

CS 524

COURSE PROJECT

Guided by : Prof. Igor Faynberg

Name : Reshma Manoj Kumar

CWID : 20007266

Course Section : CS 524 – A

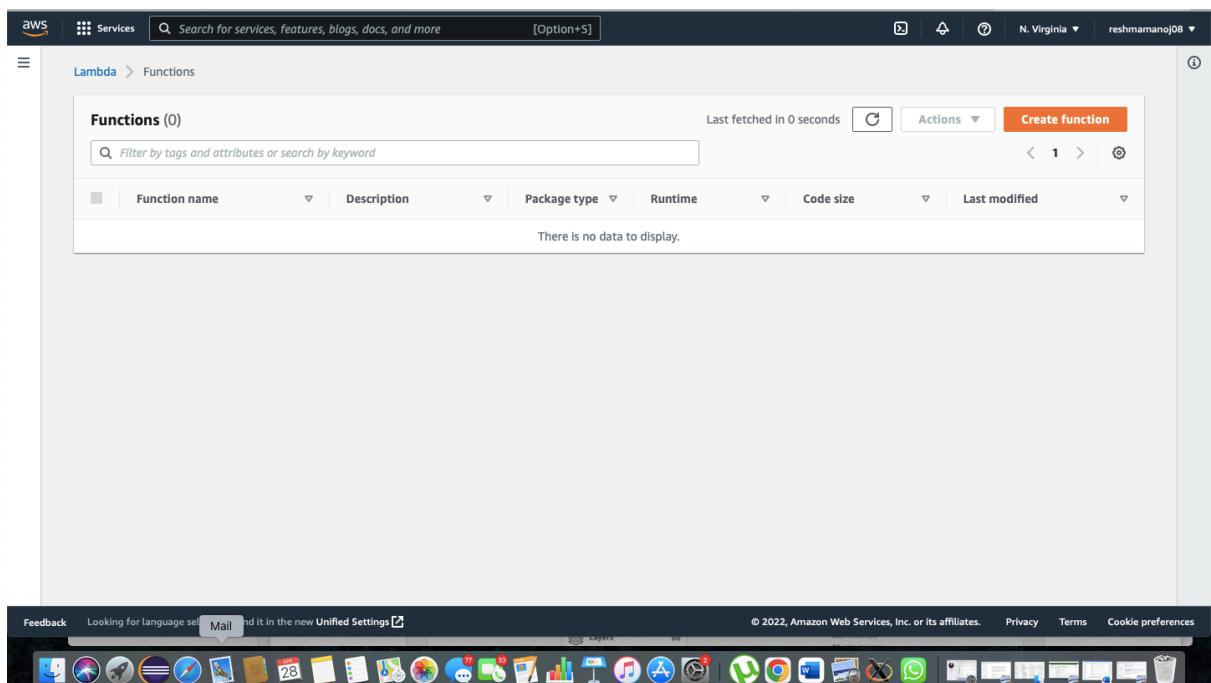
Part - A - AWS

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.

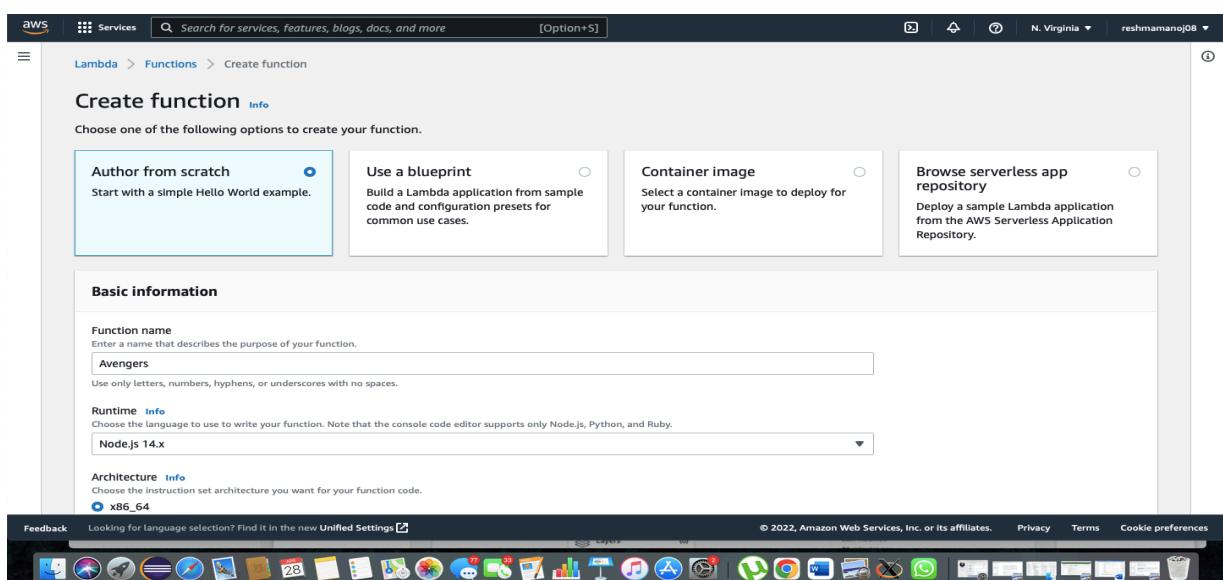
It indicates how many resources were used and how much time was billed. It's also reasonably priced.

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.

The screenshot shows the AWS Lambda Functions page. At the top, there is a search bar and a navigation bar with options like 'Actions' and 'Create function'. Below the search bar, a table titled 'Functions (1)' lists the single function 'Avengers'. The table columns include 'Function name', 'Description', 'Package type', 'Runtime', 'Code size', and 'Last modified'. The 'Avengers' function is listed with a description of '-', package type 'Zip', runtime 'Node.js 14.x', code size '957.0 byte', and last modified '6 hours ago'. At the bottom of the page, there are links for 'Feedback', 'Unified Settings', 'Privacy', 'Terms', and 'Cookie preferences'.

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

The screenshot shows the AWS Lambda Code source editor. The top navigation bar includes 'Services', a search bar, and a 'Code' tab which is currently selected. On the left, there's a sidebar with 'Layers (0)', 'Add trigger', and 'Add destination' buttons. The main area shows the 'Code source' editor with tabs for 'Info', 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions'. The 'Code source' tab is active, showing the 'index.js' file content. The code is as follows:

```
1 var json = { "service": "lambda", "reference": "https://aws.amazon.com/lambda/avengers/", "questions": [ { "q": "What is the real name of the Scarlet Witch?", "a": "Wanda Maximoff" }, { "q": "Which film did the Aether first appear in?", "a": "Thor: The Dark World" }, { "q": "Which of the Infinity stones is hidden on Vormir?", "a": "Soul Stone" }, { "q": "What is Captain America's shield made of?", "a": "Vibranium" }, { "q": "Which country is Black Panther next in line to be king of?", "a": "Umuaro" } ] }
```

To the right of the code editor, there's a 'Test your function' panel with instructions about event triggers and invoke options. Below the code editor, there are 'Upload from' and 'Deploy' buttons, and a 'Changes not deployed' status indicator. At the bottom, there are links for 'Learn more', 'Programming Model', 'Invoking Lambda functions', and 'Testing functions'. The bottom of the screen shows the macOS dock with various application icons.

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.

The screenshot shows the AWS Lambda console interface. The top navigation bar includes the AWS logo, a search bar, and account information for 'N. Virginia' and 'reshmamanoj08'. Below the navigation, there are tabs for 'Code', 'Test' (which is selected), 'Monitor', 'Configuration', 'Aliases', and 'Versions'. The main content area is titled 'Test event'. It contains a note: 'To invoke your function without saving an event, modify the event, then choose Test. Lambda uses the modified event to invoke your function, but does not overwrite the original event until you choose Save changes.' There are two options for 'Test event action': 'Create new event' (radio button) and 'Edit saved event' (radio button, which is selected). The 'Event name' field is set to 'CloudComp_Proj'. Below this is a 'Event JSON' section containing a code editor with the following JSON:

```
1- {  
2-   "key1": "value1",  
3-   "key2": "value2",  
4-   "key3": "value3"  
5- }
```

At the bottom of the page, there are links for 'Feedback', 'Unified Settings', and copyright information: '© 2022, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

This screenshot shows the same AWS Lambda interface after saving the test event. A green notification bar at the top left says: 'The test event CloudComp_Proj was successfully saved.' The 'Test' tab is still selected. The 'Execution result: succeeded (logs)' section is expanded, showing a link to 'Details'. The rest of the page is identical to the first screenshot, including the 'Test event' configuration and the right-hand sidebar with Lambda documentation.

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**.

Trigger configuration

API Gateway

Add an API to your Lambda function to create an HTTP endpoint that invokes your function. API Gateway supports two types of RESTful APIs: HTTP APIs and REST APIs. [Learn more](#)

Create a new API or attach an existing one.

API type

HTTP API
Create an HTTP API.

REST API
Create a REST API.

Security

Configure the security mechanism for your API endpoint.

Additional settings

Lambda will add the necessary permissions for Amazon API Gateway to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

Feedback Looking for language selection? Find it in the new [Unified Settings](#)

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

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

Code Test Monitor Configuration Aliases Versions

Triggers

Triggers (1)

Find triggers

Trigger

API Gateway: Avengers-API

arn:aws:execute-api:us-east-1:557992640698:d5smmm8680c/*/*/Avengers

API endpoint: <https://d5smmm8680c.execute-api.us-east-1.amazonaws.com/default/Avengers>

Details

API: **api-gateway/d5smmm8680c/*/*/Avengers**

API name: **Avengers-API**

API type: **rest**

Authorization: **NONE**

Enable metrics and error logging: **No**

Method: **ANY**

Resource path: **/Avengers**

Security: **NONE**

Stage: **default**

Function configuration

Use the function overview to see triggers, layers, and destinations to your function. You can see the following types of resources in the visualization:

Triggers are AWS services or resources that invoke the function.

Destinations are AWS resources that receive a record of an invocation after success or failure. You can configure Lambda to send invocation records when your function is invoked asynchronously, or if your function processes records from a stream. The contents of the invocation record and supported destination services vary by source.

Layers are resources that contain libraries, a custom runtime, or other dependencies. Create layers to separate your function code from its dependencies.

Learn more [What Is Lambda?](#)

Feedback Looking for language selection? Find it in the new [Unified Settings](#)

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

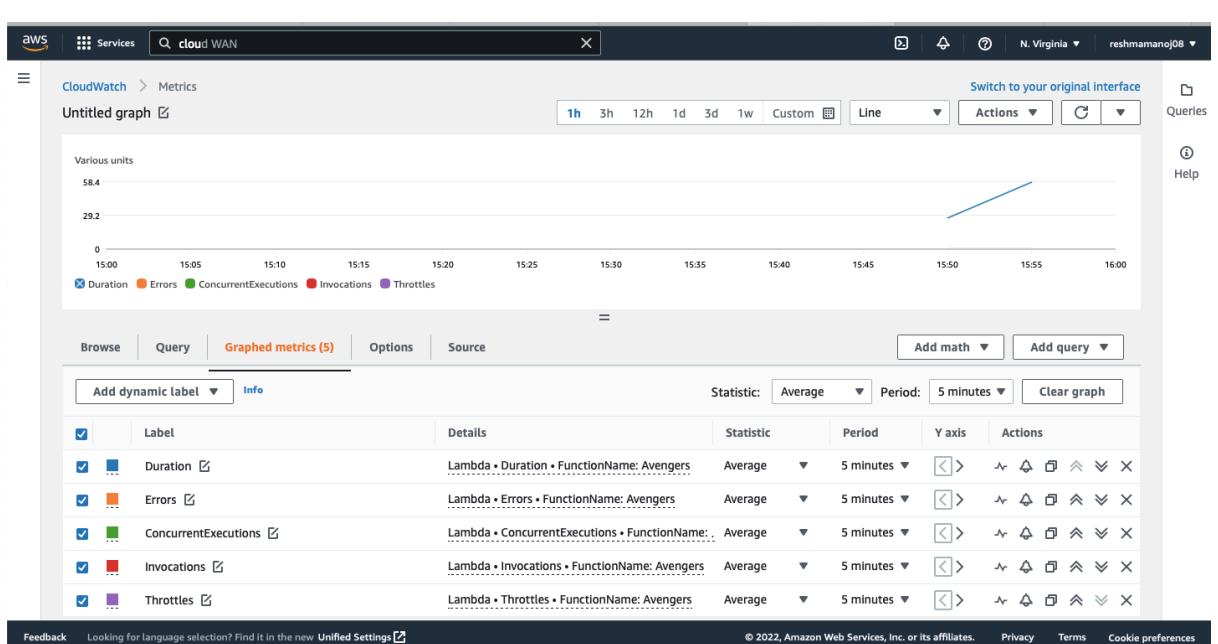
We get results. Each time we reload the browser tab with the URL, we get different results.

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

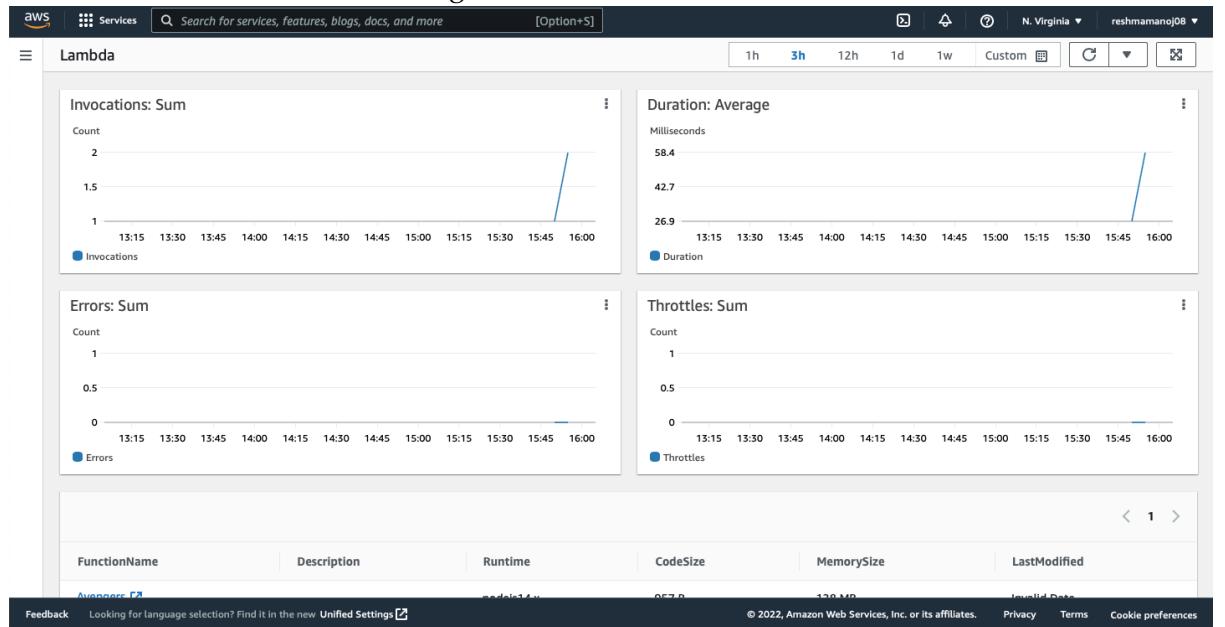
```
{"q": "What is the name of the organisation which is revealed to have infiltrated S.H.I.E.L.D. in Captain America: The Winter Soldier?", "a": "Hydra"}
```

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.

The screenshot shows the AWS CloudWatch interface. On the left, the navigation pane includes 'Logs' and 'Log groups'. The main content displays the log group details for '/aws/lambda/Avengers'. It shows retention set to 'Never expire', creation time as '8 minutes ago', and ARN as 'arn:aws:logs:us-east-1:557992640698:log-group:/aws/lambda/Avengers:*'. Below this, the 'Log streams' tab is selected, showing one log stream entry: '2022/04/28/[\$LATEST]a29f4199f2f14e71b614e19615791222' with a last event time of '2022-04-28 11:52:35 (UTC-04:00)'.



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.

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

The screenshot shows the Salesforce Trailhead Getting Started page. The sidebar on the left lists 'Administrator' and 'Build' sections. Under 'Administrator', 'Data.com Administration' is expanded, showing 'Clean' and 'Clean Info' options. Under 'Build', 'Customize' is expanded, showing 'Campaigns' and 'Campaign Members' (Accounts As Campaign Members). The main content area features a 'Recent Items' table and a 'Quick Links' section.

Name	Type	Object
SFDC_DevConsole	Debug Level	
RESHMA MANOJ	User	
Student Information	Custom Object Definition	
Date Of Joining	Custom Field Definition	Student Information
Semester	Custom Field Definition	Student Information
CWID	Custom Field Definition	Student Information
Department	Custom Field Definition	Student Information
Student Information Layout	Page Layout	Student Information
Student Information	Custom Tab Definition	Student Information

Quick Links

- Tools**
 - App Quick Start
 - Schema Builder
 - New custom object
- Users**
 - New user
 - Add multiple users
 - Reset users' passwords
- App**
 - Manage apps
 - Manage profiles
 - Enable Chatter feeds
- Security**
 - New profile
 - New permission set
- Data**
 - Import accounts & contacts
 - Import custom objects
 - Mass delete records

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.

The screenshot shows the 'Account Custom Fields & Relationships' page. The sidebar on the left lists 'Mobile Administration' and 'Build' sections. Under 'Build', 'Customize' is expanded, showing 'Accounts'. The main content area displays a table of custom fields for the Account object.

Action	Field Label	API Name	Data Type	Indexed	Controlling Field	Modified By
Edit Del Replace	Active	Active__c	Picklist			RESHMA MANOJ 4/15/2022, 11:38 PM
Edit Del Replace	Customer Priority	CustomerPriority__c	Picklist			RESHMA MANOJ 4/15/2022, 11:38 PM
Edit Del	Number of Locations	NumberofLocations__c	Number(3, 0)			RESHMA MANOJ 4/15/2022, 11:38 PM
Edit Del Replace	SLA	SLA__c	Picklist			RESHMA MANOJ 4/15/2022, 11:38 PM
Edit Del	SLA Expiration Date	SLAEExpirationDate__c	Date			RESHMA MANOJ 4/15/2022, 11:38 PM
Edit Del	SLA Serial Number	SLASerialNumber__c	Text(10)			RESHMA MANOJ 4/15/2022, 11:38 PM
Edit Del Replace	Upsell Opportunity	UpsellOpportunity__c	Picklist			RESHMA MANOJ 4/15/2022, 11:38 PM

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

The screenshot shows the 'New Custom Field' wizard on the 'Step 1. Choose the field type' screen. In the 'Data Type' section, 'Checkbox' is selected. Other options like 'Currency', 'Date', etc., are listed with their descriptions. The interface includes a sidebar with 'Lightning Experience Transition Assistant' and a main content area with a search bar and help links.

Step 2 – Field label : Field Update

The screenshot shows the 'New Custom Field' wizard on the 'Step 2. Enter the details' screen. The 'Field Label' is set to 'Field Update', 'Default Value' is set to 'Unchecked', and 'Field Name' is set to 'Field_Update'. A checkbox for 'Auto add to custom report type' is checked. The interface includes a sidebar with 'Lightning Experience Transition Assistant' and a main content area with a search bar and help links.

Step 3 - Default Value: Unchecked ;

Field Name: Field_Update (This will be populated automatically)

Security : Visible

The screenshot shows the "New Custom Field" setup page in the Lightning Experience Transition Assistant. It's Step 3 of 4. The field details are: Field Label: "Field Update", Data Type: "Checkbox", Field Name: "Field_Update", and Description: blank. Below this, a table lists profiles with checkboxes for "Visible" and "Read-Only". All checkboxes are checked for every profile listed.

Field-Level Security for Profile	Visible	Read-Only
Analytics Cloud Integration User	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Analytics Cloud Security User	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Authenticated Website	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Authenticated Website	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Contract Manager	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Cross Org Data Proxy User	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Custom: Marketing Profile	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Custom: Sales Profile	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Custom: Support Profile	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Customer Community Login User	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Customer Community Plus Login User	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Customer Community Plus User	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Customer Community User	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Customer Portal Manager Custom	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Customer Portal Manager Standard	<input checked="" type="checkbox"/>	<input type="checkbox"/>

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

The screenshot shows the "New Custom Field" setup page in the Lightning Experience Transition Assistant. It's Step 4 of 4. The field details are: Field Label: "Field Update", Data Type: "Checkbox", Field Name: "Field_Update", and Description: blank. Below this, a table lists page layouts with checkboxes for "Add Field" and "Page Layout Name". All checkboxes are checked for every layout listed.

Add Field	Page Layout Name
<input checked="" type="checkbox"/>	Account (Marketing) Layout
<input checked="" type="checkbox"/>	Account (Sales) Layout
<input checked="" type="checkbox"/>	Account (Support) Layout
<input checked="" type="checkbox"/>	Account Layout

d) Click on **Save**

Quick Find / Search...

Expand All | Collapse All

Lightning Experience Transition Assistant

Move to the new, more productive Salesforce.

Get Started

Salesforce Mobile Quick Start

Home

Administrator

- Release Updates
- Manage Users
- Manage Apps
- Manage Territories
- Company Profile
- Data Classification
- Privacy Center
- Security Controls
- Domain Management
- Communication Templates
- Translation Workbench
- Data Management
- Mobile Administration
- Desktop Administration
- Outlook Integration and Sync
- Gmail Integration and Sync
- Email Administration
- Google Apps
- Analytics
- Data.com Administration

Account Custom Field Field Update

Field Update

[Back to Account Fields](#)

[Validation Rules](#)

Custom Field Definition Detail

[Edit](#) [Set Field-Level Security](#) [View Field Accessibility](#) [Where is this used?](#)

Field Information

Field Label	Field Update	Object Name	Account
Field Name	Field_Update	Data Type	Checkbox
API Name	Field_Update__c		
Description			
Help Text			
Data Owner			
Field Usage			

Data Sensitivity Level

Compliance Categorization

Created By RESHMA MANOJ, 4/28/2022, 9:35 AM Modified By RESHMA MANOJ, 4/28/2022, 9:35 AM

General Options

Default Value Unchecked

Field Dependencies

New No dependencies defined. [Field Dependencies Help](#)

Validation Rules

New No validation rules defined. [Validation Rules Help](#)

[Back To Top](#)

Always show me more records per related list

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

salesforce 22

Search... Search

[Switch to Lightning Experience](#) RESHMA MANOJ [Setup](#) [Help](#) [Content](#)

It's Better in Lightning
Move to Lightning Experience and give your users a productivity boost. [Tell Me More](#) [Check Readiness](#)

Getting Started

Recent Items

Name	Type	Object
RESHMA MANOJ	User	
Field Update	Custom Field Definition	Account
updateAccount	Apex Trigger	Contact
SFDC_DevConsole	Debug Level	
Student Information	Custom Object Definition	
Date Of Joining	Custom Field Definition	Student Information
Semester	Custom Field Definition	Student Information
CWID	Custom Field Definition	Student Information
Department	Custom Field Definition	Student Information
Student Information Layout	Page Layout	Student Information

[Add App](#) [Get Started](#)

TRAILHEAD
the fun way to learn Salesforce [VIEW FREE TUTORIALS >](#)

Browse Docs Explore documentation and API reference guides.

Ask Questions Get help when you need it from Salesforce & community experts.

[Logout](#) [OpenChatter](#) [Help](#) [Feedback](#)

2) Creating Apex trigger on Contact Object :

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

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

The screenshot shows the Salesforce Setup interface. On the left, there's a sidebar with sections like 'Administrator' (Data.com Administration, Clean), 'Build' (Customize, Accounts, Contacts, Triggers), and 'Recent Items' (beta). The 'Recent Items' table lists various objects:

Name	Type	Object
RESHMA MANOJ	User	
Field Update	Custom Field Definition	Account
updateAccount	Apex Trigger	Contact
SFDC_DevConsole	Debug Level	
Student Information	Custom Object Definition	
Date Of Joining	Custom Field Definition	Student Information
Semester	Custom Field Definition	Student Information
CWID	Custom Field Definition	Student Information
Department	Custom Field Definition	Student Information
Student Information Layout	Page Layout	Student Information

On the right, there are 'Build App' and 'Salesforce Lightning' cards with 'Add App' and 'Get Started' buttons respectively. A message at the bottom says: "Turn on the power, speed, and simplicity of the new Salesforce user experience".

It will navigate to contact triggers.

The screenshot shows the Salesforce Lightning Experience interface. At the top, it says "It's Better in Lightning" and "Move to Lightning Experience and give your users a productivity boost". The main area is titled "Contact Triggers" and contains a message: "Define the Apex triggers for Contacts here." Below this is a "Triggers" section with a "New" button and a message: "No triggers defined". The left sidebar includes links for "Lightning Experience Transition Assistant", "Salesforce Mobile Quick Start", "Home", and "Administrator" (Manage Users, Manage Apps, Manage Territories, Company Profile, Data Classification, Privacy Center, Security Controls).

- 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> ();

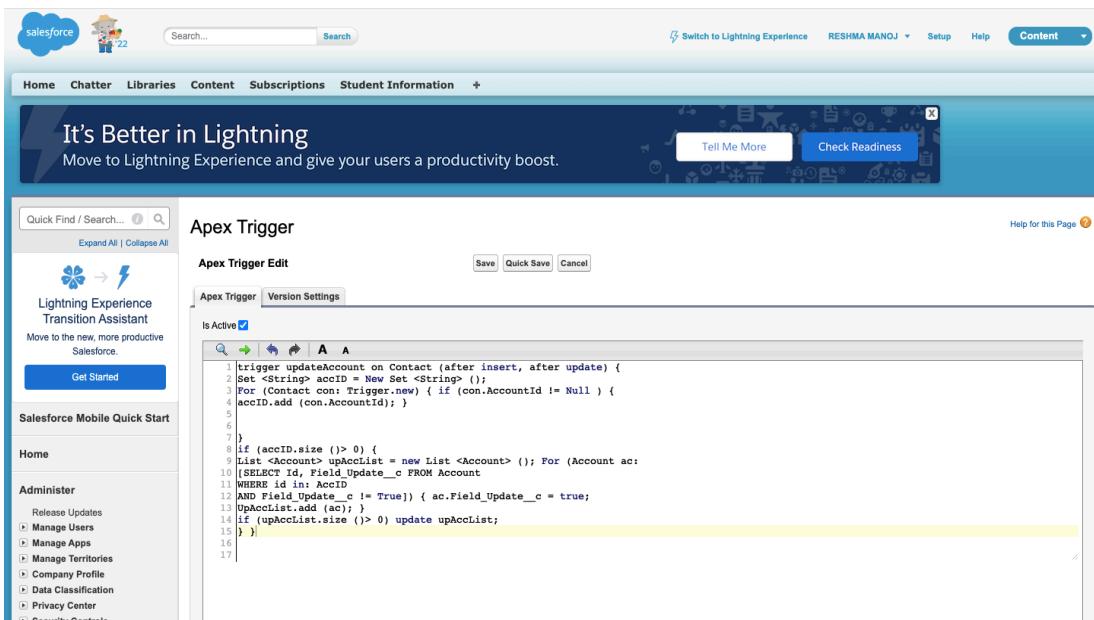
    For (Contact con: Trigger.new) { if (con.AccountId != Null ) {
        accID.add (con.AccountId); }

    }
    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*.

The screenshot shows two views of the Salesforce interface. The top view displays the 'updateAccount' Apex Trigger details, including its code:

```

1trigger updateAccount on Contact (after insert, after update) {
2Set <String> accID = New Set <String>();
3For i(Contact con: Trigger.new) { If (con.AccountId != Null) {
4accID.add (con.AccountId);
5}
6}
7}
8If (accID.size ()> 0) {
9List <Account> upAccList = new List <Account> ();
10SELECT Id, Field_Update__c FROM Account
11WHERE id in: AccID
12FOR Field_Update__c != True) { ac.Field_Update__c = true;
13upAccList.add (ac); }
14If (upAccList.size ()> 0) update upAccList;
15}

```

The bottom view shows the 'Contact Triggers' list, which contains one entry:

Action	Name	Api Version	Status	Size Without Comments	Last Modified By
Edit Del	updateAccount	54.0	Active	484	RESHMA MANOJ 4/28/2022, 10:19 AM

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 “+”

The screenshot shows the 'All Tabs' view in Salesforce. On the left, there's a sidebar with 'Recent Items' containing three entries: 'RESHMA MANOJ', 'RESHMA MANOJ', and 'RESHMA MANOJ'. Below that is a 'Recycle Bin' section. The main area is titled 'All Tabs' and contains a grid of links to various Salesforce objects. The objects listed include:

- Accounts**
- Alternative Payment Methods**
- Analytics**
- App Launcher**
- Asset Actions**
- Asset Action Sources**
- Assets**
- Asset State Periods**
- Sync Operation Logs**
- Authorization Form**
- Authorization Form Consent**
- Authorization Form Data Use**
- Authorization Form Text**
- Business Brands**
- Campaigns** Tell me more!
- Card Payment Methods**
- Individuals**
- Invoices**
- Leads**
- Legal Entities**
- Libraries**
- List Emails**
- Location Groups**
- Locations**
- Macros**
- Operating Hours**
- Opportunities**
- Orders**
- Party Consent**
- Payment Authorization Adjustments**
- Payment Authorizations**
- Payment Gateway Logs**

- 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.

The screenshot shows the 'Account Detail' page for an account named 'RESHMA MANOJ'. The page displays various account information such as:

- Rating:** Warm
- Phone:** (210) 484-5449
- Fax:** (210) 484-5449
- Website:** www.reshma-manoj.com
- Ticker Symbol:** RMANOJ
- Ownership:** Public
- Employees:** 100
- SIC Code:** 3351
- Billing Address:** 123 Main St, San Francisco, CA 94103
- Customer Priority:** Standard
- SLA Expiration Date:** 2022-07-15
- Number of Locations:** 1
- Active:** Yes
- Created By:** RESHMA MANOJ (4/17/2022, 8:47 PM)
- Last Modified By:** RESHMA MANOJ (4/17/2022, 8:47 PM)
- Description:** This is a test account.
- Custom Links:** Billing

Below the main details, there are tabs for other objects:

- Contacts:** No records to display
- Opportunities:** No records to display
- Cases:** No records to display
- Open Activities:** No records to display

- b) Click on **New Contact** and update the “**Your Contact Information**” and Save.

The screenshot shows a Salesforce Contact record for 'RESHMA MANOJ'. The contact details include:

- Contact Owner:** RESHMA MANOJ (Change)
- Name:** RESHMA MANOJ
- Account Name:** RESHMA MANOJ
- Title:** RESHMA MANOJ
- Department:** Sales
- Birthdate:** 1990-01-01
- Reports To:** [View Org Chart]
- Lead Source:** Online Form
- Mailing Address:** 123 Main St, Anytown, USA
- Languages:** English
- Created By:** RESHMA MANOJ, 4/28/2022, 10:47 AM
- Description:** None
- Phone:** (210) 484-5449
- Home Phone:** None
- Mobile:** None
- Other Phone:** None
- Fax:** None
- Email:** None
- Assistant:** None
- Asst. Phone:** None
- Other Address:** None
- Level:** None
- Last Modified By:** RESHMA MANOJ, 4/28/2022, 10:47 AM

Below the contact details, there are sections for Opportunities and Cases, both showing 'No records to display'.

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**.

The screenshot shows a Salesforce Account record for 'RESHMA MANOJ'. The account details include:

- Account Owner:** RESHMA MANOJ (Change)
- Account Name:** RESHMA MANOJ [View Hierarchy]
- Parent Account:** None
- Account Number:** 343546567
- Account Site:** Stevens
- Type:** Prospect
- Industry:** Engineering
- Annual Revenue:** \$1M - \$5M
- Field Update:** ✓ (checked)
- Billing Address:** 123 Main St, Anytown, USA
- Customer Priority:** Standard
- SLA Expiration Date:** 2023-12-31
- Number of Locations:** 1
- Active:** Yes
- Created By:** RESHMA MANOJ, 4/17/2022, 8:47 PM
- Description:** None
- Custom Links:** Billing
- Rating:** Warm
- Phone:** (210) 484-5449
- Fax:** None
- Website:** None
- Ticker Symbol:** None
- Ownership:** Public
- Employees:** 10-49
- SIC Code:** None
- Shipping Address:** SLA
- SLA Serial Number:** None
- Upsell Opportunity:** None

Below the account details, there is a section for Contacts, showing a new contact record for 'RESHMA MANOJ'.