

COS20019 Cloud Computing Architecture

Assignment 2

Develop a High Availability Photo Album Web Site



Due date to ESP <https://esp.swin.edu.au> midnight Saturday 15 May (Week 10)

Contribution to final assessment: 20%

The assignment needs to be completed to attain a Credit grade or above in this unit. (Late submission penalty: 5% of total available marks per day)

Due to the current situation no live demonstration will be required for this assignment. Your tutor will contact you if anything needs clarification. Please carefully read the submission section.

You are required to use AWS Educate Classroom (Cloud Computing Architecture) via Vocareum to complete this assignment so your work would be accessible to your tutor through the Vocareum classroom. You need to let the resources running until you get your mark for this assignment.

To complete this assignment, you will need to have ...

- Successfully completed Assignment 1.
- Completed all Labs week (1 – 7).
- Explored how to write a PHP webpage to upload and search files.

Objectives

This assignment will extend/modify the infrastructure and program you developed in Assignment 1. It has the following additional objectives:

1. Create IAM roles to enable an EC2 instance to access S3.
2. Create a custom AMI with User data.
3. Create a Launch Configuration based on your custom AMI.
4. Create an Autoscaling Group across two Availability Zones with rules for scaling up and down.
5. Create a Load Balancer to distribute service requests.
6. Create a cached distribution of your website using CloudFront.
7. Document the deployment of your website with CloudFormation.

AWS Accounts for Web Site deployment

You have to use the following account to complete this assignment:

1. **Vocareum account:** A Vocareum-managed AWS environment has been created for you by your convenor to complete this assignment. The class will be named **Cloud Computing Architecture**. This will allow your tutor to gain access to your AWS console so your work can be marked. This class gives you US\$100 AWS credit. This account is deleted at the end of semester. If you run out of your credit you can request an extra \$25 through AWS educate website. Your tutors can not request that on your behalf.

Important:

In your COS20019 assignments, all AWS resources you create (e.g. VPC, 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

Getting Help

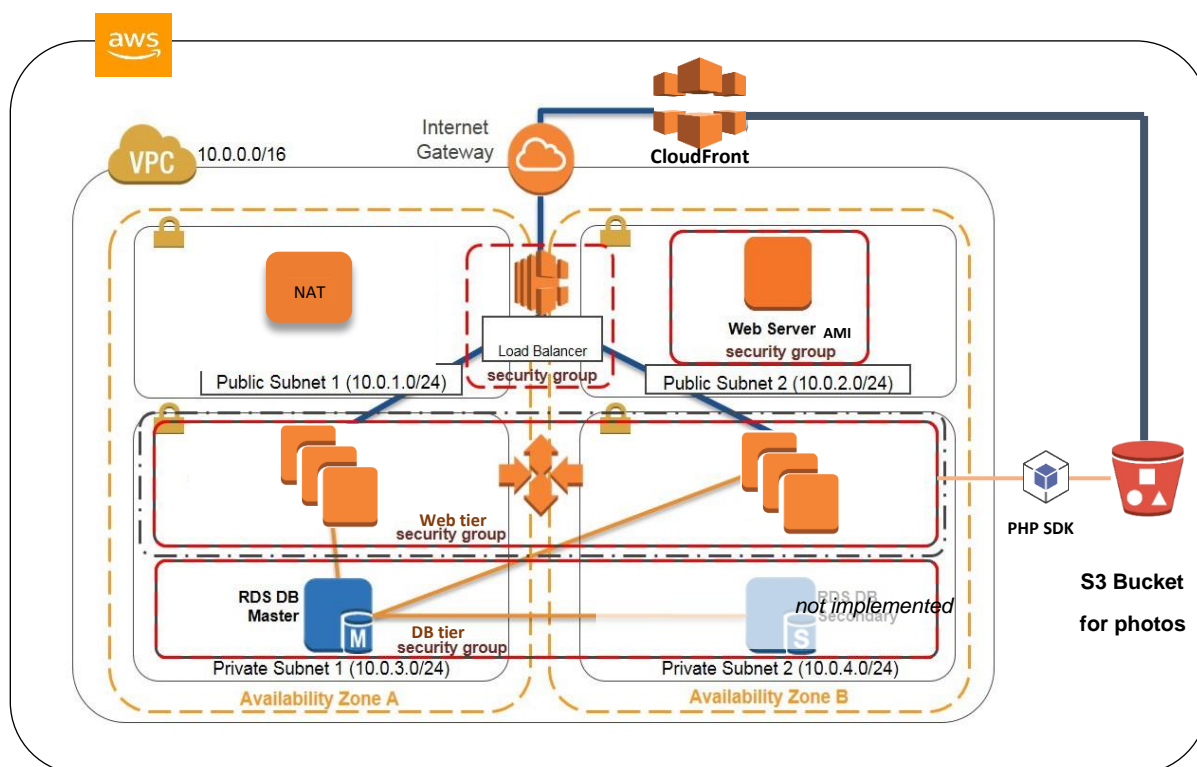
To get help on the assignment **do not email** the convenor or your tutor. An Assignment 2 Discussion board has been created on Canvas.

Students are encouraged to *post* both **questions and answers** to the discussion board. Appropriate help can include identifying errors, suggesting solutions and pointing to appropriate online resources. However, **do not** post multi-line code solutions to the board. You can attach the code to your post. Tutors will also monitor the board and post responses as appropriate. **One Q&A session is offered to help you with this assignment in lab 9** (Feel free to attend any tutorial sessions).

Requirements

Infrastructure

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



The infrastructure you need to develop is illustrated in the diagram above. The following points should be noted:

- The VPC is as per Assignment 1B. Remember, 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
- The AMI you create will be based on *Amazon Linux 2 AMI (HVM), SSD Volume Type*
- The RDS instance is the same as Assignment 1B.
- The auto-scaling group should launch instances into the private subnets of the two AZs.
- Web servers should only accept incoming from the Elastic Load Balancer.
- Web servers need to be given permission to read and write to the S3 bucket.
- The security groups should be as follows:

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

Web pages and Photo Storage

You can reuse the S3 bucket you created in Assignment 1 to store your photos. In this assignment, you will 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. You will need to:

1. Create an IAM role that gives permission to your EC2 web server instance to *write* photos to your S3 bucket.
2. Temporarily upload the photo to your web server then write it to S3 using the AWS PHP SDK.
3. Restrict access to your S3 bucket so that only web pages served from your PHP application will be able to *read* photos. Make the bucket private so that users can't use a direct URL to the S3 bucket to access an image there.

There are many resources on the web that explain how to upload a file from a web form to a server using PHP. The Assignment Resources in Canvas contains instructions on how to set up the appropriate policies on S3 and how to use the AWS PHP SDK.

Please note: your code and the way you implement it is flexible. You do not have to use PHP SDK.

Caching

Images read from S3 should be cached in a CloudFront distribution that is available across all AWS edge locations. The CloudFront URL Images should be cached for 30 minutes.

More information and instructions on CloudFront can be found at:

<http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html>
<http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/distribution-webcreating.html>

Database

Same as Assignment 1.

While the database is a private subnet, you should be able to manage it via utility such as phpMyAdmin or via a bastion host.

Load balancing

Web request load needs to be distributed across the web servers in the auto-scaling group. Ensure that your ELB is running health checks on all instances. You can use either a 'Classic' or 'Application' load balancer. **Scaling**

You need to define a scaling policy for your auto-scaling group with at least the following rules:

- The minimum number of servers is to be two.
- The maximum number of servers is 3.
- Servers should be spread as evenly as possible across the two availability zones.
- Web servers should not go above 75% CPU utilization.
- If utilization drops below 25% the number of servers should gradually reduce.

Testing

Using your Web application, upload a number of photos (at least 6) along with their meta-data. A number of the photos should have keywords in common. Thoroughly test the photos and their metadata are correctly displayed based on search for title, keywords and/or dates.

The additional tests should include:

- Terminating servers and checking to see if replacement EC2 instances are automatically deployed.
- Restart a stopped web server and see if the number of server automatically scale back
- Test access to your S3 photos is not accessible from another browser.

- To test if CloudFront is working as expected, follow [this instruction](https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/distribution-webtesting.html):
<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/distribution-webtesting.html>

Reflecting on Your Learning

Finally, complete the following task. Reflect on the overarching process above, the tasks completed, and the results delivered.

- Did you face any challenges? Explain them – and how you overcame them
- In terms of your learning, highlight 3 key takeaway points.

Submission

Do NOT include AWS libraries or photos in the assignment zip file. These are very large. Large files will be automatically rejected.

Submission is a single file to ESP <https://esp.swin.edu.au>. Create a zip file called **assign2.zip** containing the following:

1. A document **assign2.pdf** containing the following information
 - a. Your name and student id
 - b. URL of your Website ELB so your tutor can view the web site
 - c. A screenshot of the data records in your database. This could either be from the MYSQL> prompt, or from phpMyAdmin. This data will be used by the marker to check the functionality of the website.
 - d. If you have done the assignment in your own Free-tier account, provide a username and password for an Admin user (read-only permissions) to you AWS account.
2. Your website source code in a directory **photoalbum** as specified above.
3. **Optional: A CloudFormation JSON template** of your AWS resource stack Instructions for doing this are on Canvas under Assignment Resources. This file must be called **assign2.json**

COS20019 CCA Assignment 2 Mark Sheet

Student Name: Cong Thanh Ngo

Student Id: 103433609

Tutorial time:

Date of submission: 03/25/2022

Infrastructure Requirements Checklist (12 marks)

- ☐ VPC configured with 2AZs both with public and private subnets. All web servers and RDS in private subnets. Server used to create AMI template is stopped or deleted. (2)
- ☐ Public and private route tables route to IGW and NAT respectively. (2)
- ☐ Security groups created and correctly configured. (2)
- ☐ Autoscaling deploys Web server instances in private subnet (2)

- ☐ Scaling policy correct. (2)
- ☐ ELB configured and working correctly. (2)

Website Functionality Checklist (23 marks)

- ☐ Photos (>6) and their meta-data display from Web server when searched for. (3)
- ☐ Photos and descriptions can be uploaded to the website (4)
- ☐ Meta-data stored in DB and photos stored by PHP in S3. Titles in DB are matching S3 meta data. (3)
- ☐ Photos stored in S3 have restricted access. S3 bucket policy correct. (2)
- ☐ Scale out (New web server instances generated in the event of web server instance failure.) (3)
- ☐ Scale in (Web server instances scale back when above desired number.) (2)
- ☐ Website only accessible via ELB. (3)
- ☐ CloudFront distribution implemented correctly. (3)

Reflection (5)

- ☐ Challenges (2)
- ☐ 3 Highlights (3)

Documentation Checklist (make sure all the following are done - deduction if not)

- ☐ All resources tagged with your name and student id. (-5)

Submit to ESP a zip file called **assign2.zip** containing:

- ☐ Your CloudFormation JSON file assign2.json (+2)
- ☐ assign2.pdf as described in the submission section above (-20)
- ☐ Website source code files you have used to create your website. (-30)