Course Project

CS 524 Introduction to Cloud Computing

Instructions:

a) This project has two parts, Part A (to be completed on AWS) and Part B (to be completed on Salesforce.com), each worth 50 points—with weight of Final.
b) Please use your AWS Academy account to complete Part A; and your Salesforce Developer Account, to complete Part B.
c) Please make sure that you provide the appropriate screenshots and explanations wherever those are required.

PART A: AWS [50 Points]
Creating a simple microservices architecture.

The microservice will return a JSON object containing a random question and answer pair (about Avengers) using an **Amazon API Gateway** endpoint that would invoke an **AWS Lambda** function.

As before, remember to double check the **Amazon SLAs** and ensure that you take all the necessary steps not to exceed the resource.

First, you need to ensure that you understand the concept of **Microservices Architecture**, **Application Programming Interface (API)**, **Restful API**. This has already been covered in the lecture.

The description is accompanied by Figure 1 below.

The **HTTP** Response will consist of questions and answers based on the popular *Marvel Avengers* series. Make sure you understand the purpose before proceeding. All steps should be well-documented with screenshots, according to your unique AWS account id.

After you complete the lab, make sure to include the conclusion of your findings. The outline of your work is as follows:
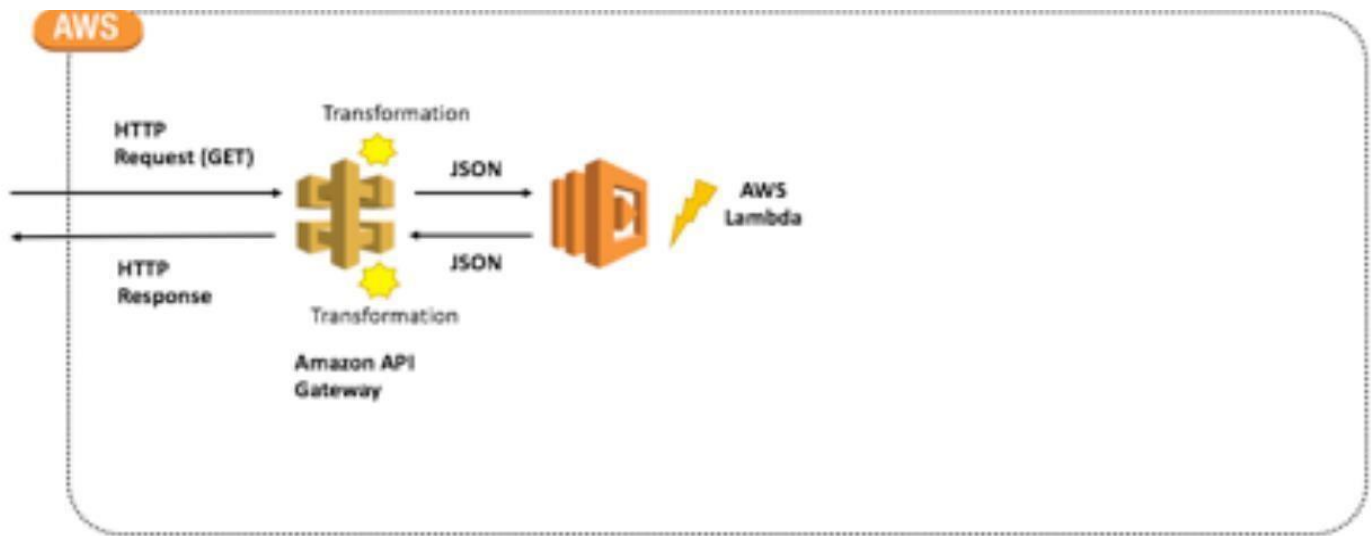
Figure 1

1) Create a **Lambda** function named "**Avengers**" (a Lambda function is a serverless compute service which helps you run your code without requiring you to manage your server.)

2) Copy and paste the following code in your Code source window:

```
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":"Wakanda" },{
"q":"What is the real name of Black Widow?",
"a":"Natasha Romanoff" },{
"q":"What is the name of the axe created for and then used by Thor in Avengers: Infinity War?",
"a":"Stormbreaker" },{
"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" },{
```

```
"q":"What nickname does Captain America know the Winter Solider as?",  "a":"Bucky"
},{
"q":"What food do the Avengers go to eat after the Battle of New York in the first Avengers film at Tony
Stark's suggestion?",
"a":"Shawarma" },{
"q":"What type of radiation caused Bruce Banner to become the Hulk?",
"a":"Gamma rays" },{
"q":"What is the name of the treaty the Avengers are asked to sign which divide the Avengers, bringing us
to Captain America: Civil War?",
"a":"The Sokovia Accords" },{
"q":"Who has directed the most MCU movies?",
"a":"The Russo Brothers" }
] }
exports.handler = function(event, context) {
var rand = Math.floor(Math.random() * json.questions.length); console.log("Quote selected: ", rand);  var
response = {
body: JSON.stringify(json.questions[rand])
};
console.log(response); context.succeed(response);
};
```

2) After deploying that function, add an API Gateway as the trigger to your Lambda service, which shoulb
   be implemented as **REST API**.

3) Test your **Lambda** function by copying your API Gateway endpoint URL and pasting it into your
   browser.

4) Invoke the Lambda function and view **CloudWatch** logs for this function.

5) Write the report with the conclusion.

PART B: Salesforce.com [50 Points]
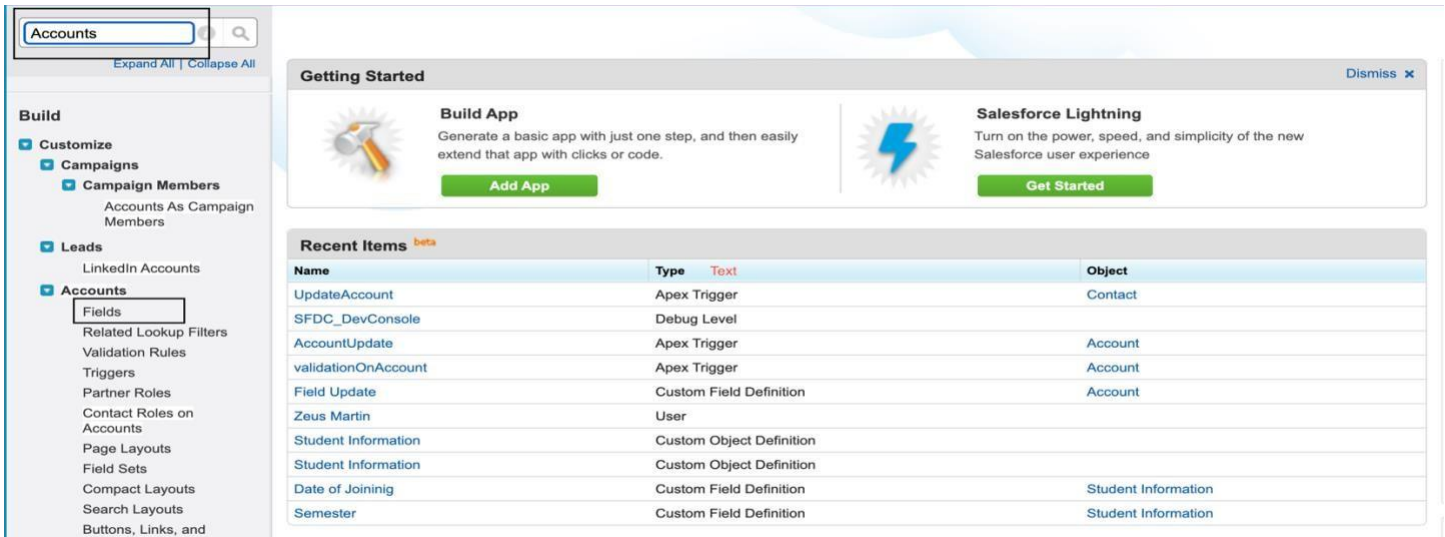**Customization on the Standard Object**

In this part you 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**.

Steps for accomplishing this task are as follows:

1) **Creating custom field on Account Object [15]:**
    Now, we will creating custom field name **Field Update** on the **Account** object. Steps are as follows:

a) Go to **Quick Find/ Search** box (You will find it on the left side) and Search Accounts and click on Fields (You will find it under Accounts)

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

d) Click on **Save**

2) **Creating Apex trigger on Contact Object [25]:**
Now we will be creating Apex trigger on the Contact Object. Steps for are as follows:

a) Go to **Quick Find/ Search** box (You will find it on the left side) and Search **Contacts** and Click on **Triggers** (You will find it under Contacts)
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;
        }
    }
```

Name	updateAccount	sObject Type	Contact
Code Coverage	0% (0/14)	Status	Active
Created By	Zeus Martin, 4/9/2022, 12:54 PM	Last Modified By	Zeus Martin, 4/9/2022, 4:54 PM
Namespace Prefix			

Apex Trigger | Version Settings | Trace Flags

```
1 trigger updateAccount on Contact (after insert, after update) {
2    Set <String> accID = New Set <String> ();
3    For (Contact con: trigger.new) {
4        if (con.AccountId != Null ) {
5        accID.add (con.AccountId);
6        }
7    }
8    if (accID.size ()> 0) {
9        List <Account> upAccList = new List <Account> ();
10       For (Account ac:
11               [SELECT Id, Field_Update__c
12               FROM Account
13               WHERE id in: AccID
14               AND Field_Update__c != True]) {
15       ac.Field_Update__c = true;
16       UpAccList.add (ac);
17       }
18       if (upAccList.size ()> 0)
19           update upAccList;
20   }
21 }
```

c) Click **Save** (Make sure your trigger status is **Active**)

3) Creating a Contact record [10]:
Now we will be creating Contact records from the Account object. Steps are as follows:

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

Home | Accounts | Cases | Campaigns | Student Information | +

All Tabs

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

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

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

END