COS20019 – Cloud Computing Architecture

Assignment 1 - part B

Photo Album web site

Due date to ESP https://esp.swin.edu.au : 9 am Monday 20 April (Week 6)

Late submission penalty: 10% of total available marks per day

Contribution to final assessment: 20%

You must write your own PHP code for this assignment.
All code will be checked for plagiarism.

To complete this Assignment you will need to…

- Successfully passed Assignment 1A.
- Make sure you have completed ACF Labs 1 to 4.
- Know how to set up and manage a MYSQL database.
- Understand how to write PHP code that interacts with MYSQL database.
- Know how to set up and manage a Web accessible S3 bucket.

Objectives

This assignment has the following objectives:

1. Create a secure Virtual Private Cloud (VPC) with subnets, routing tables and security groups.
2. Control access to and from your VPC via an Internet Gateway.
3. Create a web site in PHP that stores meta-data information about photos uploaded to S3 in a MySQL database managed by Amazon RDS. The website should enable the user to search for and display photos using meta-data.
4. Deploy and test your PHP web site on an Apache web server running on an EC2 virtual machine instance.

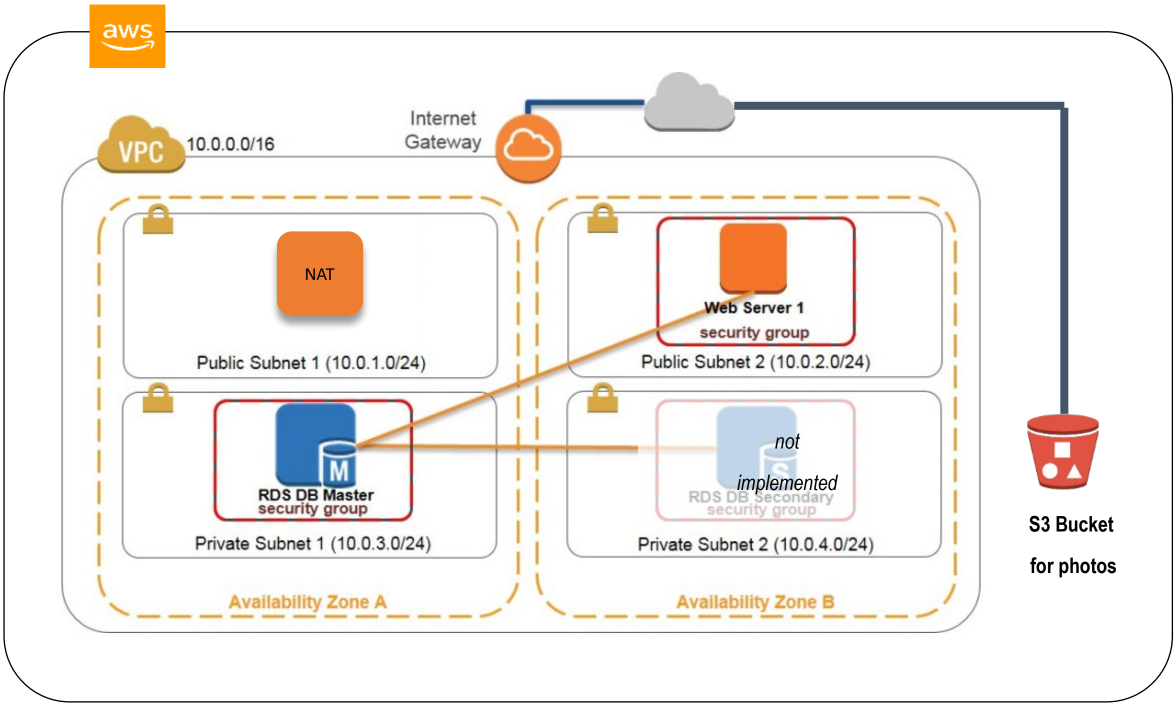
Important:
In your COS20019 assignments, all AWS resources you create (e.g. EC2 instances, Security groups, RDS database instances, etc.) should have the following additional tags added:

- StudentName (with a value of your name)
- StudentID (with a value of your id)

These tags are in addition to any other tags that are appropriate to add to the resource.
These tags will be used to assist in the assessment of your work

Infrastructure requirements

You will set up a VPC with the structure and services as illustrated in the diagram below.



Note: Do not use the default VPC. All services should be in your custom VPC.

VPC

- The name of your VPC **must** be in the format [*FirstNameInitial*][*LastName*]VPC. For example if your name is Bill Gates your VPC would be named **`BGatesVPC`**.
- Region: us-east-1
- Two availability zones each with a private and public subnet with suitable CIDR.
- Associate public subnets with a route table that routes to an Internet Gateway

Hint:

If you use the VPC Wizard to create a Public and Private subnet, the wizard will create the necessary Internet Gateway and Routing tables for you. You can then associate these routing tables with any additional subnets you create.

When you use the VPC wizard, the default NAT is an AWS NAT Gateway service. THIS IS NOT A FREE-TIER SERVICE. To avoid being charged we recommend you create a NAT instance running on a free-tier EC2 instance. Don't forget to stop your instance when you are not using it.

Security Groups

Create the following security groups

SG Name	Protocols	Source
web_tier_sg	HTTP (80), SSH (22)	Anywhere
db_tier_sg	MySQL (3306)	web_tier_sg

Virtual Machine

- Amazon Machine Image: *Amazon Linux AMI 2018.03.0 (HVM)*
- Instance type: *t2.micro*
- User data: install Appach Web server and PHP (as in Assignment 1A)

RDS Database instance

- DB engine version: *mysql 5.6.39*
- DB instance class: *db.t2.micro*
- Public accessibility: *No*
- Backup retention period: *0 days*

RDS Hint #1:

RDS requires that you have at least two Availability Zones. However, while in a production deployment it is desirable to have a master-slave Multi-AZ RDS, this is ***not*** available on as part of the AWS Free-Tier. Therefore *do not* check Multi-AZ checkbox when you set up your RDS. (The RDS wizard has a Free-tier only checkbox on the first page that limits selections to Free-tier options.)

RDS Hint #2:

In *production* you need your RDS to be in a private subnet with the only the Web tier security group being able to access it. However, your need to be able to access your database over the internet so that you can set it up and maintain it. There are a number of ways you can do this. It is up to you to choose.

1. Install phpMyAdmin on your EC2 Webserver instance and create and maintain the database through its UI. Instructions on how to do this are on the Assignment Resources section on Canvas.
2. Create a ‘Photo Admin’ interface in PHP that allows a user with the correct credentials to create, insert and delete SQL operations so the database, tables and records can be managed.
3. Create a ‘Bastion Host’ in a public subnet and enable SSH forwarding to the RDS EC2 instance (most secure method).

Functional requirements

Storage

Create a S3 bucket to store your photos. This bucket should be configured to be accessible as a web page. Manually upload 6 or more photos into S3 and record their URLs so they can be referenced in the database.

Note: In **this** assignment you **will not** be required to add functionality to the PHP web page (*upload.php*) you created in Assignment 1A so photos can be uploaded to S3 via PHP.

However, in Assignment 2 you will need to do this using the AWS PHP SDK. So feel free to add this functionality to this assignment if you wish. See the **How to …** pages on Canvas for instructions on how to do this.

Database

Before you create your database, carefully read the RDS hints on Page 3 above.

Create a database in your RDS instance with table called ***photos*** that stores meta-data about photos stored in an S3 bucket:

- photo title
- description
- date of photo
- keywords (comma delimited list is ok but a separate linked table would be preferable)
- reference to the photo object in S3.

Web pages

You can choose your own design for the Web site. Create a web page (***getphotos.php***) that allows the user to search for a photo based on title, keywords or date ranges. It will display all the photos found along with their meta-data. **For example, you should be able to search for photos with the keyword "cat" taken anytime in March this year**.

The directory structure of your website is described below. You can create additional HTML, CSS, JavaScript files if needed. The Apache HTTP server serves files located in a directory called Apache document root (/var/www/html).

```
COS20019/
        . . . AWS SDK, other support libraries
        photoalbum/
                upload.php (not yet fully implemented)
                getphotos.php
                . . . other PHP, HTML, CSS, JavaScript files
```

Testing

Upload a number of photos along with their meta-data. A number of the photos should have keywords in common. Thoroughly test the photos and their meta-data are correctly displayed based of search based on **any combination** of one or more of the following fields:

title, keywords, after date, before date.

Submission

Do NOT include:

- ***AWS libraries in the assignment zip file. These are very large. Any submission that includes AWS libraries will be rejected by ESP***
- ***Photos from your album.***

Submission is a single file to **ESP** https://esp.swin.edu.au/. Create a zip file called **assign1b.zip** containing the following:

1. Your web site source code in a directory **photoalbum** as specified above.
2. PDF document called **assign1b.pdf**. The document should include the following:
   a. Title page with your name and student id.
   b. Table of Contents and page numbers
   c. URL of your **upload.php** on your EC2 web server so your tutor can view the web site.
   d. Screenshots of your

   □ VPC page
   □ Subnets page showing the four subnets your created
   □ Private Route table page with the Routes tab setting visible
   □ Private Route table page with the Subnet Associations tab settings visible
   □ Public Route table page with the Routes tab setting visible
   □ Public Route table page with the Subnet Associations tab settings visible
   □ EC2 instances page showing public ip address and tags
   □ Web tier security group showing the inbound rules
   □ Database tier security group showing the inbound rules
   □ S3 bucket showing the files inside
   □ RDS instance details window
   □ Database stucture (e.g. a screenshot from phpMyAdmin)
   □ Database records (e.g. a screenshot from phpMyAdmin)
   □ Expanded listing of your files in the /var/www/html directory (e.g. using WinSCP).
   □ Web pages.

Note: all screenshots should include something that identifies you: e.g. a tag as below or your AWS account number on the console.

COS20019 CCA: Assignment 1b Checklist

May sure all the following are completed.

Submission Checklist

Student Name: …………………………………………….

Student Id: ……………………………………………..

Tutorial time: ………………………………………………

Date of submission: ………………….

Submit to ESP a zip file called `assign1b.zip` containing

- ☐ Web site source code any other associated files *you* have written to create your web site.
- ☐ A document file called `assign1b.pdf` as specified in the Submission section above.

Marking Scheme

Infrastructure Requirements		
VPC with 2 public and 2 private subnets	2	
Correct Public and Private Routing tables with correct subnet associations	2	
Web and DB tier Security groups correct	2	
Correct EC2 instance running in correct subnet	1	
NAT Instance or Gateway defined	1	
Database storing data on specified RDS instance type	1	
Database schema as specified	1	
S3 bucket created with at least 6 photos in it	1	
S3 configured as Web site	1	
Functional Requirements		
Search page displayed from EC2 Web server	2	
Photo loads from S3 with matching metadata from RDS	2	
Search on title	1	
Search on keywords	1	
Search on date range	1	
Combination search	1	
Deductions		
Documentation not as specified or poorly presented (up to minus 5)		
Deduction - 3rd Party code not acknowledged (up to minus 20)		
Resources not properly tagged (up to minus 5)		

Comments