



ADNAN MENDERES UNIVERSITY CSE423 CLOUD COMPUTING

LAB 01 Virtual Machines & Compute Instances

Introduction

In this exercise, you will learn how to launch an Amazon EC2 instance and setup a web server on the instance. You will submit documents/snapshots related the questions asked in “exercise” sections (Exercise1-2). Zip all your files as a single folder. Named the zipped folder as:

studentNo_NameSurname_LAB1.

- Answer of Exercise1 should be in.doc or.pdf format, for answer of Exercise2 submit your python code and screenshot (public dns should be seen in url section!!).
- Late submissions are not allowed.
- You should do homework YOURSELF. Group working is not allowed.
- Copy homework will be evaluated as 0.
- Use Google Classroom for your questions. Do not send private messages.

What is Amazon EC2 ?

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

Join to AWS Classroom By Accepting Invitation Mail (highly recommended, no fee)

To join AWS Academy Classroom, accept the Classroom invitation which is send to your student email. A "Classroom" is a hands-on learning environment for you to access AWS services and practice AWS without cost or fees. Each student has \$100 credits to use in AWS services. You should stop the instance when you are not using. For more info about registering AWS classroom, please review a shared document: *AWS Academy Learner Lab - Student Guide.pdf*

AWS Academy Student Login Page: awsacademy.com/vforcesite/LMS_Login

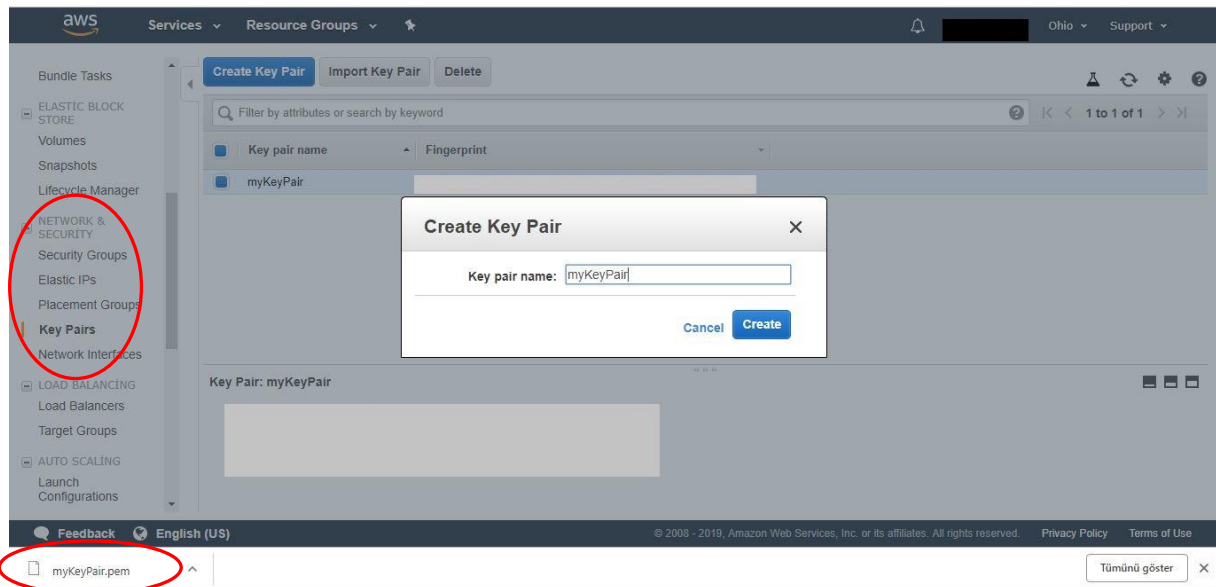
Launching Amazon EC2 Instance

Follow the steps as shown in the screenshots:

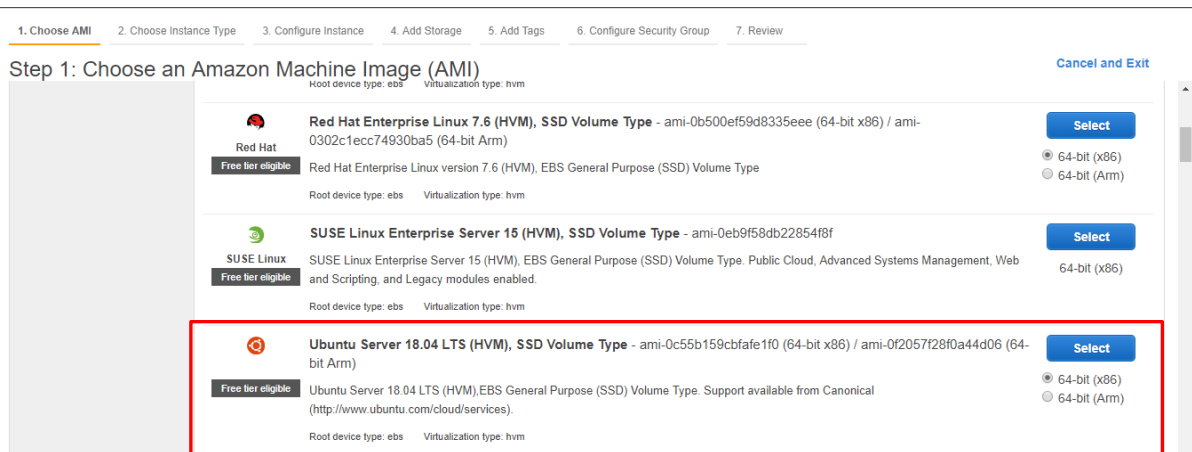
- 1- Select **services >> EC2**. From the **Amazon EC2 Dashboard**, under the **Network & Security tab**, click **Key Pairs**. In opening window, click on **create a key-pair button**. Enter a name for the new key pair in the Key pair name field of the Create Key Pair dialog box, and then choose **Create**. Save the **key-pair (.pem) file**. You will require this file later for securely connecting to EC2 instances. pem file needs to be converted to .ppk file in further stages. As an alternative, you may directly download **key-pair (.ppk) file**. In this way, there won't be need for converting this file for further use.

ADNAN MENDERES UNIVERSITY

CSE423 CLOUD COMPUTING



- 2- From EC2 Dashboard, click on the **launch instance button** to open the wizard. Select **free tier eligible "Ubuntu Server 18.04 LTS (HVM), SSD Volume Type"**.



- 3- Choose the instance type as shown below, click next.

ADNAN MENDERES UNIVERSITY

CSE423 CLOUD COMPUTING

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: **All instance types** **Current generation** **Show/Hide Columns**

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Instance Details](#)

4- Configure the instance as shown below, click next.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 3: Configure Instance Details

Auto-assign Public IP ☒ Use subnet setting (Enable)

Placement group ☐ Add instance to placement group

Capacity Reservation ☐ Open [Create new Capacity Reservation](#)

IAM role ☐ None [Create new IAM role](#)

Shutdown behavior ☐ Stop

Enable termination protection ☐ Protect against accidental termination

Monitoring ☐ Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy ☐ Shared - Run a shared hardware instance
Additional charges will apply for dedicated tenancy.

Elastic Inference ☐ Add an Elastic Inference accelerator
Additional charges apply.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

5- Configure the instance storage as follows, click next.

ADNAN MENDERES UNIVERSITY

CSE423 CLOUD COMPUTING

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type ⓘ	Device ⓘ	Snapshot ⓘ	Size (GiB) ⓘ	Volume Type ⓘ	IOPS ⓘ	Throughput (MB/s) ⓘ	Delete on Termination ⓘ	Encrypted ⓘ
Root	/dev/xvda	snap-0d2b36e2ba06f14f6	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

[Add New Volume](#)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Tags](#)

6- Add tags to identify the instance as shown below, click next.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)	Instances ⓘ	Volumes ⓘ	
Name	Web Server	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="✕"/>

[Add another tag](#) (Up to 50 tags maximum)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Security Group](#)

7- Create a new security group. Open ports 22 and 80. Add these two rules as shown below, click **review and launch**.

ADNAN MENDERES UNIVERSITY

CSE423 CLOUD COMPUTING

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Anywhere	0.0.0.0/0:::0
Custom TCP	TCP	80	Anywhere	0.0.0.0/0:::0

Add Rule

Warning

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Previous **Review and Launch**

- 8- Review your instance launch details, click **launch** to assign a key pair to your instance and complete the launch process.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

Warning

Improve your instances' security. Your security group, launch-wizard-1, is open to the world. Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details [Edit AMI](#)

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0c55b159cbfafa1f0

Free tier eligible

Root Device Type: ebs Virtualization type: hvm

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups [Edit security groups](#)

Cancel Previous **Launch**

- 9- In opening window, select an existing key-pair which is created in step1, fill other areas as shown below, click **launch instances** button.

ADNAN MENDERES UNIVERSITY

CSE423 CLOUD COMPUTING

Select an existing key pair or create a new key pair ✕

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair ▼

Select a key pair

myKeyPair ▼

☒ I acknowledge that I have access to the selected private key file (myKeyPair.pem), and that without this file, I won't be able to log into my instance.

Cancel Launch Instances

Exercise 1: Describe the type of virtualization is used by the EC2 instance launched?

Connecting to Your Linux Instance from Windows Using PuTTY

PuTTY is a software terminal emulator for Windows and Linux. It provides a text user interface to remote computers running any of its supported protocols, including SSH and Telnet. Download and install PuTTY from the PuTTY download page. Go to <https://www.putty.org/> and choose file which has **.msi** extension for downloading program. Follow the instructions for installing program.

Converting Your Private Key Using PuTTYgen (If you have .pem file, this is necessary. If you have .ppk file, skip this step.)

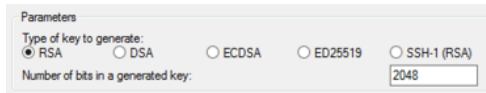
PuTTY does not natively support the private key format (.pem) generated by Amazon EC2. PuTTY has a tool named PuTTYgen, which can convert keys to the required PuTTY format (.ppk). You must convert your private key into this format (.ppk) before attempting to connect to your instance using PuTTY.

ADNAN MENDERES UNIVERSITY

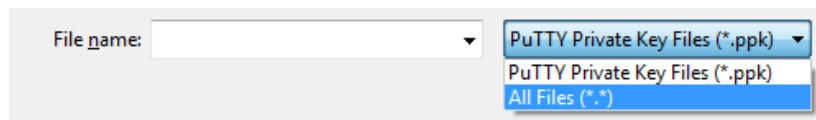
CSE423 CLOUD COMPUTING

To convert your private key :

- 1- Start PuTTYgen (for example, from the **Start** menu, choose **All Programs > PuTTY > PuTTYgen**).
- 2- Under **Type of key to generate**, choose RSA



- 3- Choose **Load**. By default, PuTTYgen displays only files with the extension **.ppk**. To locate your **.pem** file, select the option to display files of all types.



- 4- Select your **.pem** file for the key pair that you specified when you launched your instance, and then choose **Open**. Choose **OK** to dismiss the confirmation dialog box.
- 5- Choose **Save private key** to save the key in the format that PuTTY can use. PuTTYgen displays a warning about saving the key without a passphrase. Choose **Yes**.
- 6- Specify the same name for the key that you used for the key pair (for example, my-key-pair). PuTTY automatically adds the **.ppk** file extension.

Your private key is now in the correct format for use with PuTTY. You can now connect to your instance using PuTTY's SSH client.

Starting a PuTTY Session

- 1- Start PuTTY (from the **Start** menu, choose **All Programs > PuTTY > PuTTY**).
- 2- In the **Category** pane, choose **Session** and complete the following fields:

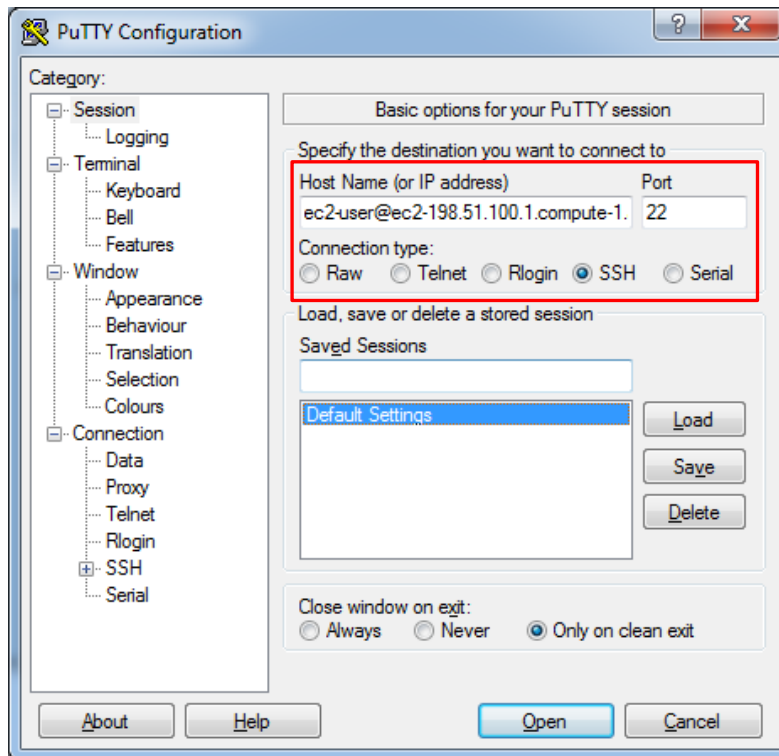
In the Host Name box, enter **user_name@public_dns_name**. Be sure to specify the appropriate user name for your AMI (Amazon Machine Instance). Ex:

For an Ubuntu AMI as in our example, the **user_name** is **ubuntu**. For a **public_dns_name**, you can find it from your **EC2 Dashboard >> Running Instance tab**.

ADNAN MENDERES UNIVERSITY

CSE423 CLOUD COMPUTING

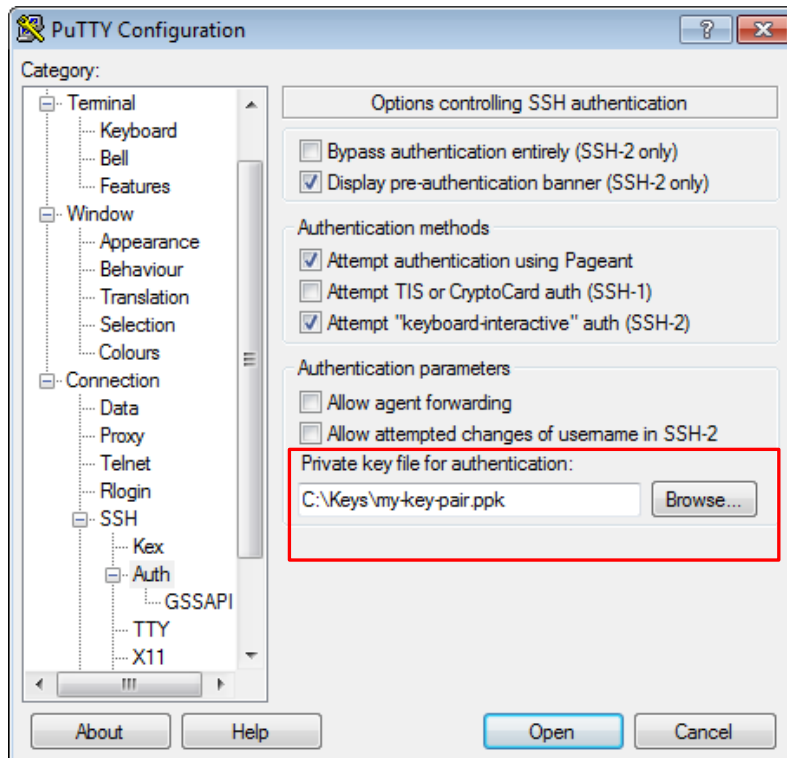
- 3- Under **Connection type**, select **SSH**.
- 4- Ensure that **Port** is 22.



- 5- In the **Category** pane, expand **Connection**, expand **SSH**, and then choose **Auth**. Complete the following:
 - a. Choose **Browse**.
 - b. Select the .ppk file that you generated for your key pair, and then choose **Open**.
 - c. Choose **Open** to start the PuTTY session.

ADNAN MENDERES UNIVERSITY

CSE423 CLOUD COMPUTING



- 6- If this is the first time you have connected to this instance, PuTTY displays a security alert dialog box that asks whether you trust the host you are connecting to.
- 7- Choose **Yes**. A window opens and you are connected to your instance. You can write your commands to there.

Install Apache Web Server On The Instance

- 1- Connect to your Ubuntu Server Instance via previous steps. Putty command window will open.
- 2- Write following commands to there:
write **`sudo apt-get update`**, press **enter**
write **`sudo apt-get install apache2`**, press **enter**

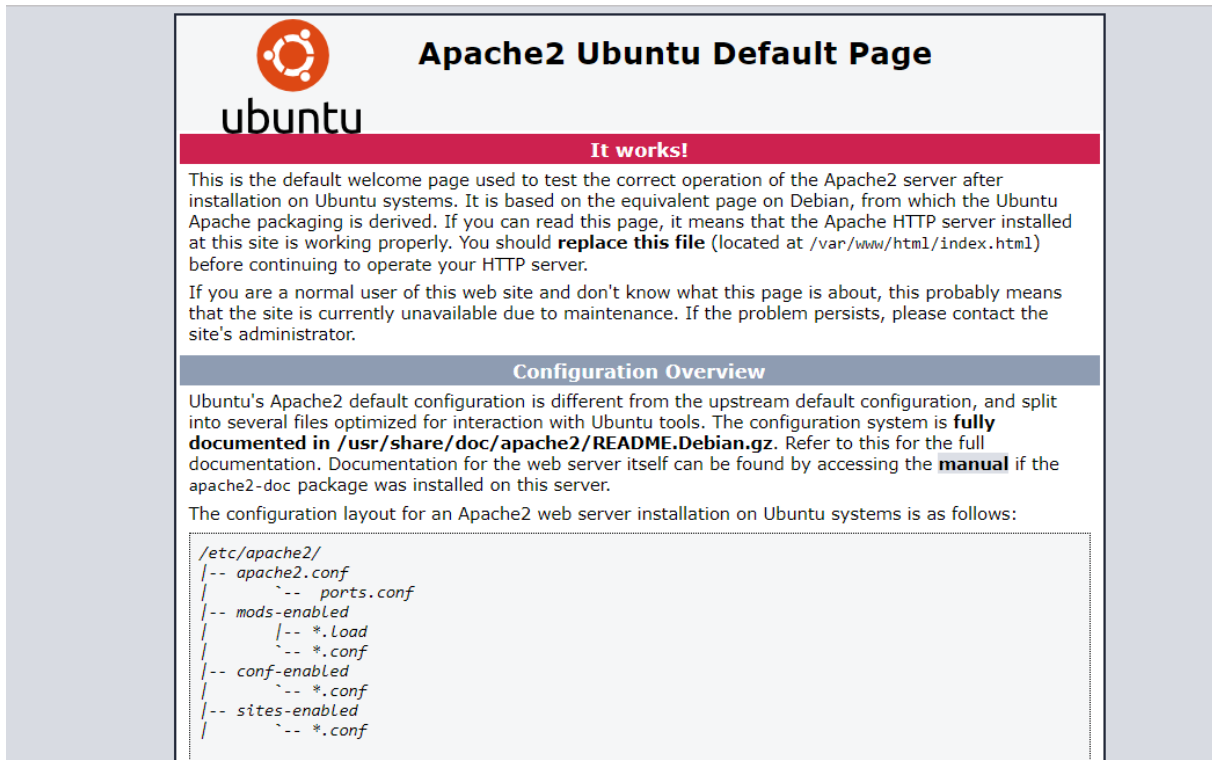
To view default webpage, open the following URL in browser:

`http://<Public-IP-Address>`

You can find **public IP address** of your instance from EC2 dashboard. Default webpage is shown below:

ADNAN MENDERES UNIVERSITY

CSE423 CLOUD COMPUTING



The screenshot shows the Apache2 Ubuntu Default Page. At the top, there is the Ubuntu logo and the text "Apache2 Ubuntu Default Page". Below this, a red banner says "It works!". The main text explains that this is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It mentions that the page is based on the equivalent page on Debian. A red box highlights the instruction: "You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server." Below this, it says: "If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator." A section titled "Configuration Overview" follows, explaining that Ubuntu's Apache2 default configuration is different from the upstream default configuration and is split into several files. It mentions that the configuration system is "fully documented in `/usr/share/doc/apache2/README.Debian.gz`". It also states that documentation for the web server itself can be found by accessing the "manual" if the `apache2-doc` package was installed. Finally, it says: "The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:" and shows a directory tree structure for `/etc/apache2/`.

```
/etc/apache2/  
|-- apache2.conf  
|   |-- ports.conf  
|-- mods-enabled  
|   |-- *.load  
|   |-- *.conf  
|-- conf-enabled  
|   |-- *.conf  
|-- sites-enabled  
|   |-- *.conf
```

Replace the default web page with your own HTML page as follows:

cd /var/www/html/ //we changed directory to var/www/html/

sudo mv index.html index-old.html //we changed index.html's name to index-old.html

sudo vim index.html //we create new file whose name is index.html & we can edit this file

(About "vim" Command

Vim is an editor to create or edit a text file. There are two modes in vim. One is the **command mode** and another is the **insert mode**. In the command mode, user can move around the file, delete text, etc. In the insert mode, user can insert text.

Some useful commands for "vim"

a Open insert mode. Append text following current cursor position.

ESC and **:w** save changes in file.

ESC and **:x** exit from file, return command page.

For more info about **vim** please see

https://www.radford.edu/~mhtay/CPSC120/VIM_Editor_Commands.htm)

ADNAN MENDERES UNIVERSITY

CSE423 CLOUD COMPUTING

After last command, you can add html codes into index.html file via “**vim**”, you can add following code:

```
<!DOCTYPE html>
<html>
<body>

<h1 style="background-color: DodgerBlue;">Hello World</h1>

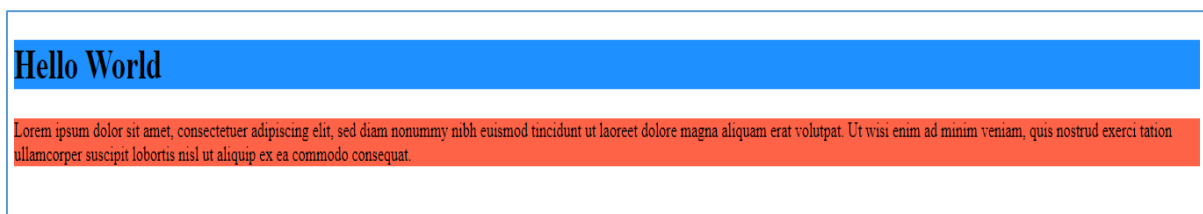
<p style="background-color: Tomato;">
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh
eismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit
lobortis nisl ut aliquip ex ea commodo consequat.
</p>

</body>
</html>
```

After adding code, press **ESC** and write **:w** then press **enter** to save the file. Go to your browser and write following URL to display the newly edited html file:

http://<Public-IP-Address>

Following screen will display in your browser:



Setting Up Flask On EC2 Instance

Flask is a lightweight web application framework written in Python. To install flask on the instance, write following commands to putty command screen. For more information about flask, visit: <http://flask.pocoo.org/>

```
sudo apt install python3-pip -y
sudo apt install python3-venv
python3 -m venv testenv
source testenv/bin/activate
pip install Flask
```

Add New Rule To EC2 Instance

Go to EC2 dashboard, select **Running Instance>> Security Groups>>** click **launchWizard**. In opening screen, in below section of page, select **inbound tab** and click **edit**. Add last rule and save.

ADNAN MENDERES UNIVERSITY

CSE423 CLOUD COMPUTING

Edit inbound rules ✕

Type <i>i</i>	Protocol <i>i</i>	Port Range <i>i</i>	Source <i>i</i>	Description <i>i</i>	
HTTP ▾	TCP	80	Custom ▾ 0.0.0.0/0	e.g. SSH for Admin Desktop	✕
HTTP ▾	TCP	80	Custom ▾ :::/0	e.g. SSH for Admin Desktop	✕
SSH ▾	TCP	22	Custom ▾ 0.0.0.0/0	e.g. SSH for Admin Desktop	✕
SSH ▾	TCP	22	Custom ▾ :::/0	e.g. SSH for Admin Desktop	✕
Custom TCP F ▾	TCP	5000	Anywhere ▾ 0.0.0.0/0, :::/0	e.g. SSH for Admin Desktop	✕

Add Rule

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Cancel Save

Create and run a hello world python web application as follows (*Note that according to your interpreter version; you may change your code slightly, e.g. changing single quotation mark to double quotation marks*):

```
vim hello.py
```

```
----- hello.py-----
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello World!'

if __name__ == '__main__':
    app.run(host='0.0.0.0')
```

```
python hello.py
```

```
python3 hello.py
```

To view the web application page, open the following URL in a browser:

<http://<Public-IP-Address>:5000>

Web Application To Get Instance Meta-data

Amazon EC2 provides the following URL to view all categories of instance metadata from within a running instance (for more info please see

https://docs.aws.amazon.com/en_us/AWSEC2/latest/UserGuide/ec2-instance-metadata.html)

ADNAN MENDERES UNIVERSITY

CSE423 CLOUD COMPUTING

use pip:

pip install ec2-metadata

There's a singleton instance at the name **ec2_metadata** which should cover 90% of use cases. Use it like in .py file:

```
from ec2_metadata import ec2_metadata
ec2_metadata.region
```

You can reach other metadata values (e.g. instance-id, ami-launch-index, public-ipv4...) as shown above.

Exercise 2:

Create a flask web application that displays the instance meta-data as shown in the following figure (for more info please see <https://github.com/adamchainz/ec2-metadata>). You should print metadata-value pairs on your browser in a table with header and border lines like given below:

Metadata	Value
instance-id	i-10a64379
ami-launch-index	0
public-hostname	ec2-203-0-113-25.compute-1.amazonaws.com
public-ipv4	67.202.51.223
local-hostname	ip-10-251-50-12.ec2.internal
local-ipv4	10.251.50.35

Submit the flask application python file and a screenshot of the web page showing the instance meta-data in bordered table, your public hostname should be seen in the URL. (note that your meta-data contains your instance configuration details).

Useful CLI Commands

Command	Explanation
pip install ec2-metadata	API that allows you to use EC2 metadata service
sudo vim hello.py	Opens the file with super user privilege (for append/update operations)
cd /var/www/html	Navigates into the desired directory from anywhere
lsof -i :8000	Checks who is using port 8000
kill 760	Terminates a process with pid : 760