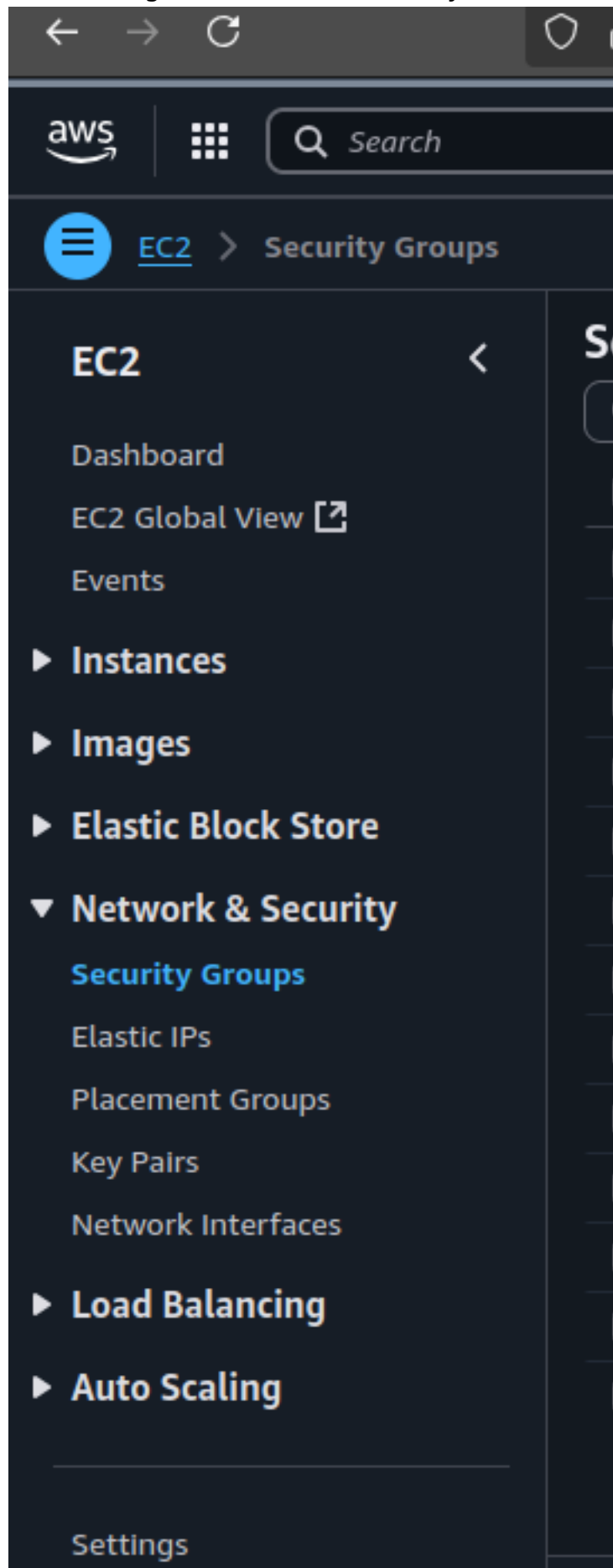


Creating a basic EC2 instance

Creating a Security group:

Under EC2, go to network and security tab and choos security group.



Click on the orange tab to create a security group.
Then name it something in relation to what the project is.
I use the name for the description also.
Inbound rules is http with source of Anywhere ipv4.

Create security group [Info](#)

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name [Info](#)

FirstSG

Name cannot be edited after creation.

Description [Info](#)

FirstSG

VPC [Info](#)

vpc-01d3e8eac088e696f

Inbound rules [Info](#)

Type	Protocol	Port range	Source	Description - optional	
HTTP	TCP	80	Anywhere...	web-server	Delete

[Add rule](#)

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

Outbound rules [Info](#)

Type	Protocol	Port range	Destination	Description - optional	
All traffic	All	All	Custom		Delete

[Add rule](#)

Rules with destination of 0.0.0.0/0 or ::/0 allow your instances to send traffic to any IPv4 or IPv6 address. We recommend setting security group rules to be more restrictive and to only allow traffic to specific known IP addresses.

I add a tag and then press the orange tab to create the security group.

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional	
Name	first-sg	Remove

[Add new tag](#)

You can add up to 49 more tags.

[Cancel](#) [Create security group](#)

Under Instances in EC2, click on launch instance.
Give your instance a name. I named mine Class7_Sep9.
Leave the Image on default Amazon Linux and default Instance type.

EC2 > Instances > Launch an instance

Launch an instance

Info

Amazon EC2 allows you to create virtual machines, or Instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Info

Name

Class7_Sep9

Add additional tags

Application and OS Images (Amazon Machine Image)

Info

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Search our full catalog including 1000s of application and OS images

Recents

Quick Start

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

SUSE Linux

Debian

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 kernel-6.1 AMI

ami-00ca32bbc84273381 (64-bit (x86), uefi-preferred) / ami-0aa7db6294d00216f (64-bit (Arm), uefi)

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

Amazon Linux 2023 (kernel-6.1) is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.8.20250818.0 x86_64 HVM kernel-6.1

Architecture

64-bit (x86)

Boot mode

uefi-preferred

AMI ID

ami-00ca32bbc84273381

Publish Date

2025-08-13

Username

ec2-user

Verified provider

Instance type

Info | Get advice

Instance type

t3.micro

Family: t3 2 vCPU 1 GiB Memory Current generation: true

On-Demand Ubuntu Pro base pricing: 0.0139 USD per Hour On-Demand SUSE base pricing: 0.0104 USD per Hour

On-Demand Linux base pricing: 0.0104 USD per Hour On-Demand RHEL base pricing: 0.0392 USD per Hour

On-Demand Windows base pricing: 0.0196 USD per Hour

Free tier eligible

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

Under Key pair if this is your first time create a new key pair.

3/9

Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

Class7_Sep9

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

Key pair type


☒ RSA
RSA encrypted private and public key pair

☐ ED25519
ED25519 encrypted private and public key pair

Private key file format

☒ .pem
For use with OpenSSH

☐ .ppk
For use with PuTTY

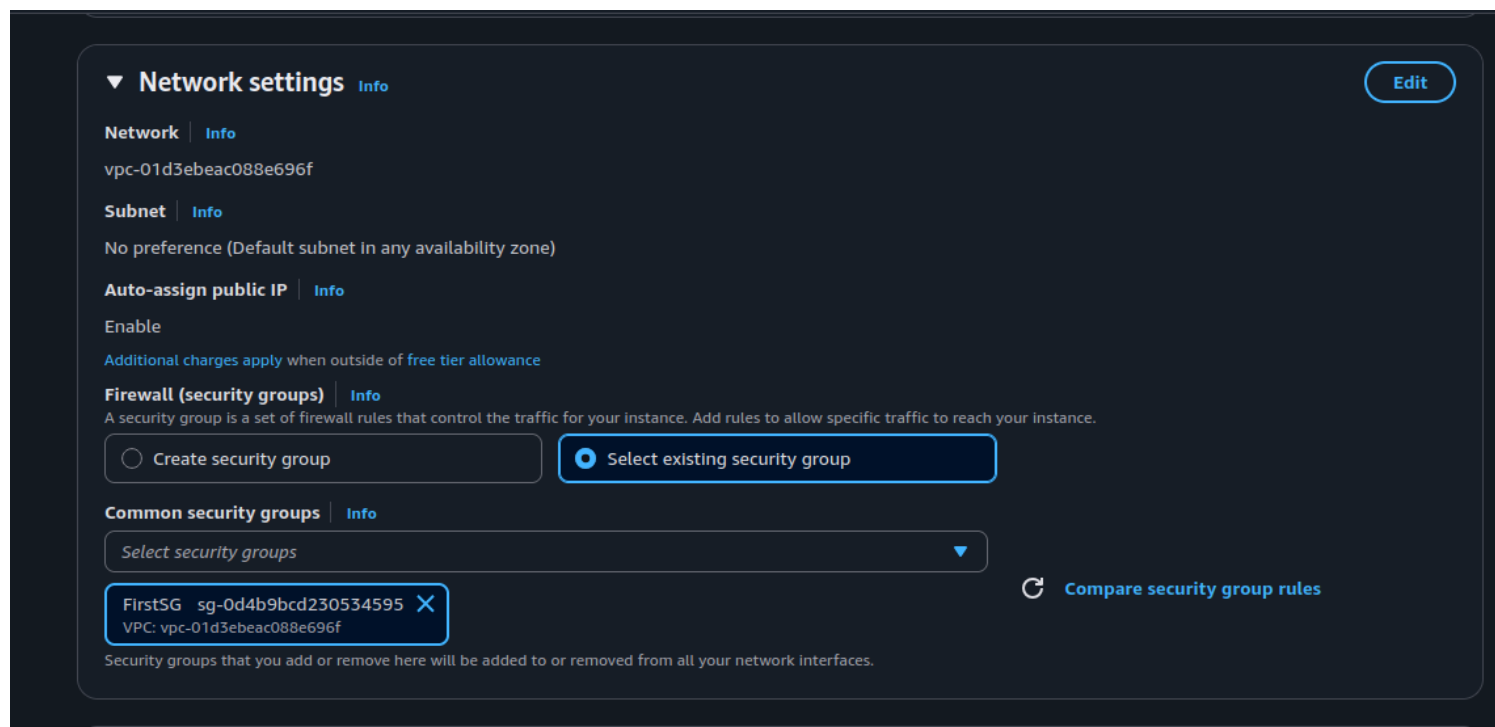
 When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Cancel

Create key pair

I used the same naming convention to keep it simple.

In the Network Settings we will choose a security group that we previously created before launching the EC2 instance.



I named it the same as the instance with SG at the end. I also added a inbound rule for http as the type, source is anywhere ipv4.

I then go into the advanced details and paste the startup.txt script into the user data before launching the instance for the first time.

Metadata response hop limit [Info](#)


2

Allow tags in metadata [Info](#)

Select

User data - optional [Info](#)

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

 Choose file

```
#!/bin/bash
# Use this for your user data (script from top to bottom)
# install httpd (Linux 2 version)
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd

# Get the IMDSv2 token
TOKEN=$(curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")

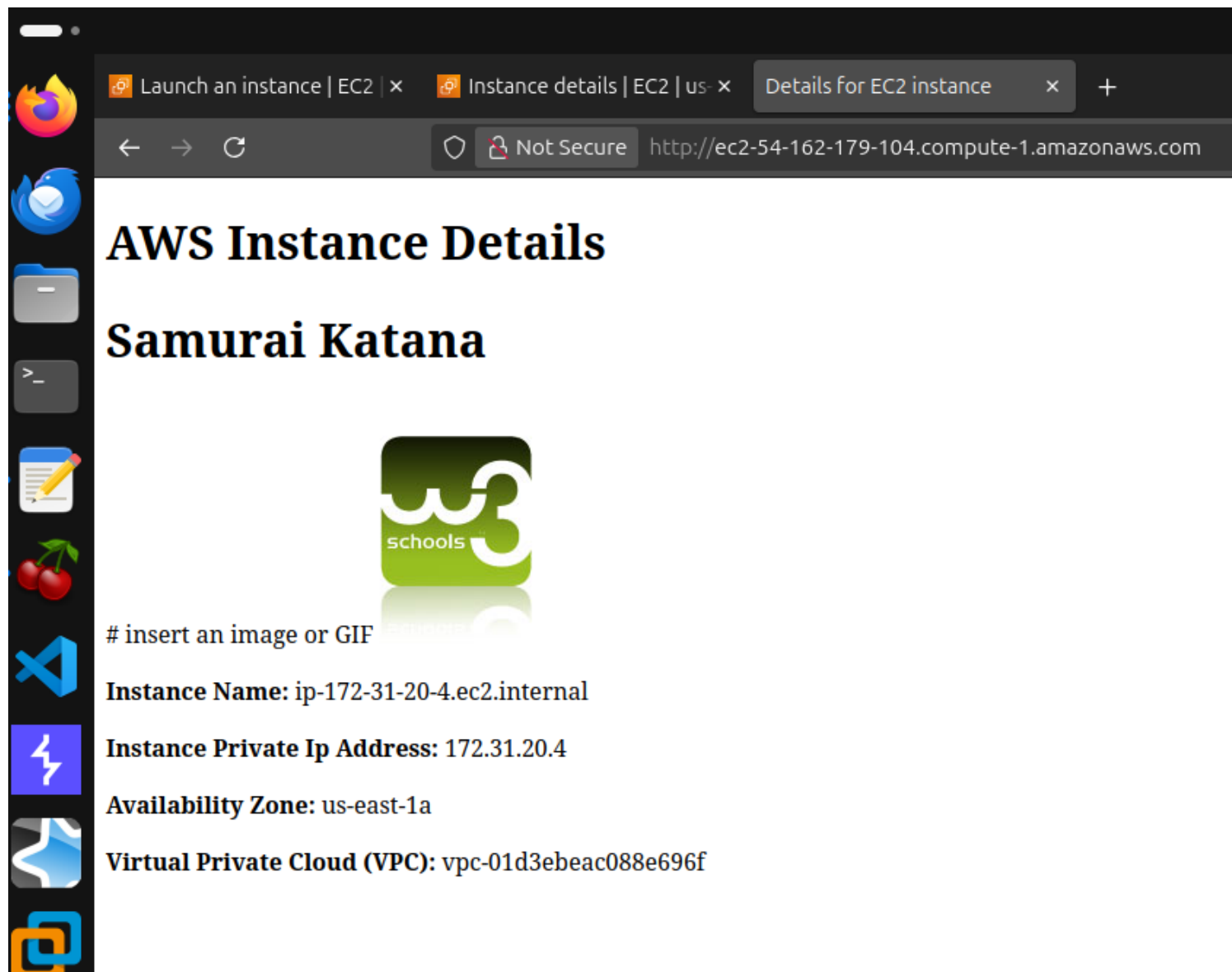
# Background the curl requests
curl -H "X-aws-ec2-metadata-token: $TOKEN" -s http://169.254.169.254/latest/meta-data/local-ipv4 &> /tmp/local_ipv4 &
```

☐ User data has already been base64 encoded

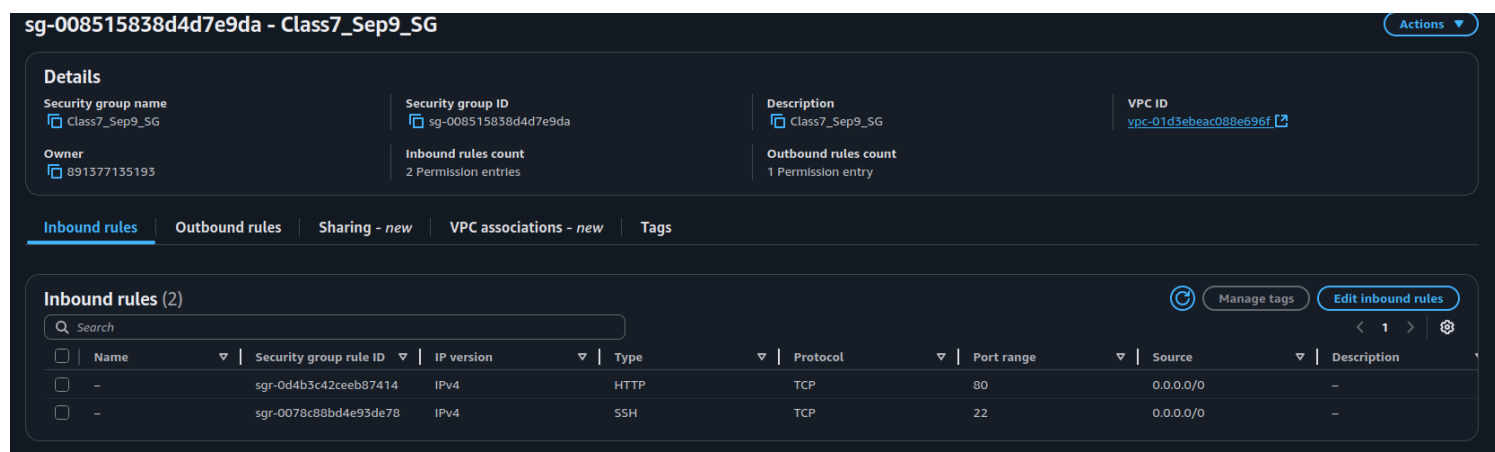
Now i launch the instance

After the instance launches i verify by going to the public dns on the browser.

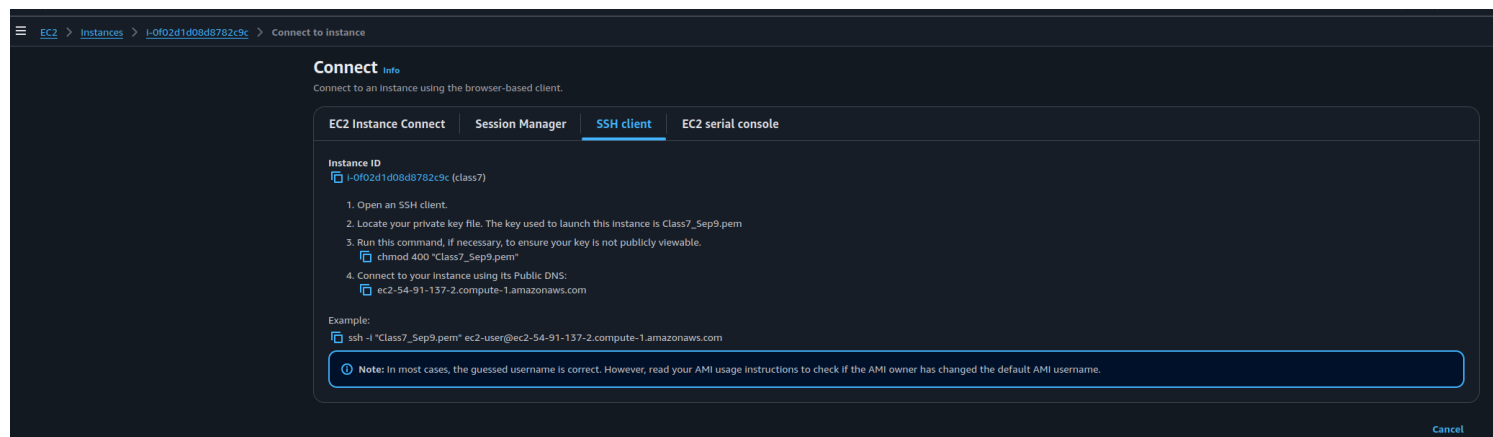
<http://>



In order to do the second part of the homework, i enabled ssh on the security group so that i can ssh into the VM.



After enabling ssh i connected to the instance according to AWs instructions:



So i first change the permissions of the .pem that i recieved that is currently on my computer with the command:

```
chmod 400 "Class7_Sep9.pem"
```

I then ssh into the VM:

```
ssh -i "Class7_Sep9.pem" ec2-user@ec2-54-91-137-2.compute-1.amazonaws.com
```



Now that i am into the VM i change to the directory that the webserver file is in.

```
cd /var/www/html/
```

Then:

```
sudo nano index.html
```

Paste the following into the editor:

```
<html lang=\"en\" class=\"h-100\">
<head>
<title>Details for EC2 instance</title>
</head>
<body>
<div>
<h1>WINNERS WIN!!!</h1>
<br>
<img src=\"https://www.nowtravelasia.com/wp-content/uploads/2018/08/trip-asia-travel.jpg\"
<br>
<p><b>"I, Larry Harris, THANK THEO AND Rob, FOR TEACHING ME ABOUT EC2s IN AWS. ONE STEP CLOSER TO ESCAPING KEISHA! "WITH THIS CLASS, I WILL NET $500.000 PER YEAR!"</b></p>
</div>
</body>
</html>
```

save it and then refresh the website to see the update.

Welcome To Larry's World



"I, Larry Harris, THANK THEO AND Rob, FOR TEACHING ME ABOUT EC2s IN AWS. ONE STEP CLOSER TO ESCAPING KEISHA! "WITH THIS CLASS, I WILL NET \$500.000 PER YEAR!"



"I found my wife on a party yacht in Leyte Phillippines! Her name is Beautiful!"