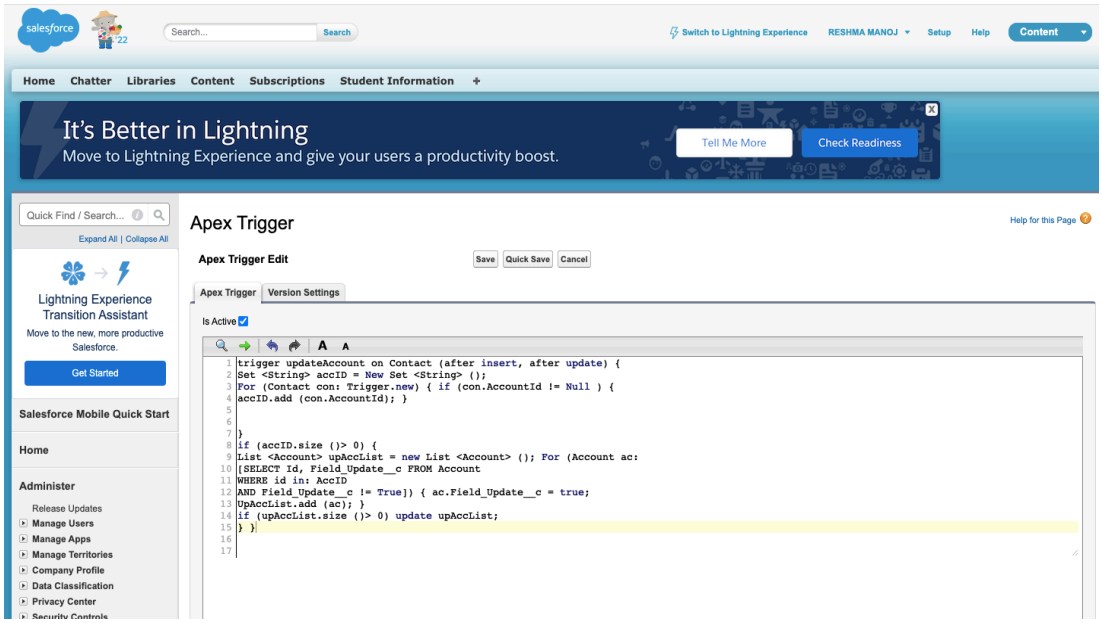



b) Click on **New** and copy paste the following code Make sure before pasting remove everything from the console and then paste.

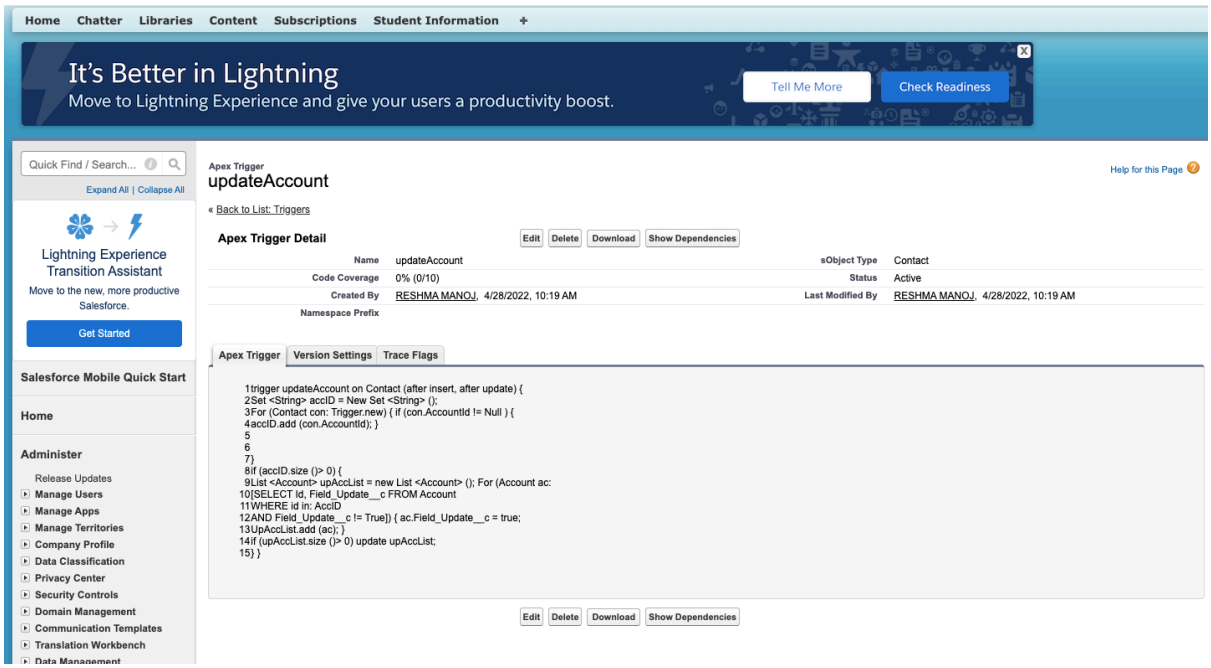
**Code :**

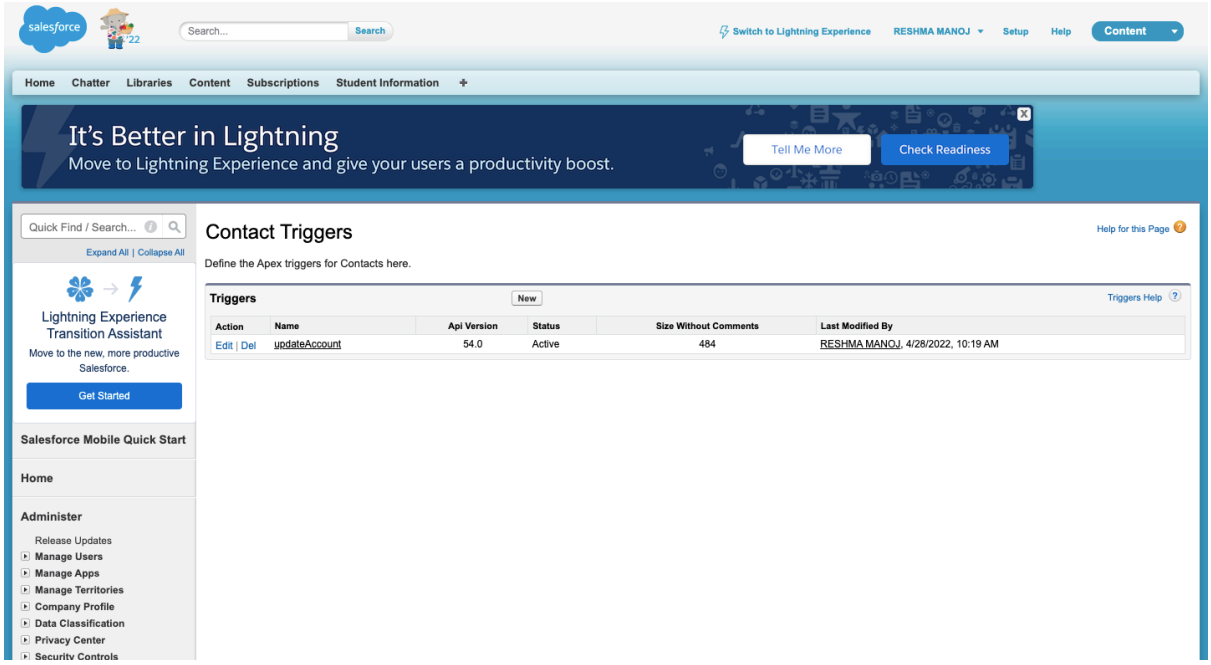
```
trigger updateAccount on Contact (after insert, after update)
{

Set <String> accID = New Set <String> ();

For (Contact con: Trigger.new) { if (con.AccountId != Null ) {

accID.add (con.AccountId); }



}
if (accID.size ()> 0) {

List <Account> upAccList = new List <Account> (); For (Account
ac:

[SELECT Id, Field_Update__c FROM Account
WHERE id in: AccID

AND Field_Update__c != True]) { ac.Field_Update__c = true;

UpAccList.add (ac); }

if (upAccList.size ()> 0) update upAccList;

} }
```

c)After pasting the code onto the console. Save it. Ensure if the trigger is active.



The status is ***Active***.

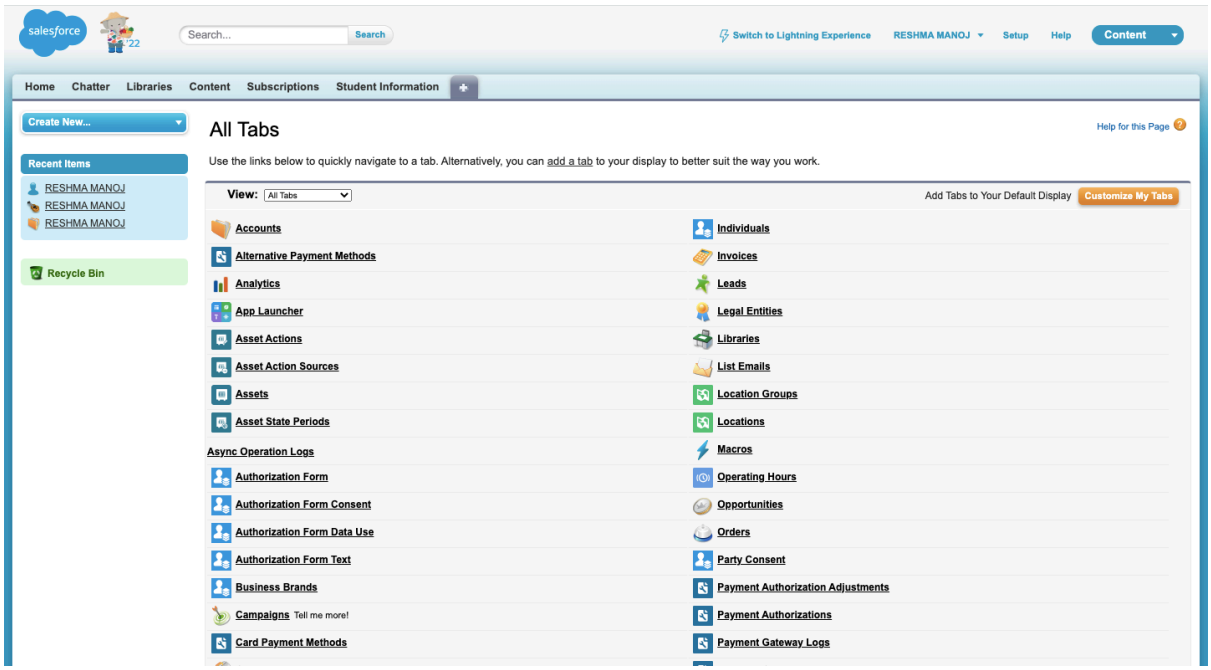


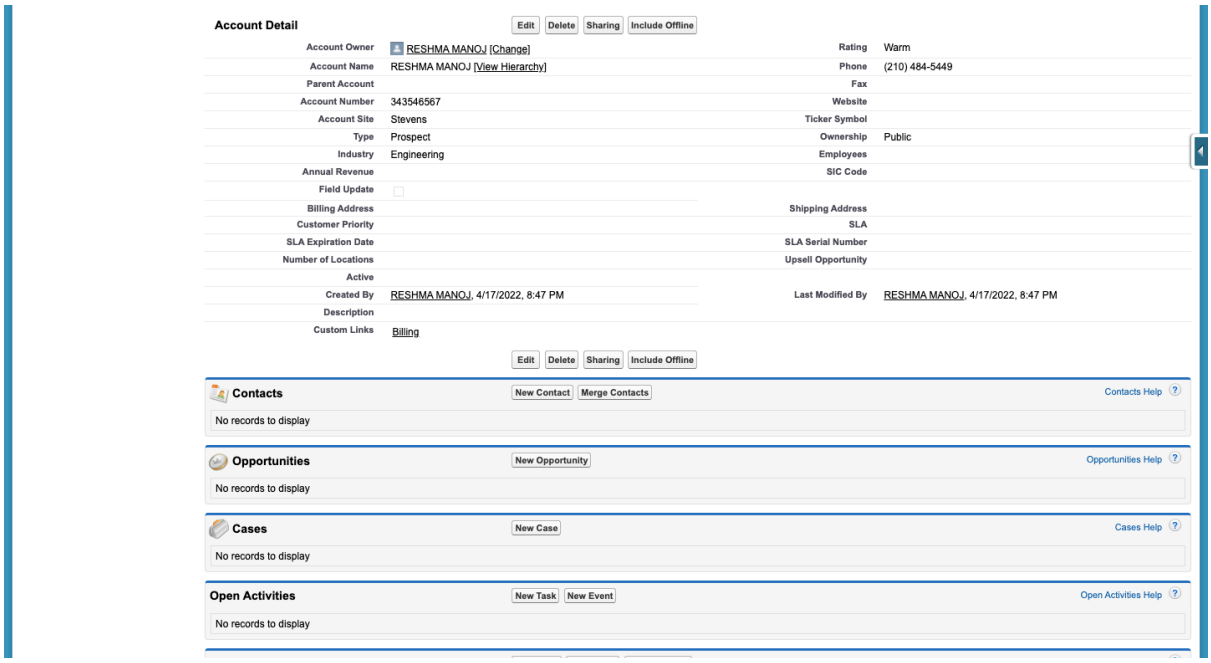

### 3) Creating a Contact record :

Now we will be creating Contact records from the Account object. Steps are as follows:

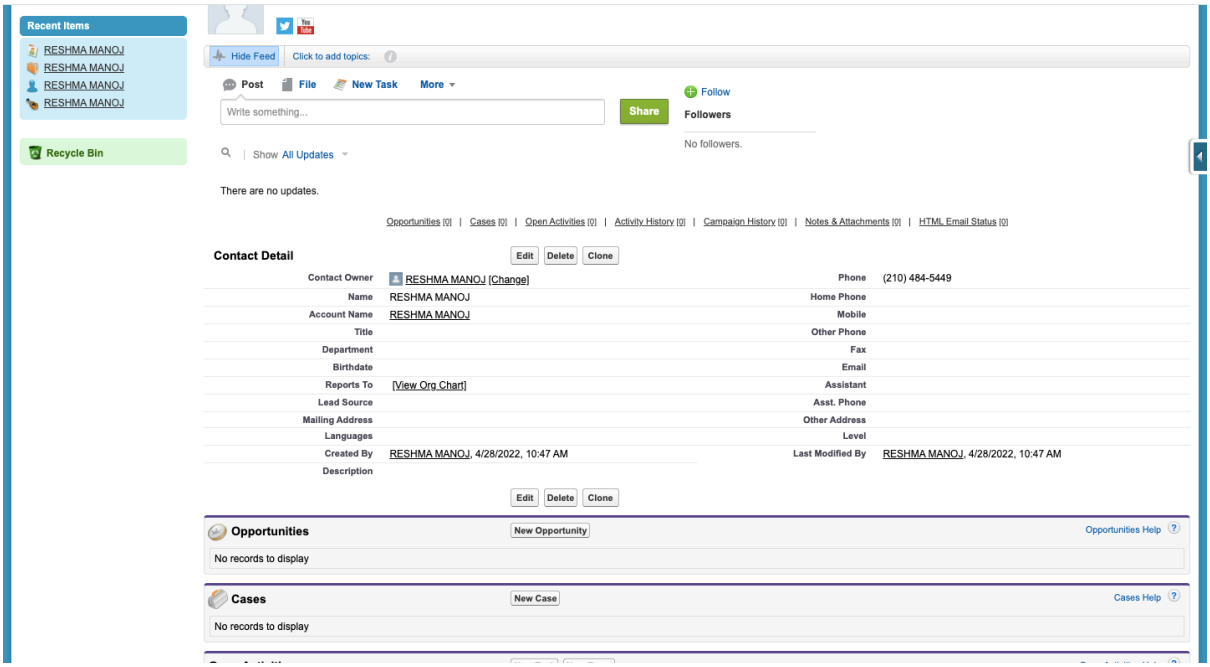
1. a) Go to **Accounts** object it will either on top bar or Click on the **"+"**




b) Go to the record which was created previously by "**Your Information**" on it. You will see the **Field update** field ***unchecked*** on the Account Object page. And scroll down you will be able to see **Contact Object** with no records on it.




**b)** Click on **New Contact** and update the "**Your Contact Information**" and **Save.**




d) Once we create a **Contact** record navigate back to the "**Your Information**" record on the Account
Object.
We will see **Field Update** field on the Account Object is ***Checked*** and "**Your Contact Information**" record is added in the **Contacts.**