



COS20019

Cloud Computing Architecture

Assignment 1 – Part A
Creating and Deploying a Web Page

Truong Ngoc Gia Hieu
105565520

Due date: This assignment will be assessed by your tutor in your tutorial in **Week 4**. No submission is required. **You must attend this demonstration to have your assignment assessed and be eligible for future assignments.**

Weighting: 5%

Preparation:

- ACF Labs 2 & 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. For more details, see *Remote Access to an EC2.pdf* file.

All supporting materials mentioned in this document can be found in the corresponding assignment page on Canvas

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.
4. Make cost-efficient use of your budget. Terminate any unnecessary extra resources that you created

AWS Accounts

You have a choice of accounts/environments you can use to complete the assignments in this unit.

1. **AWS Academy Learner Lab (recommended):** accessible through AWS Canvas. Note that this is NOT the sandbox in ACA/ACF courses that you use for your weekly labs. This is a managed environment that allows your tutor to gain access to your AWS console so your work can be marked/troubleshooted. This class gives you **US\$100 credit**. **Use it carefully.** This account is deleted at the end of the semester.
2. **Regular AWS account (NOT recommended):** new AWS accounts are eligible for a free tier. This gives you more freedom, but you need to be careful as you will be charged for the services if you go outside the free tier offering. Make sure to keep track of your AWS services usage (using Billing & Cost Management Dashboard) throughout the semester to avoid paying fees. This account is on-going, but some services are no longer free after 12 months. If you choose this option, you will need to create a (read-only) IAM user and provide its credentials to your tutor so they can mark the assignment

A. TASK 1 – Launch your own Linux EC2 instance

- First and foremost, opening the Learner Lab and then wait until the circle next to the AWS is green.

**Figure 1: Launching the AWS Console Home**

- After launching the AWS Console Home successfully, search and open the EC2 main homepage to create own key pair.

Figure 2: EC2 main homepage

- Then, configure settings for own key pair including:
 - + Name: Assignment1a
 - + Key pair type: RSA
 - + Private key file format: .pem

Create key pair Info

Key pair
A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name
 The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type Info RSA ED25519

Private key file format
 .pem For use with OpenSSH
 .ppk For use with PuTTY

Figure 3: Key pair configuration

- As I expect, the own key pair named “Assignment1a” is successfully created key pair and the key pair “Assignment1a” has downloaded as “Assignment1a.pem”

The screenshot shows a terminal window with a green header bar indicating "Successfully created key pair". Below the header, there is a table titled "Key pairs (2) Info" with two entries:

Name	Type	Created	Fingerprint	ID
vockey	rsa	2025/09/23 10:18 GMT+7	95:b8:fc:6a:1e:b8:8f:90:d6:29:97:4e:d2:a4:0a:...	key-007e5...
Assignment1a	rsa	2025/09/23 10:29 GMT+7	f2:07:33:97:ca:1c:fb:2b:a6:83:05:5a:42:b7:4a:...	key-04197..

Figure 4: Successfully created key pair

- Then, I open the PuTTY Key Generator which convert file format from .pem to .ppk

The screenshot shows the PuTTY Key Generator application. In the "Key" section, the public key for pasting into OpenSSH authorized_keys file is displayed:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAQDD7cSW21ga0wVV4rKjS0xpmSLkir7cwyLSVPa/UCDxXmqbx5kzCfB8ZQvriZkt
faZSC051gA3pySm99Mee9V66pf9YCSaxyvhBR8jOUjzVqG592h4CtRVGhfM4Ru4i9X87V758i6TUAj/zbSI8p3FwaQ2e3MZ
JnCC5H7gai5H6eZ4VfFaA8dFvR8hdo8EE9152NCQMpA9cE1ynB0pyGA5hkHdvUDOVLEfzbntMJCFCWWq8odSjt
+omElRmByN7zCKsohbhkdq5v7ZJrvTfTP5jqlmR7CulilQDVmzbxYgxEbIBE/BnFAwT1ao5sP4Mt8qwLEiFfk8XwgOPBCL
```

The "Actions" section contains buttons for "Generate", "Load", "Save public key", and "Save private key". Under "Parameters", the "Type of key to generate" is set to RSA, and the "Number of bits in a generated key" is set to 2048.

Figure 5: Convert from .pem to .ppk

- After that, the file is successfully converted from .pem to .ppk and stored in the folder named “Assignment1a”

This PC > DATA (D:) > University > Sep_2025 > Cloud_Computing_Technology_COS20019 > Assignments > Assignment 1a				
Name	Date modified	Type	Size	
Assingment1a-WebsiteFiles	9/20/2025 10:17 AM	File folder		
Assignment requirements	9/16/2025 2:32 PM	Microsoft Edge PD...	467 KB	
Assignment1a.pem	9/23/2025 10:29 AM	PEM File	2 KB	
Assignment1a.ppk	9/23/2025 10:56 AM	PPK File	2 KB	

Figure 6: Converting successfully

- Next, after setting up important feature, I decide to launch the EC2 instance with following settings:
- + Amazon Machine Image (AMI): Amazon Linux 2023 AMI
- + Instance type: t2.micro
- + Key Pair: Assignment1a
- + Security group named: “WebServer-SG”

The screenshot shows the AWS Lambda console interface. At the top, there's a search bar with the placeholder "Search our full catalog including 1000s of application and OS images". Below it, a "Quick Start" section lists several operating system and application options: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. To the right of this section is a "Browse more AMIs" button with a magnifying glass icon, accompanied by the text "Including AMIs from AWS, Marketplace and the Community". At the bottom, there's a detailed view of the "Amazon Machine Image (AMI)" for "Amazon Linux 2023 kernel-6.1 AMI", which includes the AMI ID, virtualization type (hvm), ENA support, and root device type (ebs). A "Free tier eligible" badge is also present.

Figure 7: Name and AMI configuration

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Windows base pricing: 0.0162 USD per Hour

On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour

On-Demand SUSE base pricing: 0.0116 USD per Hour On-Demand RHEL base pricing: 0.026 USD per Hour

On-Demand Linux base pricing: 0.0116 USD per Hour

All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Assignment1a

[Create new key pair](#)

Figure 8: Instance type and Key pair configuration

Security group name - required

WebServer-SG

Type Info	Protocol Info	Port range Info
ssh	TCP	22
Source type Info	Source Info	Description - optional Info
Anywhere	Add CIDR, prefix list or security group	e.g. SSH for admin desktop
0.0.0.0/0 X		

▼ Security group rule 2 (TCP, 443, 0.0.0.0/0) [Remove](#)

Type Info	Protocol Info	Port range Info
HTTPS	TCP	443
Source type Info	Source Info	Description - optional Info
Anywhere	Add CIDR, prefix list or security group	e.g. SSH for admin desktop
0.0.0.0/0 X		

▼ Security group rule 3 (TCP, 80, 0.0.0.0/0) [Remove](#)

Type Info	Protocol Info	Port range Info
HTTP	TCP	80

Figure 9: Security group configuration

User data - optional | [Info](#)

Upload a file with your user data or enter it in the field.

[Choose file](#)

```
#!/bin/bash
PHP_VERSION=8.2
dnf update -y
dnf install -y httpd wget php${PHP_VERSION}-fpm php${PHP_VERSION}-mysqlnd
php${PHP_VERSION} php${PHP_VERSION}-devel
dnf install -y mariadb105-server
sed -i "s/upload_max_filesize = 2M/upload_max_filesize = 10M/g" /etc/php.ini
systemctl start httpd
systemctl enable httpd
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 COS20019. Installed PHP version: '. phpversion() ."
```

User data has already been base64 encoded

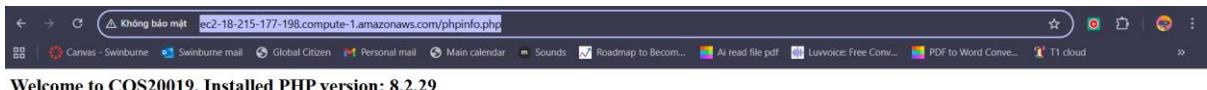
Figure 10: User Data Script

- After configuring, the EC2 instance has successfully launched

Instances (1/2) Info		Connect	Instance state ▾	Actions ▾	Launch instances	▼		
		All states ▾				◀ 1 ▶ 🔍		
▢	Name ⚡	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
▢	Assignment1a	i-0c1bb00b56e0a9959	Terminated	t2.micro	-	View alarms +	us-east-1a	-
<input checked="" type="checkbox"/>	Assign1a	i-0c6ecc7988428d131	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-18-2-

Figure 11: EC2 instance successfully launched

- According to the requirement, after waiting the instance launch successfully, visit the link <http://ec2-18-215-177-198.compute-1.amazonaws.com/phpinfo.php> and picture below is the result.



B. TASK 2 – Create a PHP website (Photo Album)

- First and foremost, I configure these following settings to access the WinSCP:
 - + Host Name: ***ec2-18-215-177-198.compute-1.amazonaws.com***
 - + User Name: ***ec2-user***
 - + ***Advanced => SSH => Authentication:***
- Then, upload the file “Assignment1a.php” to the Private Key Field:
- As a result, the WinSCP has been activated

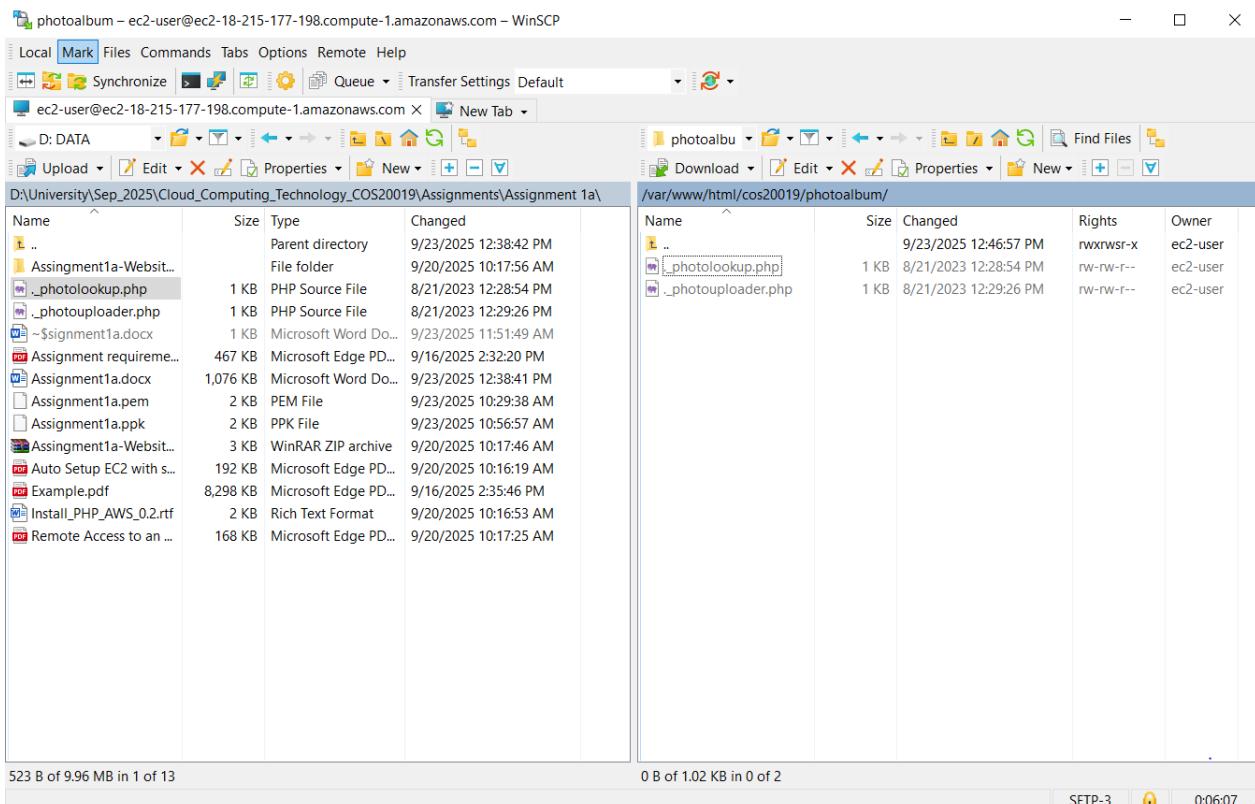
**Figure 12: WinSCP**

Photo uploader

Photo title:

Select a photo: Chon tệp Không có tệp nào được chọn

Description:

Date: dd/mm/yyyy

Keywords (comma-delimited e.g. keyword1; keyword 2, ...):

[Photo Lookup](#)

Figure 13: “photouploader.php” file

Photo lookup

Photo title:

Keyword:

From Date: dd/mm/yyyy

To Date: dd/mm/yyyy

[Photo Uploader](#)

Figure 14: “photolookup.php” file

EC2

Instances (1/2) [Info](#)

Last updated less than a minute ago [C](#) Connect [Instance state](#) Actions [Launch instances](#)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input type="checkbox"/>	Assignment1a	i-0c1bb00b56e0a9959	Terminated	t2.micro	-	View alarms +	us-east-1a	-
<input checked="" type="checkbox"/>	Assign1a	i-0c6ecc7988428d131	Stopped	t2.micro	-	View alarms +	us-east-1a	-

Figure 15: Stop the Assign1a instance