Assignment 8

New Attempt

- Due Nov 12, 2024 by 11:59pm
- Points 100
- Submitting a file upload
- File Types pdf

Due: Tuesday, November 12, 2024 (11:59 PM). Please submit your assignment on Canvas as a PDF.

Part 1: Amazon Web Services (AWS) Introduction

In the first part of the assignment, you will be introduced to Amazon Web Services, one of the most popular cloud infrastructure service providers. You will also create an AWS account and get familiar with the core components of AWS.

- Visit **https://aws.amazon.com/education/awseducate/** **(https://aws.amazon.com/education/awseducate/)** and click Join AWS Educate.
- After finishing the AWS Educate sign-up process you will be asked to link an existing AWS account or continue with the AWS Educate starter account. If you select the latter option your account will be restricted and you will not be able to practically experience different aspects of AWS. Therefore, it is preferable to own an AWS account and use the former option. You can create a free tier AWS account by visiting **https://aws.amazon.com/free/** **(https://aws.amazon.com/free/)** and clicking on Create a Free Account (your credit card will be charged approximately $2).

1.1 Amazon IAM

AWS Identity and Access Management (IAM) is a web service to securely control access to AWS resources by defining authenticated and authorized users. IAM is a *user*, *group*, *policy* and *role* management tool for AWS that manages not only your users, but also groups of users, policies for other AWS tools as well as roles.

AWS account root user has full access to all AWS services and resources in the account but it is not recommended to use the root user for daily access. The best practice is to use the root user to create an IAM user, then lock away the root user and use this account for service management tasks.

To create a new *group* visit IAM: **https://console.aws.amazon.com/iam/home?region=us-east-1#/home** **(https://console.aws.amazon.com/iam/home?region=us-east-1#/home)** > Access Management -> User Groups > Create Group > Select a name “admin” for the group > Select policy **AdministratorAccess**.

To create a new *user* click to Users > Create user > Set username to “admin” > Select the “admin” group you created earlier > Create user

Click the newly created user "admin" > Create access key > Application running on an AWS compute service > Add a description tag value > Save the access key copied to somewhere where you can get to it later > Done

Now you can safely sign out the root account and login from your newly created user account. Keep in mind that for non-root users you need the account ID which can be found at **https://console.aws.amazon.com/billing/home (https://us-east-1.console.aws.amazon.com/billing/home?region=us-east-1#/bills)**

From this point you can add as many users/roles/policies as you feel in the following tasks.

Task 1 (5%): In your opinion is it a good practice to create multiple users in AWS? Justify your answer and give examples.

Task 2 (5%): What could be a use case for an IAM *role*?

Task 3 (5%): What is the difference between an IAM *role* and an IAM *policy*?

1.2 Amazon EC2 & DynamoDB

Elastic Compute Cloud or EC2 is a virtual server used by customers to run their applications on the AWS cloud infrastructure. AWS EC2 provides instances with different resource configurations of CPU, memory, storage, and networking available in different sizes satisfying the customers' requirements.

In this set of tasks, you will learn how to initiate an AWS server. This server is essentially a Virtual Machine Managed by AWS. Go to EC2 (**https://console.aws.amazon.com/ec2/v2/home?region=us-east-1 (https://console.aws.amazon.com/ec2/v2/home?region=us-east-1)**) > Launch Instance > Select Ubuntu Server 22.04 > choose the default t2.micro Type (Free Tier) > Launch > Create a pair of keys (and save them for later use) > Launch Instance. After creating the EC2 instance you should be able to SSH into the machine with the *keys* you saved, setup a web server or even an API.

Next, you will create a database table in DynamoDB. Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. To do so, go to **https://console.aws.amazon.com/dynamodb/home?region=us-east-1# (https://console.aws.amazon.com/dynamodb/home?region=us-east-1#)** and click Create Table > enter for table name “usersTable” and Partition Key “userId” > click Create table. You can add elements in the table after selecting it and clicking Actions > Create Item > Enter your item.

You will get an error if you try to access the newly created table from the EC2 instance initiated previously. You can try accessing the table using Python or Node.js or any language of your choice. You can check the python examples **here (https://boto3.amazonaws.com/v1/documentation/api/latest/guide/dynamodb.html)** . However, it is not required that you write an active, working code that connects to the database.

Task 4 (10%): What is needed in order for the EC2 instance to be able to access the newly created DynamoDB table? Please consider following the best practices.

Task 5 (5%): Report the steps you took in order to EC2 instance access the DynamoDB table.

1.3 Running a Web App on EC2

In this section, you will learn how to create a new Web App running on EC2. We assume that you have saved the *keys* in the process of creating an EC2 instance as **aws_key.pem**. The commands provided below are for the bash terminal and it is preferred that you use Linux in order to complete them.

In the following next steps we will create a sample Node.js application that displays "Hello World".

STEP 1

You need to change permissions to the saved *keys* first. To do so, open a terminal and type:

```
chmod 400 aws_key.pem
```

STEP 2

In order to make any changes to your server you need to connect to it. To connect to the server type in a terminal

```
ssh -i aws_key.pem ubuntu@ecx-xx-xxx-xx-xxx.compute-1.amazonaws.com
```

Where ecx-xx-xxx-xx-xxx.compute-1.amazonaws.com is the public DNS address of your instance.

STEP 3

After connecting to our machine we can install any software we want in order to setup our Web App. We install Node.js since this is what we are going to use as Web server, but you can install any Web server you wish using the package manager of Ubuntu `apt`.

In order to install Node.js you need to run the following commands (in the Ubuntu instance):

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.35.1/install.sh | bash
source ~/.bashrc
nvm install v10 --lts
nvm alias default v10
```

STEP 4

In this step we create a Web App inside the Ubuntu instance. First we need to create a folder for our App.

```
mkdir webapp && cd webapp
```

Then initiate a Node.js project

```
npm init
<enter> as many times as needed
```

Install express as the main driver for our Web App

```
npm install express --save
```

Create an index.js file and paste the following:

```
const express = require("express");
const app = express();

app.get("/", (req, res, next) => res.send("Hello World!"));

app.listen(8081, () => console.log("Listening on port 8081"));
```

This is a very simple Web App that displays "Hello World!". You can change it as you wish.

In order to run it type:

```
node index.js
```

Now, you successfully created a Web App and launched it on port 8081.

STEP 5

In order to try your Web App you can visit your machine's public DNS address.

e.g. in your browser:

```
ecx-xx-xxx-xx-xxx.compute-1.amazonaws.com:8081
```

If you have completed all steps properly (including **Task 7**), then, you should be able to see "Hello World!".

Task 6 (5%): If you go to EC2 > Instances and click on the Instance you have created, then you will notice that there is a plethora of information about your newly created machine. There is an IPv4 Public IP created for your EC2 instance. If you right click and Stop the machine and then Start it again, you will realize that the IP assigned to that machine is changed. Why is that? What would you do in order to give your machine an IP Address that persists through reboots.

Task 7 (5%): If you try to setup a Web server listening to the port 8081 inside your instance you will soon realize that it is not accessible from the outside world. What is the AWS component responsible for allowing traffic to be sent to port 8081? What steps would you take in order to make it accessible from the outside world?

STEP 6

Please note that you need to delete your instance after you are done. It will keep running as long as you do not delete it. In order to do so right click the instance > Terminate (delete) instance. Always keep in mind that in Amazon Web Services you pay for what you use. So if you keep an instance alive for days you are going to pay for the usage.

Part 2: Amazon Web Services Networking

In the second part of the assignment you will get introduced to AWS networking and more specifically to VPC, Subnets, Internet Gateway, NAT Gateway, Security Groups and Network ACLs. You will create a VPC with two types of Subnet: Private and public. The public subnet will be accessible from the internet while the private subnet will not.

2.1 Amazon VPC

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you have defined. This virtual network closely resembles a traditional network you operate in your own data center, with the benefits of using the scalable infrastructure of AWS. For more information please read **here** **(https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Subnets.html)** .

In order to create a VPC you will need to head to VPC module at **https://console.aws.amazon.com/vpc/home?region=us-east-1#** **(https://console.aws.amazon.com/vpc/home?region=us-east-1#)** > Your VPCs > Create VPC > For the name tag choose a name of your own e.g: CYBERSECURITY_VPC, for the IPv4 CIDR block* select 10.0.0.0/16 > Click Create.

Task 8 (5%): What is the range of the IPs in the VPC you just created?

Task 9 (5%): What is the difference between a VPC and a Virtual Private Network (VPN)?

2.2 Subnets

AWS VPC subnets are essentially networks that operate like the subnets in regular networks. In this Task you will create private and public subnets.

Go to VPC > Subnets > Create Subnet > Fill

```
Name tag:           CYBER_SECURITY_SUBNET_PUB
VPC:                Select the VPC you previously created
Availability Zone:  us-east-1b
IPv4 CIDR block:    10.0.1.0/24
```

Click Create

For the Private network do the same but instead provide the following:

```
Name tag:           CYBER_SECURITY_SUBNET_PRIV
VPC:                Select the VPC you previously created
Availability Zone:  us-east-1c
IPv4 CIDR block:    10.0.2.0/24
```

Task 10 (5%): What are the IP ranges of the two subnets you created?

Task 11 (10%): Why would someone create a public and a private subnet. What are the uses of each of them? Provide an example.

2.3 Internet Gateways

In this section you will make the public subnet accessible from the Internet. This is done using Internet Gateways. An Internet Gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the Internet.

An Internet gateway serves two purposes: providing a target in your VPC route tables for Internet-routable traffic, and performing network address translation (NAT) for instances that have been assigned public IPv4 addresses.

In order to make the Public subnet accessible from the Internet we need to create an Internet Gateway and a Route Table that uses this Gateway for all connections to the Internet:

Go to VPC > Internet Gateways > Create internet gateway > Fill In

```
CYBERSECURITY_IGW
```

Click Create

Then right click the Internet Gateway you just created and Select Attach to VPC > Select the VPC you created earlier.

Now you need to “route” all the traffic from the internet to the public subnet. In order to do so go to VPC > Route Tables > Create Route Table > Fill In

```
Name tag: CYBERSECURITY_RTB
VPC*: Select the VPC you just created
```

Click Create

Now select the route table you just created and in the lower bottom panel select Routes > Edit Routes > Add Route > Fill in

```
Destination Target 0.0.0.0/0 Select the Internet Gateway you just created
```

Click Save Routes. In the same panel select Subnet Associations > Edit Subnet Associations > Select the Public Subnet > Save

Now any instance launched in a public subnet has data flow to and from the Internet, while anything launched inside the private subnet does not.

Task 12 (5%): If we launch two instances, one in the public subnet and one in the private subnet, the one in the private subnet will not have internet access. How is it possible to connect to the instance in the private subnet through SSH?

Task 13 (10%): We can give internet access to the private subnet by creating a NAT Gateway. What is the difference between the NAT Gateway and the Internet Gateway?

Task 14 (5%): What are the steps needed to be taken in order to create a NAT Gateway into the public subnet to provide the private subnet with internet access? You can try it by launching two instances and experimenting with the NAT Gateway.

2.4 Security Groups

In this section you will get introduced to the Security Groups concept. In AWS every instance is launched with an associated security group and any security group is associated with a VPC. Security groups are essentially whitelists that allow data to flow from the Internet to the instance inside the VPC.

In order to create a security group go to VPC > Security Groups > Create Security Group > Fill in

```
Security group name: CYBERSECURITY_SG
Description: CYBERSECURITY SECURITY GROUP
VPC: Select the VPC you created
```

Click Create > Select the Security Group you just created > Inbound Rules > Edit Rules > Add Rule > Fill in

```
Type Source SSH Anywhere
```

Click Save Rules

Now any instance that is associated with this newly created security group will have SSH access enabled if no other restriction applies.

Task 15 (10%): In VPC under Security there is another module called Network ACL. What is the difference between Network ACL and Security Groups?

Task 16 (5%): Report the steps required to create a Network ACL. How would you integrate it into the public subnet you previously created?