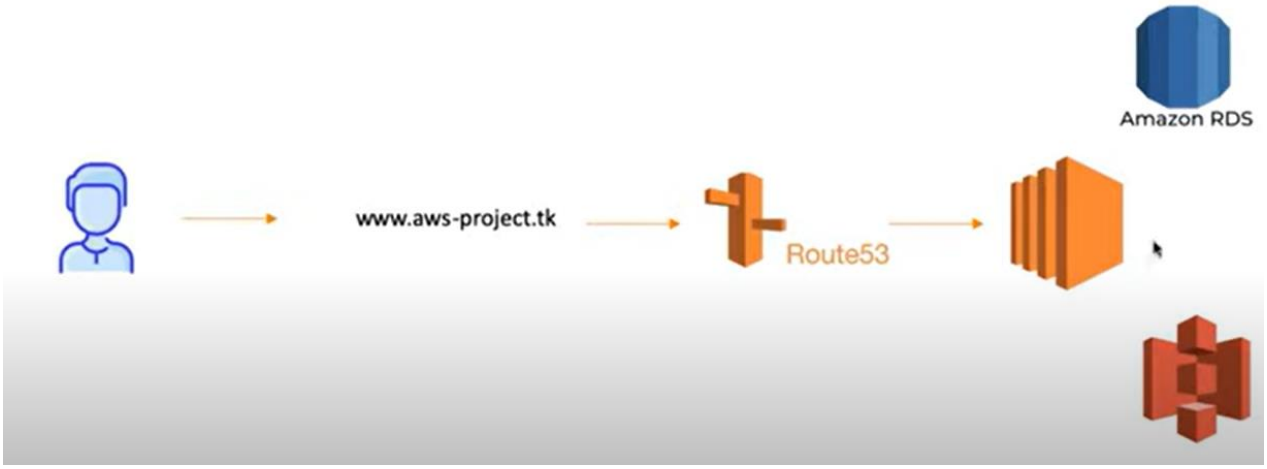
Project 3: Hosting an end-to-end website on AWS system

Components used:

1. EC2
2. Amazon RDS(MySQL Engine)
3. S3
4. Route 53
5. IAM

Summary: We will be creating employee website to store their data along with a photo ID. We will be deploying an EC2 instance, a S3 bucket and a MySQL DB in this case. When the request will be made by the user the request will pass to the EC2 instance which should be able to talk to S3 for uploading the images and at the same time it will communicate to amazon RDS that would store the data entry made to this website. We will map the domain that is brought from Freenom to route 53, the route 53 will then be pointing to the EC2 instance, so that we can access this website from the domain URL.

Overview/diagram:



Steps:

1)Deploy Amazon RDS:

- Navigate to RDS service in AWS
- Hit Create a database
- Select MySQL as engine option, the version can be left to default
- Since this is a test project, we select the free tier template

- Set DB instance setting = “Employee”, master username = “admin”, master password = “password01”, confirm the password
- Choose the DB instance call as t2 micro
- Autoscaling can be left unchecked
- VPC can be left to default
- Publicly accessible = “Yes”
- Hit on create DB, in sometime DB should be up and running

2)Create a S3 bucket for storing data:

- Navigate to S3 service, create a bucket named “addemployee”, region = “ohio”

3)Create a EC2 server:

- Navigate to EC2, select launch instance
- Select the default settings for this server, like VPC, subnet, auto enable IP, storage and security group
- After you have validated all the settings are correct, launch the EC2 instance
- Name this instance per your choice
- Connect to this server using putty
- We need to first connect to the MySQL RDS to see everything is working fine
- Run the following commands to connect to RDS :
    1. Sudo su
    2. Yum install -y mysql
    3. Mysql -h <rds endpoint> -u <admin name> -p <press enter>
    4. Enter your password and hit enter
    5. We should be in MySql RDS
- Once we are in RDS , run the command “show database”, no database created automatically
- Run “create database employee”
- Based on the code that we have used, we will create five tables in this “employee” database namely empid, fname, lname, skill, location
- Query used: use Employee

```
Create table emp
empid(varchar 20),
fname(varchar 20),
lname(varchar 20),
skill(varchar 20),
location(varchar 20)
```

- Validate the above table by running “show tables”
- The code will need to be pushed to the EC2 instance, we have to do it through github
- Once the code is deployed the IP of the EC2 server would pop the website up but when we try to enter the data and upload the image to the website, it will fail. The reason here is because

the EC2 is not able to talk to S3, in order to resolve this we need to create a IAM role for the EC2 instance so that it can communicate to S3

4)Create IAM role:

- Navigate to IAM, select create role
- Select the role for EC2 service, permission = 'AdministratorAccess"(only for this project), name = "ec2role", create role

5)Map the IAM role to EC2:

- Go to EC2 dashboard, select the instance
- Go to actions, select Attach IAM role, select the role = "ec2role"
- Once this is done, go back to the website with the server URL and launch the website, fill the information and the upload the image, the database and the S3 should have the data

6)Map the domain through Route 53:

- Navigate to route 53, create a hosted zone
- Name should be same as domain that is brought from Freenom
- Fill the nameserver from route 53 to freenom portal and hit save
- Hit "create hosted zone", name = blank, type = "IPv4-address" Alias = 'no', value = "IP address of the EC2 server", hit save
- Create another record set name = "www", type = "IPv4-address" Alias= 'no', value = "IP address of the EC2 server", hit save

Once this is done, the domain URL should come up with our website when launched from the web brower.