COS20019 - Cloud Computing Architecture

Assignment 1 - part A

Creating and deploying a simple web page

Due Date: 30 March 9 am. Submission to Canvas

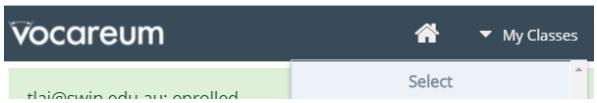
Contribution to final assessment: 5%, graded as pass/fail.

Before you start this Assignment

Make sure you have completed ACF Labs 3.
You will also need to create your own key pair and be able to access your EC2 instance via SSH using utilities like PuTTY and/or WinSCP. See the notes on **Remote Access to your EC2 Instance** on Canvas.

You have a choice of accounts you can use to complete this assignment:

1. Use the classroom in Vocareum managed environment. In the Vocareum console under `My Classes` select the `Cloud Computing Architecture` class.



This class gives you US$100 AWS credit. Use it carefully. This account is deleted at the end of semester.

Objectives

This assignment has the following objectives:

1. Get familiar with the AWS management console.
2. Launch your own EC2 instance.
3. Deploy your first PHP web page (PhotoAlbum) on Apache web server on your EC2 instance.

***Note**: In this introductory assignment you will create an EC2 Web server in the Default VPC. In general the default VPC is suitable only for experimental / toy deployments, and it use is considered bad practice for production resources.*
In the next assignment you will create your own secure VPC.

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

Task 1 – Launch your own Linux EC2 instance

Before launching an EC2 instance, a key pair is required for logging in to your instance in the future, follow the '*AWS Remote Access Tutorial*' on how to create one. Then launch an EC2 instance from the AWS management console in **us-east-1 region**. It must have the following properties:

- Amazon Machine Image: *Amazon Linux AMI 2018.03.0 (HVM)*
- Instance type: *t2.micro*
- Advanced Details – User data: enter the following script to automatically set up the Apache server at launch (a copy is on Canvas under the **Assignment Resources**):

```
#!/bin/bash
yum update -y
yum install -y httpd24 php70 mysql56-server php70-mysqlnd
service httpd start
chkconfig httpd on
usermod -a -G apache ec2-user
chown -R ec2-user:apache /var/www
chmod 2775 /var/www
find /var/www -type d -exec sudo chmod 2775 {} \;
find /var/www -type f -exec sudo chmod 0664 {} \;
echo "<?php echo '<h2>Welcome to COS80001. Installed PHP version: ' .
phpversion() . '</h2>'; ?>" > /var/www/html/phpinfo.php
```

- New security group named "*WebServer-SG*" that allows only necessary traffic types (SSH, HTTP, HTTPS) to reach the instance from anywhere. Do not allow every traffic types.

Other configurations should be left as default such as default Virtual Private Cloud (VPC), default subnet, etc.
Allow a few minutes for the instance to launch and execute the commands in the above script, the Instance State and Status Check should change to '*running'* and '*2/2 checks passed'*, respectively. After that, visit *http://your.public.dns.amazonaws.com/phpinfo.php*, if you see a welcome page, that means the EC2 instance and Apache server have been installed correctly.

Note: Due to the pay-as-you-go pricing approach, the longer your instances run the more you pay; therefore, you are advised to stop the instances after each working session.

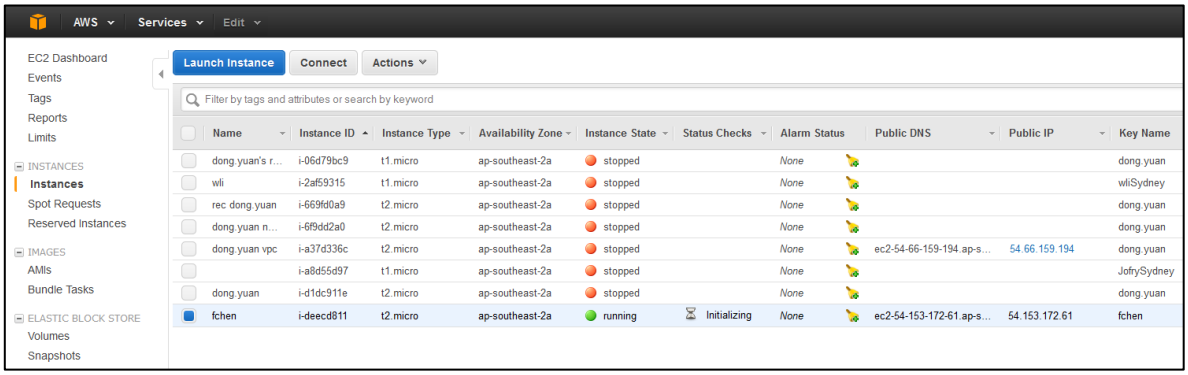


Figure 1 - AWS management console - EC2

Hint: Don't forget file and directory names in Linux are case sensitive

Task 2 – Create a PHP web page (PhotoAlbum)

Create a PHP web page (*upload.php*) with the functionality as visualised as below. Feel free to individualise you web page as you wish.

Figure 2 - Photo uploader page (upload.php)

The Photo Album link at the bottom of the page should link to a web page called *getphotos.php*. For this part of Assignment 1 a all this page needs to do is display a simple photo. For assignment 1 b you need to add the search functionality.

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); thus the *COS20019* folder must be in the Apache document root folder. Follow the '*AWS Remote Access Tutorial*' to learn how to transfer files to the Linux EC2 instance.

```
COS20019/
        . . . AWS SDK, other support libraries
        photoalbum/
                upload.php
                getphotos.php
                . . . other PHP, HTML, CSS, JavaScript files
```

You do not need to implement any upload functions at this stage. After having the website deployed on the Apache server on your instance, your web page should be accessible from anywhere on the Internet via this URL:

http://your.public.dns.amazonaws.com /cos20019/photoalbum/upload.php

You should try accessing your website from different devices on different networks to make sure it works correctly.

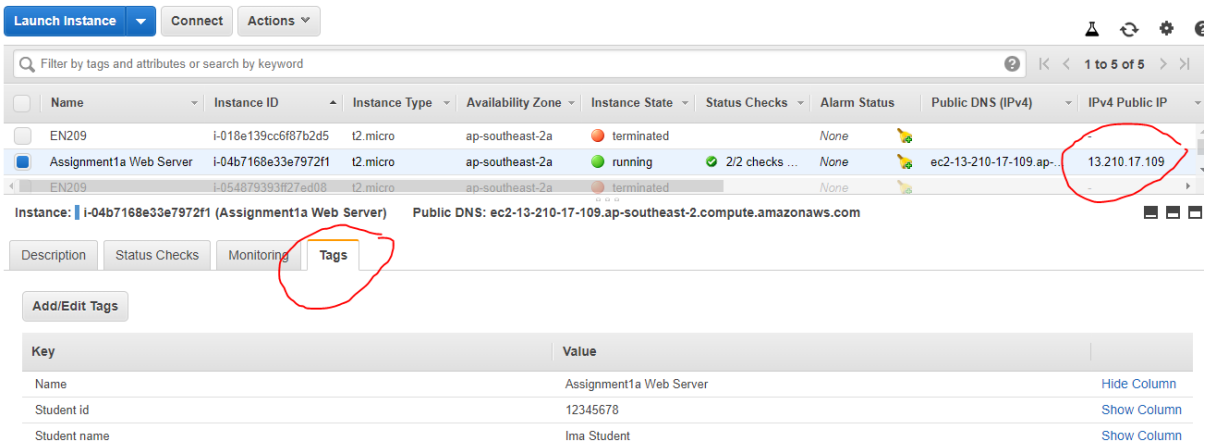
Submission

The code for this assignment does not need to be submitted.

As there is not an opportunity to demonstrate your assignment, you will need to document your Web site deployment in a PDF document called **assignment1a.pdf** to be submitted to Canvas. The document should include the following:

1. Title page with you name and student id.
2. URL of your **upload.php** on your EC2 web server so your tutor can view the web site.
3. Screenshots of
    a. Your EC2 instances page showing public ip address and tags (example below)
    b. EC2 security group showing the inbound rules
    c. A screenshot of an expanded listing of your files in the /var/www/html directory e.g. using WinSCP.
    d. Your web page.

Note: all screenshots should include something that identifies you: e.g. a tag as below or your AWS account number on the console. You do not have to include your source code in this submission.



Note: This is a formative assignment. That is, an assignment designed to provide feedback and ensure you have mastered some basic techniques. If your assignment submission fails this time, you have one week to make corrections and resubmit to pass.

FAQ

What happens if assignment submission is graded as a 'fail'?
You will have to repeat the task and submit in the following week's lab session. Students can repeat the task and submit for feedback again. If your submission is graded as 'fail' twice then you may fail this unit.

What happens if a student is unable to submit the assignment?
If you are unable to submit due to medical reasons, then a doctor certificate will have to be shown to the Convenor of the unit. Under normal conditions, ***all students are expected to make a submission by the due date, otherwise the assignment is graded as a fail***.

Requirements Checklist

- ☐ The web page **`upload.php`** is served from your EC2 instance and correctly displayed on a browser.
- ☐ The Photo Album link at the bottom of the page links to a web page called *getphotos.php* that displays a photo.