

# AWS EVALUATION

## 1. AWS Academy Cloud Foundations (Theory)

• Módulo 1 - Información general sobre los conceptos de la nube	Completa todos los ítems ✓
• Módulo 2 - Facturación y economía de la nube	Completa todos los ítems ✓
• Módulo 3 - Información general sobre la infraestructura global de AWS	Completa todos los ítems ✓
• Módulo 4 - Seguridad en la nube	Completa todos los ítems ✓
• Módulo 5 - Redes y entrega de contenido	Completa todos los ítems ✓
• Módulo 6 - Informática	Completa todos los ítems ✓
• Módulo 7 - Almacenamiento	Completa todos los ítems ✓
• Módulo 8 - Bases de datos	Completa todos los ítems ✓
• Módulo 9 - Arquitectura en la nube	Completa todos los ítems ✓
• Módulo 10 - Monitoreo y escalado automático	Completa todos los ítems ✓

## 2.a. Creación de una VPC donde instalar las instancias que vamos a usar.

Crear una VPC con las siguientes condiciones:

- Se trata de montar una infraestructura para una empresa que va a tener dos subredes, una pública y otra privada. Cada una de las subredes estará dispersa en dos zonas de disponibilidad de la región para más disponibilidad y seguridad.
- La zona pública deberá tener salida a Internet a través de un gateway propio creado para la VPC.
- El rango de direcciones de la VPC será el CIDR 10.0.0.0/16 y cada una de las dos subredes tendrá los CIDR 10.0.1.0/24 y 10.0.2.0/24 respectivamente.
- La subred pública se llamará "vpc\_publica\_test" y la privada "vpc\_privada\_test".
- Se creará una tabla de enrutamiento que permita salida a Internet a la subred pública, pero no a la privada.

Ficha técnica:

- Nombre VPC: vpc\_< tus iniciales >
- Subred pública: vpc\_publica\_test
- Subred privada: vpc\_privada\_test
- CIDR VPC: 10.0.0.0/16
- CIDR público: 10.0.1.0/24
- CIDR privado: 10.0.2.0/24

Salida a Internet (subred pública):



Lo primero es iniciar el Learner Lab del entorno de AWS.

The screenshot shows the AWS Learner Lab interface. The main area displays a large blue 'V' logo. On the right side, there is a sidebar with the title 'Learner Lab'. Below the title, there are several links: [Environment Overview](#), [Environment Navigation](#), [Access the AWS Management Console](#), [Region restrictions](#), [Service usage and other restrictions](#), [Using the terminal in the browser](#), [Running AWS CLI commands](#), [Using the AWS SDK for Python](#), [Preserving your budget](#), [Accessing EC2 instances](#), [SSH Access to EC2 instances](#), [SSH Access from Windows](#), and [SSH Access from a Mac](#). Below these links, there is an 'Environment Overview' section with text explaining the lab environment and its long-lived nature. At the bottom of the sidebar, there is an 'Environment Navigation' section.

Una vez dentro del panel de control, buscamos "vpc" en la barra de búsqueda y clicamos en la opción que nos llevará al menú principal para poder trabajar con las VPC.



Entre las opciones que ofrece este panel de control, debemos elegir primero la de crear una nueva VPC.



Nos encontramos con una serie de ajustes que podemos cambiar o dejar de forma predeterminada. Para empezar, marcamos la opción "VPC only" de la izquierda, ya que por ahora solo queremos crear una VPC básica.

Se le asigna un nombre ("vpc\_< tusiniciales >"). Se establece el CIDR que nos pide el ejercicio y, por último, añadimos una etiqueta "AWSEvaluation" con valor "1daw" (esto es solo un ejemplo, la etiqueta puede ser cualquiera, simplemente debe ser identificable).

**VPC settings**

Resources to create [Info](#)  
Create only the VPC resource or the VPC and other networking resources.

☒ VPC only ☐ VPC and more

Name tag - *optional*  
Creates a tag with a key of 'Name' and a value that you specify.

vpc\_cms

IPv4 CIDR block [Info](#)  
☒ IPv4 CIDR manual input ☐ IPAM-allocated IPv4 CIDR block

IPv4 CIDR  
10.0.0.0/16

Esto último lo haremos siempre que se presente la opción de añadir una nueva etiqueta para poder encontrar los elementos de forma más sencilla en caso de ser necesario.

**Tags**

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	vpc_cms	Remove
AWSEvaluation	1daw	Remove

[Add new tag](#)

You can add 48 more tags.

Hacemos clic en el botón "Create VPC" y podemos ver cómo la VPC se ha creado correctamente con los ajustes definidos.

You successfully created vpc-09555c98614ee63ff / vpc\_cms

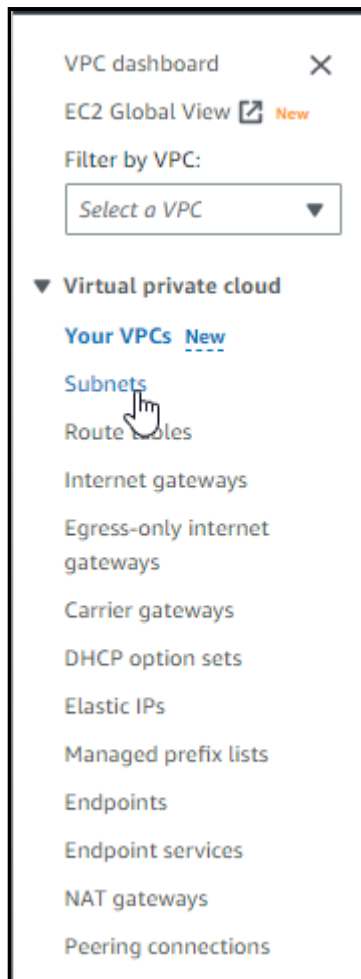
VPC > Your VPCs > vpc-09555c98614ee63ff

vpc-09555c98614ee63ff / vpc\_cms

**Details** [Info](#)

VPC ID vpc-09555c98614ee63ff	State <span>Available</span>	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-080c8b9907a7f43ac	Main route table rtb-06b16fec18bd21985	Main network ACL acl-0160feb37706714c3
Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool -	IPv6 CIDR (Network border group) -
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups <span>Failed to load rule groups</span>	Owner ID 808925875888	

El siguiente paso es dirigirse al panel lateral de la izquierda para entrar en el apartado "Subnets".



Una vez dentro, localizamos la opción "Create subnet", en la esquina superior derecha del menú. Hacemos clic en ella.

Se nos presenta un menú de creación similar al anterior para crear la subred nueva.

En el desplegable que aparece en el primer cuadro se debe seleccionar la VPC que creamos anteriormente:



De esta forma hemos vinculado la nueva subred que vamos a crear a la VPC con la que trabajaremos en este ejercicio.

En el siguiente cuadro aparecen nuevos ajustes. Debemos establecer un nombre para la nueva subred (tendremos que crear dos subredes, por lo que esta primera se llamará "vpc\_publica\_test").

Es importante que la zona de disponibilidad sea "US East (N. Virginia) / ...", ya que es donde se localiza la versión de AWS que tenemos disponible con nuestro correo corporativo.

Adicionalmente, se le asigna un rango CIDR a la subred (el cual lo da el enunciado del ejercicio) y una etiqueta nueva para tener la posibilidad de localizar este elemento de forma fácil y rápida si es necesario.

El rango CIDR para la subred pública es 10.0.1.0/24. Se pone la misma etiqueta en todos los elementos.

**Subnet settings**  
Specify the CIDR blocks and Availability Zone for the subnet.

**Subnet 1 of 1**

**Subnet name**  
Create a tag with a key of 'Name' and a value that you specify.  
vpc\_publica\_test  
The name can be up to 256 characters long.

**Availability Zone** [Info](#)  
Choose the zone in which your subnet will reside, or let Amazon choose one for you.  
US East (N. Virginia) / us-east-1a

**IPv4 CIDR block** [Info](#)  
10.0.1.0/24

**Tags - optional**

Key	Value - optional	
Name	vpc_publica_test	Remove
AWEvaluation	1daw	Remove

Terminamos de crear la nueva subred y esto nos lleva al panel de control de las subredes, donde podemos ver que la subred se ha creado con éxito.

You have successfully created 1 subnet: subnet-01549b01ed8c7782c

**Subnets (1)** [Info](#)

Filter subnets

Subnet ID: subnet-01549b01ed8c7782c Clear filters

	Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available IPv4 addresses	Availability Zone	Availability Zone ID	Name tags
<input type="checkbox"/>	vpc_publica_test	subnet-01549b01ed8c7782c	Available	vpc-09555c98614ee63ff	10.0.1.0/24	--	251	us-east-1a	us-east-1a	us-east-1a

Ahora es necesario realizar el mismo proceso para la subred privada, pero cambiando los valores pertinentes.

Se vuelve a seleccionar la VPC que creamos anteriormente en el desplegable del primer cuadro y se establece el nombre correspondiente a esta segunda subred, su rango CIDR (en este caso, 10.0.2.0/24) y su zona de disponibilidad, la cual es distinta de la anterior subred (esta zona de disponibilidad es la que termina en "us-east-1b").

## VPC

VPC ID  
Create subnets in this VPC.

vpc-09555c98614ee63ff (vpc\_cms)

### Associated VPC CIDRs

IPv4 CIDRs

10.0.0.0/16

## Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

### Subnet 1 of 1

Subnet name  
Create a tag with a key of 'Name' and a value that you specify.

vpc\_privada\_test

The name can be up to 256 characters long.

Availability Zone [Info](#)  
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

US East (N. Virginia) / us-east-1b

IPv4 CIDR block [Info](#)

10.0.2.0/24

▼ Tags - optional

Key	Value - optional	
Name	vpc_privada_test	Remove
AWSEvaluation	1daw	Remove

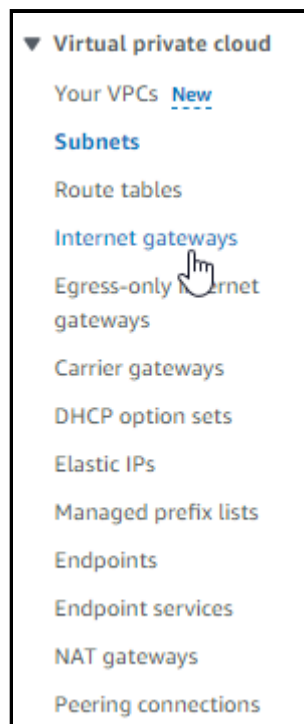
[Add new tag](#)

You can add 48 more tags.

[Remove](#)

[Add new subnet](#)

Una vez creadas las dos subredes, volvemos al menú lateral izquierdo para entrar en el menú "Internet gateways". La subred pública debe poder salir a Internet si es necesario, para ello hace falta una puerta de enlace.



Seguimos el mismo proceso inicial que en los demás paneles de control y hacemos clic en el botón "Create internet gateway" en la esquina superior derecha.

Establecemos un nombre para la nueva puerta de enlace y la asignamos la misma etiqueta de siempre. El nombre puede seguir el patrón "gw\_vpc\_< tusiniciales >", por ejemplo, aunque se le puede poner cualquier nombre (preferiblemente reconocible).

A screenshot of the 'Internet gateway settings' form in the AWS console. The form has a title 'Internet gateway settings'. Below it, there's a section for 'Name tag' with a description 'Creates a tag with a key of 'Name' and a value that you specify.' and a text input field containing 'gw\_vpc\_cms'. Below that is a section for 'Tags - optional' with a description '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 your resources or track your AWS costs.' This section contains a table with two columns: 'Key' and 'Value - optional'. There are two rows of tags. The first row has 'Name' as the key and 'gw\_vpc\_cms' as the value, with a 'Remove' button. The second row has 'AWSEvaluation' as the key and '1daw' as the value, also with a 'Remove' button.

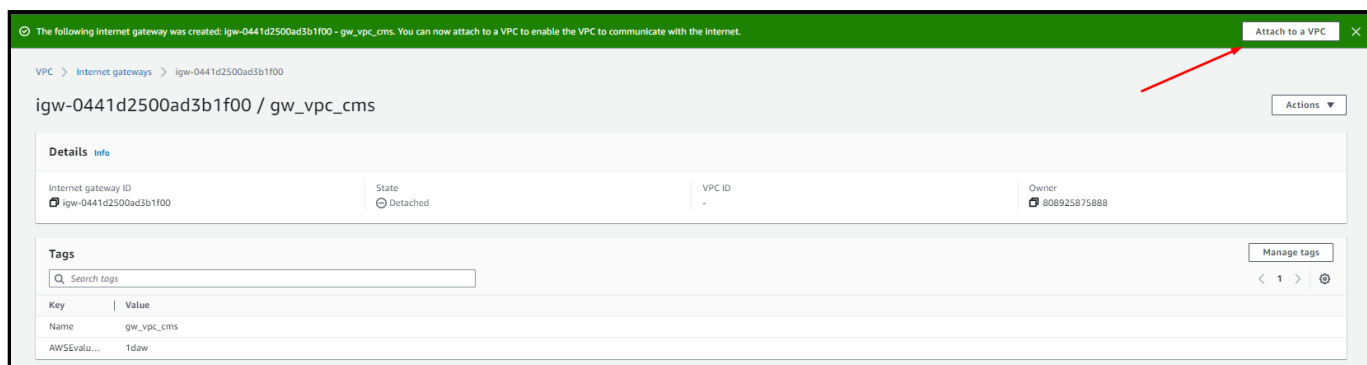
Clicamos en el botón de creación para terminar de crear el gateway.

Al crearse, salimos automáticamente al panel de control, y más concretamente al apartado del propio gateway que se acaba de crear.

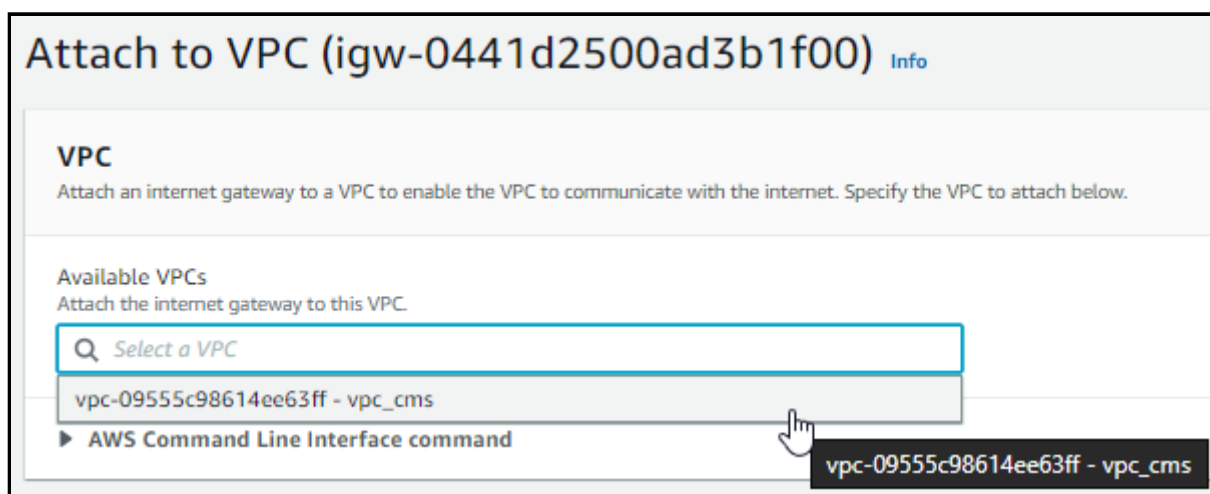
Ahora es necesario vincular esta puerta de enlace recién creada con la VPC que creamos anteriormente. Para ello debemos hacer clic en el botón "Attach to a VPC" que aparece en la esquina superior derecha justo



después de terminar de crear el gateway.



Una vez dentro, simplemente seleccionamos nuestro VPC en el desplegable que aparece en el cuadro de ajustes.



Clicamos en el botón de confirmación y se vincula la puerta de enlace con la VPC.

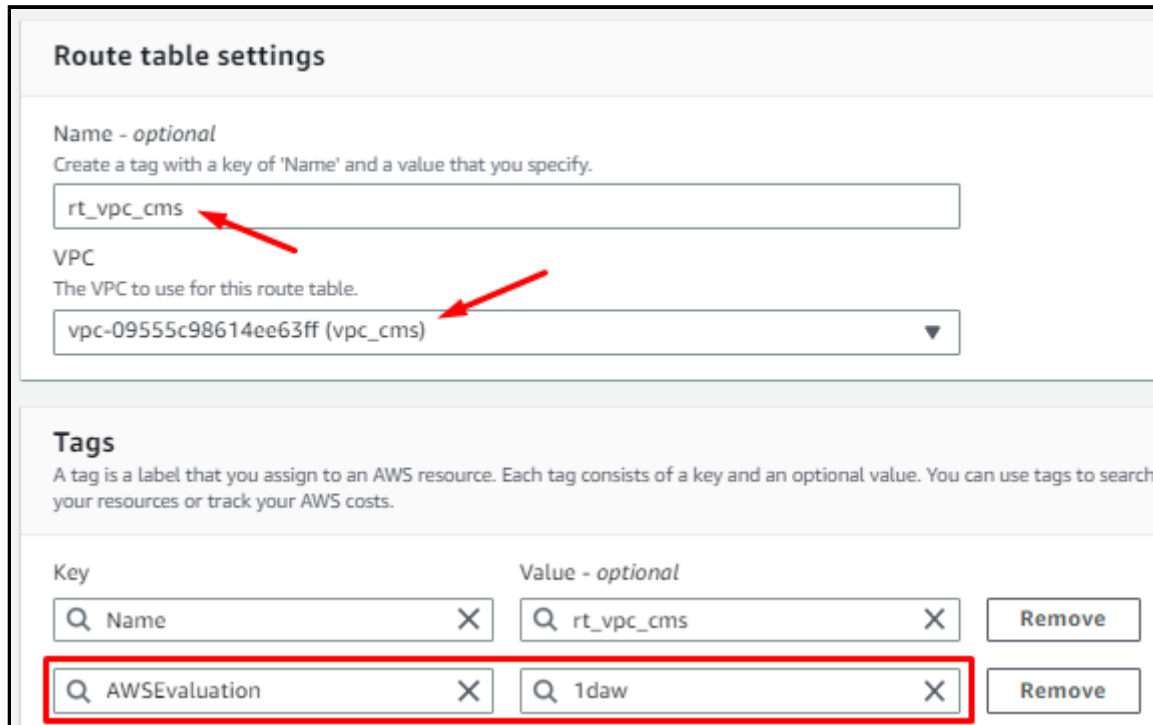
Una vez más, vamos al menú lateral izquierdo, esta vez a la opción "Route tables".



Como venimos haciendo en cada menú, hacemos clic en el botón "Create route table" en la esquina superior derecha.

Para el nombre de la nueva tabla de enrutamiento podemos seguir el mismo patrón que utilizamos anteriormente, el cual es "rt\_vpc\_< tusiniciales >".

Seleccionamos de nuevo la VPC a la que se añade esta nueva tabla de enrutamiento en el desplegable, y por último, añadimos la etiqueta que utilizamos en todos los elementos.



**Route table settings**

**Name - optional**  
Create a tag with a key of 'Name' and a value that you specify.

rt\_vpc\_cms

**VPC**  
The VPC to use for this route table.

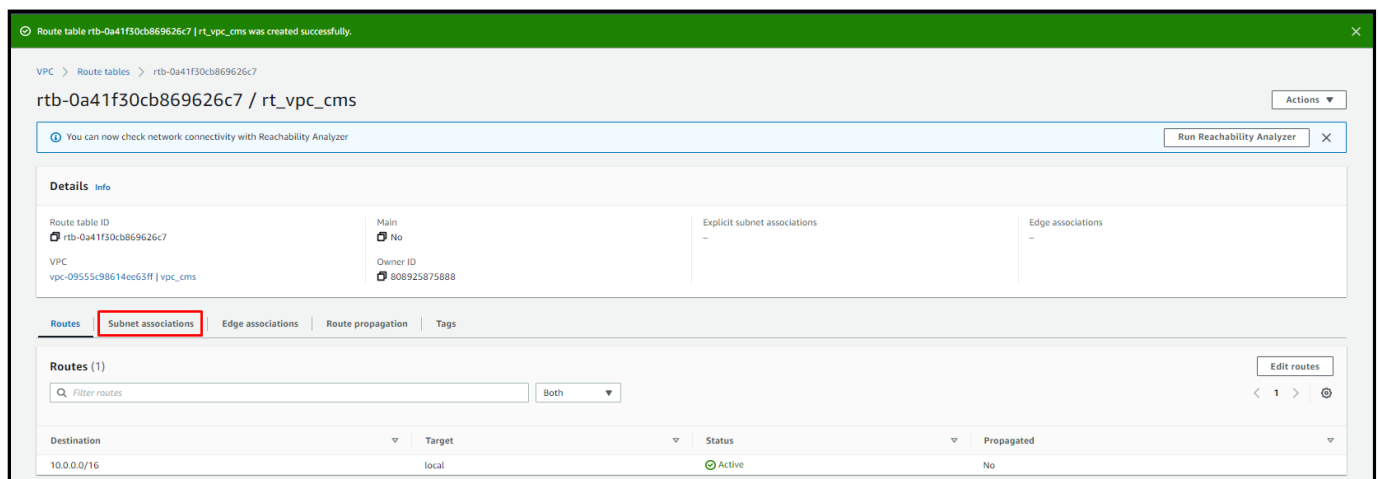
vpc-09555c98614ee63ff (vpc\_cms)

**Tags**  
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 your resources or track your AWS costs.

Key	Value - optional	
Name	rt_vpc_cms	Remove
AWSEvaluation	1daw	Remove

Una vez hacemos clic en el botón de confirmación, aparece un menú similar al de siempre que se crea un nuevo elemento.

En este nuevo menú, vemos una pestaña llamada "Subnet associations". Entramos en dicha pestaña.



Route table rtb-0a41f30cb869626c7 | rt\_vpc\_cms was created successfully.

VPC > Route tables > rtb-0a41f30cb869626c7

rtb-0a41f30cb869626c7 / rt\_vpc\_cms

You can now check network connectivity with Reachability Analyzer. [Run Reachability Analyzer](#)

**Details**

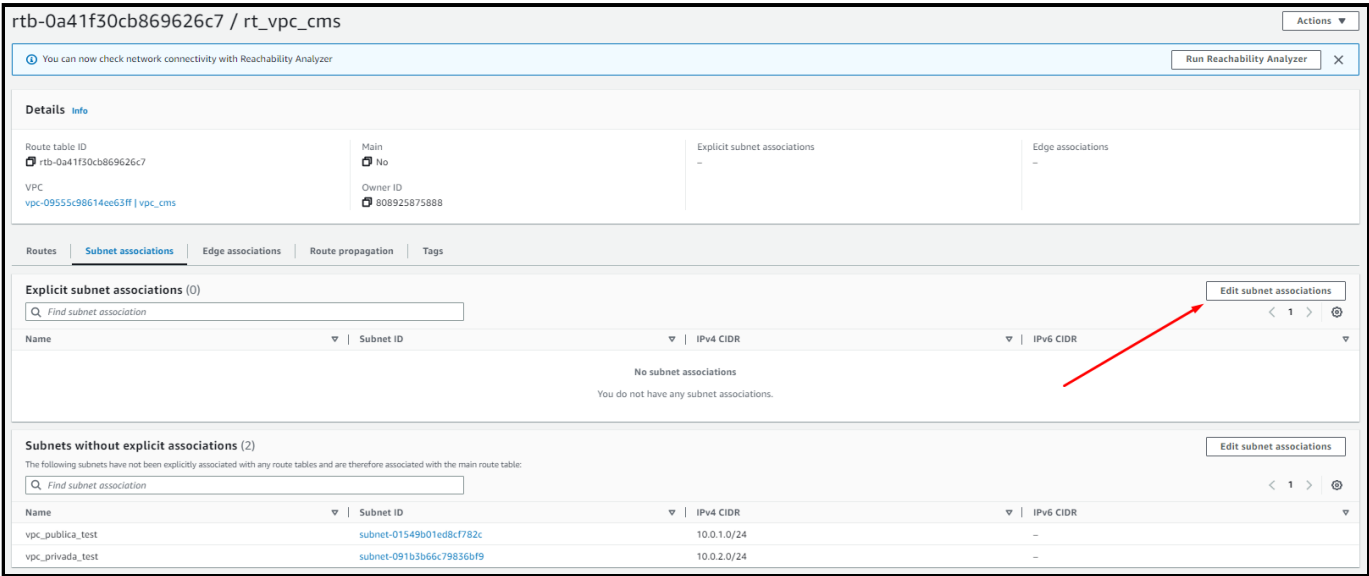
Route table ID rtb-0a41f30cb869626c7	Main No	Explicit subnet associations -	Edge associations -
VPC vpc-09555c98614ee63ff   vpc_cms	Owner ID 808925875888		

**Subnet associations**

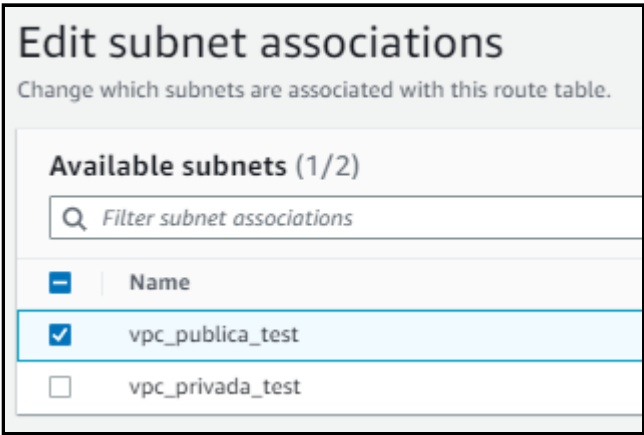
**Routes (1)**

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

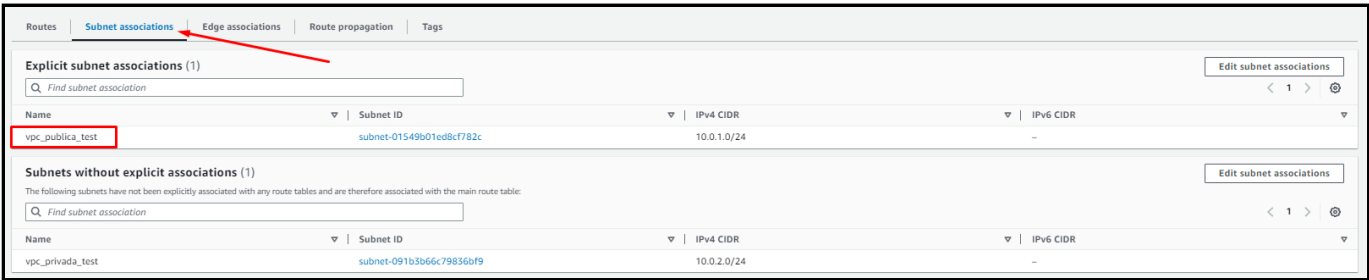
Vemos 2 cuadros llamados "Explicit subnet associations" (arriba) y "Subnets without explicit associations" (abajo). En el cuadro de arriba, clicamos en el botón "Edit subnet associations", para poder asociar la subred pública que creamos anteriormente a esta nueva tabla de enrutamiento.



Nos aparece una lista de las subredes que tenemos creadas. Simplemente seleccionamos la subred pública y hacemos clic en el botón de confirmación.

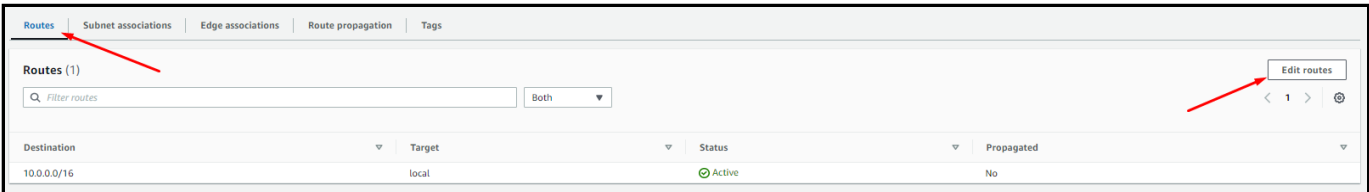


Una vez hecho, si volvemos a la pestaña "Subnet associations", podemos ver que en el cuadro de arriba ya aparece la subred que acabamos de asociar a esta tabla de enrutamiento.



Ahora cambiamos de pestaña a la principal, llamada "Routes".

En ella podemos ver otro cuadro con un botón en su esquina superior derecha que dice "Edit routes". Clicamos en él.



Lo que tenemos que hacer en este menú de ajustes es añadir una ruta nueva conocida como es la 0.0.0.0/0.

Esta ruta se encarga de que, en caso de que la dirección con la que se quiere establecer conexión no se encuentre entre el rango de IPs locales establecidas en la ruta justo encima (es decir, la comunicación no se va a establecer con un equipo local, sino con uno fuera de la red interna, un equipo de Internet), se establecerá comunicación con una IP cualquiera, que no pertenece a la red local.

Por lo tanto, en el primer recuadro, se establece el rango 0.0.0.0/0 y, en el segundo recuadro (el cual es un desplegable), debemos seleccionar la opción "Internet Gateway", la cual mostrará los gateways disponibles, entre los que estará el gateway que creamos anteriormente.

The image shows two screenshots from the AWS VPC console. The top screenshot shows the 'Target' dropdown menu with 'Internet Gateway' selected. The bottom screenshot shows the 'Destination' and 'Target' fields with '0.0.0.0/0' and 'igw-0441d250ad3b1f00' respectively.

**Target Dropdown Menu:**

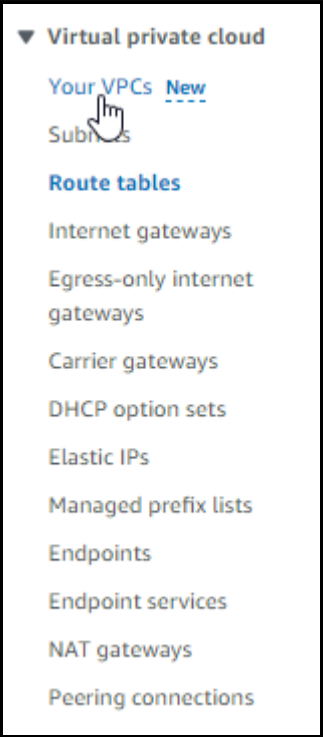
- Carrier Gateway
- Core Network
- Egress Only Internet Gateway
- Gateway Load Balancer Endpoint
- Instance
- Internet Gateway**
- local
- NAT Gateway
- Network Interface
- Outpost Local Gateway
- Peering Connection
- Transit Gateway
- Virtual Private Gateway

**Destination and Target Fields:**

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-0441d250ad3b1f00

Guardamos los cambios.

Para terminar con la creación de la VPC, volvemos al menú lateral izquierdo, a la opción llamada "Your VPCs".



Nos aparece una lista de las VPCs que tenemos disponibles. Hacemos clic en la que acabamos de terminar de crear.

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route table	Main network ACL	Tenancy	Default VPC
-	vpc-0319e0fe077aa36dc	Available	172.31.0.0/16	-	dopt-080e8bf907a7f4...	rtb-0cb03a7a8d4632994	acl-0dfb9ccfa64550851	Default	Yes
<input checked="" type="checkbox"/> vpc_cms	vpc-09555c98614ee63ff	Available	10.0.0.0/16	-	dopt-080e8bf907a7f4...	rtb-06b16fec18bd21985	acl-0160feb377067f4c3	Default	No

Dentro de ella podemos volver a ver el resumen de la VPC configurada al completo, donde ahora aparece la tabla de enrutamiento principal, entre otras cosas:

**Details**

VPC ID: vpc-09555c98614ee63ff

State: Available

DHCP option set: dopt-080e8bf907a7f43ac

IPv4 CIDR: 10.0.0.0/16

Main route table: rtb-06b16fec18bd21985

DNS hostnames: Disabled

DNS resolution: Enabled

Owner ID: 808925875888

**Resource map**

VPC: vpc\_cms

Subnets (2): us-east-1a, vpc\_publica\_test, us-east-1b, vpc\_privada\_test

Route tables (2): rt\_vpc\_cms, rtb-06b16fec18bd21985

Network connections (1): gw\_vpc\_cms

2.b. Una vez creada la VPC, vamos a crear una instancia EC2 en la red pública.

Detalles de la instancia a crear:

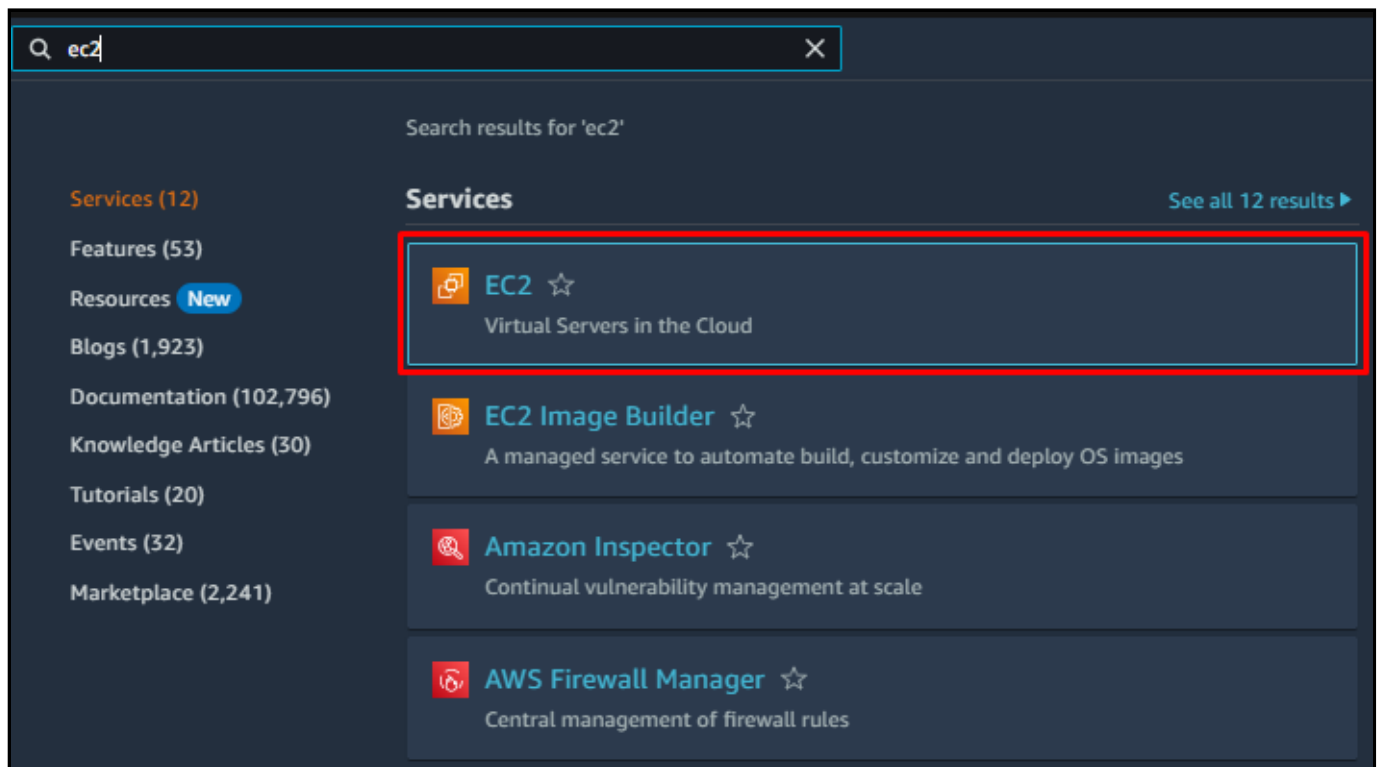
- Nombre: Ubuntu\_WebServer

- **Capa gratuita:** instancia tipo "t2.micro", VPC: vpc\_< tusiniciales >, subred: vpc\_publica\_test, IP local fija asignada: 10.0.1.10/24, grupo de seguridad: sg\_ubuntu
- **Acceso mediante vockey con el usuario < tunombre >**

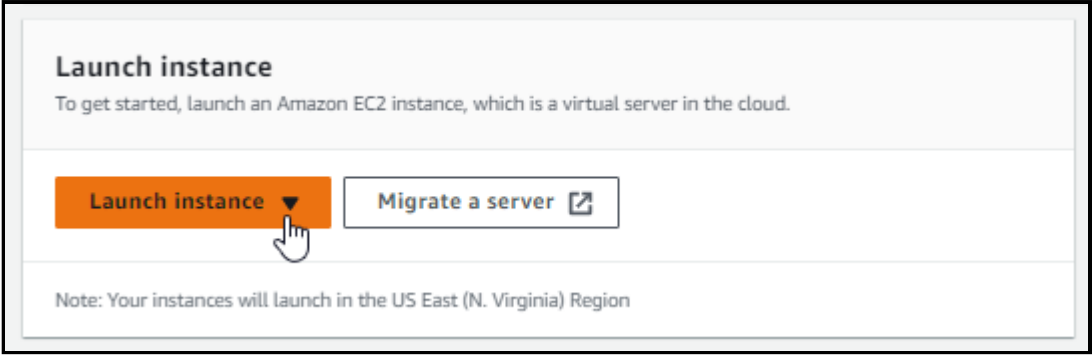
En este equipo se debe:

- **Crear una carpeta en el workdir (/home/user) con el nombre "miweb".**
- **Instalar Docker.** Añade al usuario < tunombre > en el grupo "docker" para poder lanzar comandos Docker.
- **Copiar el contenido de una página web estática que tengas creada usando el correspondiente comando scp desde tu equipo y usuario Ubuntu.**
- **Lanzar la instrucción "docker run", una vez tengamos el contenido web en "/home/user/miweb", para ejecutar un contenedor llamado "webserver" que ponga en contacto el directorio "/home/user/miweb" con el directorio de publicación por defecto de nginx. La imagen a usar será "nginx:1.23.3-alpine-slim". Comprueba que al visitar en un navegador "http://< ippublicaserver >" se visualiza el contenido de tu web.**
- **Demostrar que, si se modifica en el equipo local la página web y se vuelve a transferir con scp este contenido a "/home/user/miweb", al visitar en cualquier navegador la página web "http://< ippublicaserver >" se visualizan los cambios.**

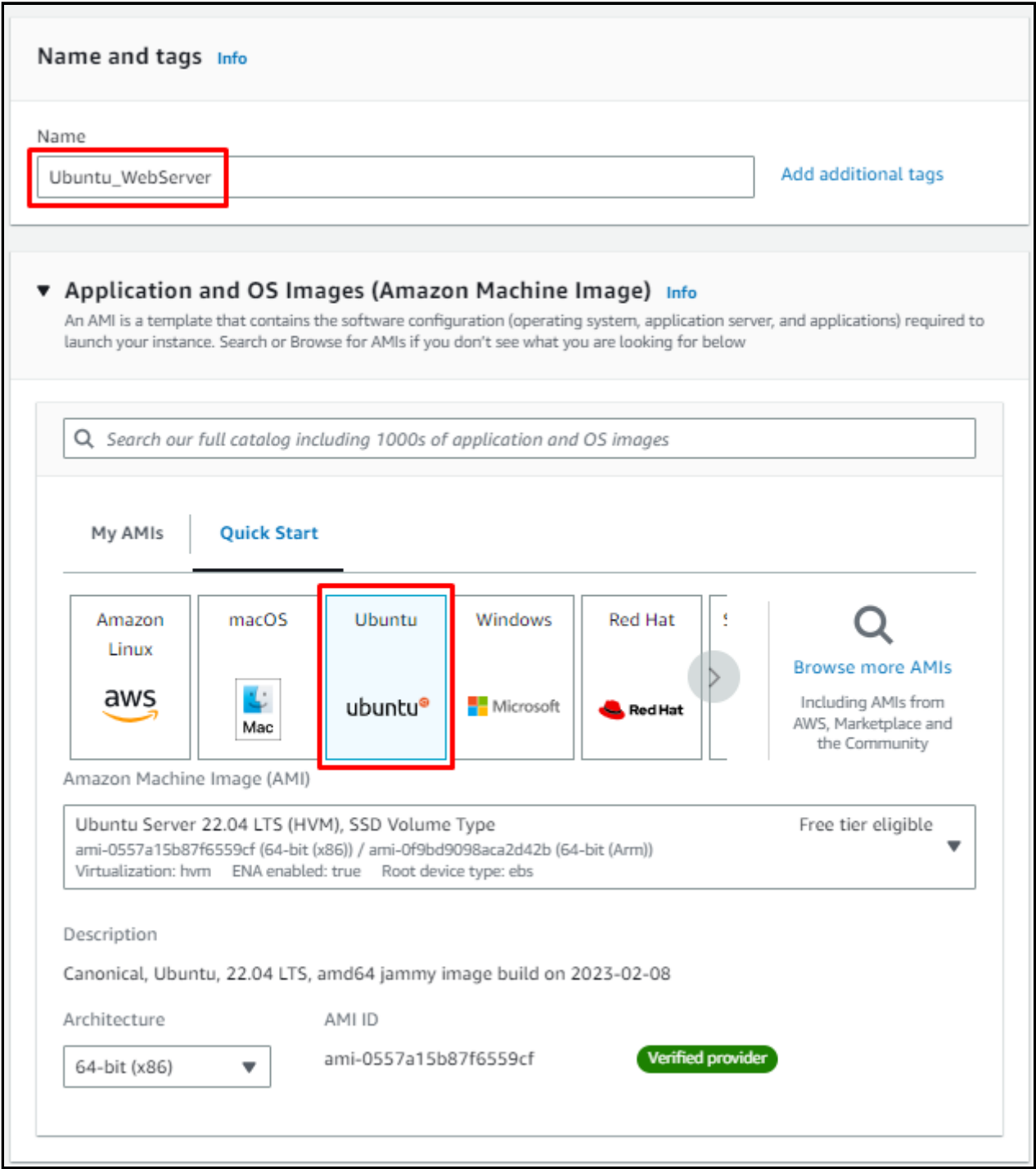
El comienzo es similar al ejercicio anterior: buscamos "ec2" en la barra de búsqueda del panel de control y hacemos clic en el menú.



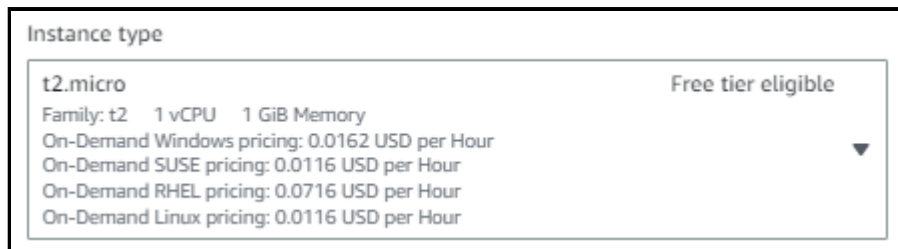
Creamos una nueva instancia:



Establecemos el nombre de la instancia y elegimos la imagen de sistema operativo que se instalará en ella.



El tipo de instancia lo dejamos con la opción predeterminada:



Seleccionamos la vockey en el siguiente desplegable, para indicar que el acceso a esta instancia se realizará mediante este método.



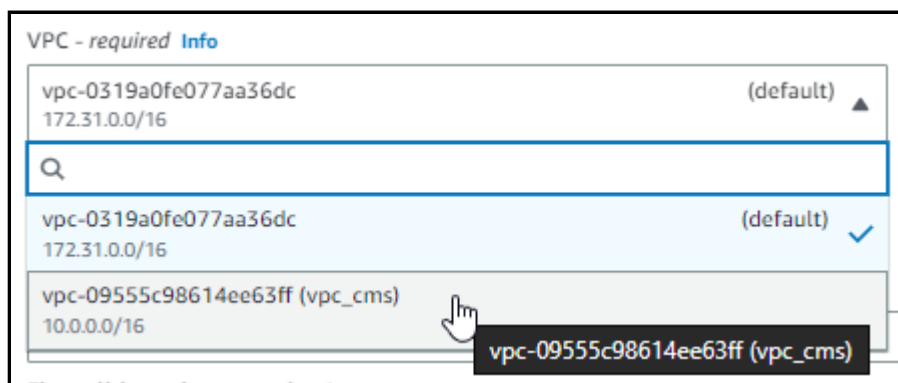
El siguiente cuadro se refiere a la configuración de la red de la nueva instancia.

Antes de hacer nada en este cuadro, bajamos un poco y activamos las casillas para permitir el tráfico mediante protocolos HTTP y HTTPS.



Ahora debemos hacer clic en el botón de edición que aparece en la esquina superior derecha de este cuadro para cambiar algunos ajustes.

Lo primero ahora es cambiar la VPC que utilizará en el primer desplegable que aparece:



En el siguiente desplegable que aparece se selecciona la subred a la que se conecta la instancia (recordemos que tenemos una subred pública y otra privada).

Tenemos que asegurarnos de que la subred que está seleccionada en este desplegable es la pública, tal y como especifica el enunciado de este ejercicio.



**Subnet** [Info](#)

subnet-01549b01ed8cf782c

vpc\_publica\_test

VPC: vpc-09555c98614ee63ff   Owner: 808925875888   Availability Zone: us-east-1a   ▼

IP addresses available: 251   CIDR: 10.0.1.0/24)

También hay que habilitar la siguiente opción, que permite que la máquina obtenga una IP pública:

**Auto-assign public IP** [Info](#)

Enable ▼

El siguiente ajuste sirve para crear un grupo de seguridad que, como especifica el enunciado de este ejercicio, debe ser "sg\_ubuntu".

**Firewall (security groups)** [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group

☐ Select existing security group

Security group name - *required*

sg\_ubuntu

Justo después aparece la opción de crear reglas para el grupo de seguridad, las cuales se deben dejar tal cual vienen de forma predeterminada, ya que habilita el puerto para el protocolo ssh.

Por último en este cuadro de configuración de la red de la instancia, tendremos que abrir el último desplegable para cambiar un ajuste extra.

### Inbound security groups rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Remove

Type [Info](#)  
ssh ▼

Protocol [Info](#)  
TCP

Port range [Info](#)  
22

Source type [Info](#)  
Anywhere ▼

Source [Info](#)  

Add CIDR, prefix list or security

0.0.0.0/0 ✕

Description - optional [Info](#)  
e.g. SSH for admin desktop

▼ Security group rule 2 (TCP, 443, 0.0.0.0/0)

Remove

Type [Info](#)  
HTTPS ▼

Protocol [Info](#)  
TCP

Port range [Info](#)  
443

Source type [Info](#)  
Anywhere ▼

Source [Info](#)  

Add CIDR, prefix list or security

0.0.0.0/0 ✕

Description - optional [Info](#)  
e.g. SSH for admin desktop

▼ Security group rule 3 (TCP, 80, 0.0.0.0/0)

Remove

Type [Info](#)  
HTTP ▼

Protocol [Info](#)  
TCP

Port range [Info](#)  
80

Source type [Info](#)  
Anywhere ▼

Source [Info](#)  

Add CIDR, prefix list or security

0.0.0.0/0 ✕

Description - optional [Info](#)  
e.g. SSH for admin desktop

⚠ 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. ✕

Add security group rule

► Advanced network configuration

Nada más abrirlo vemos que aparece una interfaz ya creada. El único ajuste que debemos tocar aquí es la IP primaria fija que se le asignará a la instancia, la cual es la 10.0.1.10, según especifica el enunciado de este ejercicio.

18 / 31

**Network interface 1**

Device index <a href="#">Info</a>	Network interface <a href="#">Info</a>	Description <a href="#">Info</a>
0	New interface	
Subnet <a href="#">Info</a>	Security groups <a href="#">Info</a>	Primary IP <a href="#">Info</a>
subnet-01549b01ed8cf782c	New security group	10.0.1.10
IP addresses available: 251		

En el siguiente cuadro, debemos establecer el tamaño del almacenamiento de la instancia, donde establecemos 30 GB de almacenamiento, que es el máximo espacio de almacenamiento gratuito que ofrece nuestro plan en AWS.

**▼ Configure storage** [Info](#) Advanced

1x  GiB  Root volume (Not encrypted)

[Free tier eligible customers can get up to 30 GB of EBS General Purpose \(SSD\) or Magnetic storage](#) ×

Una vez configurada la nueva instancia, podemos hacer clic en el botón de confirmación y esperar a que se inicie por primera vez la máquina.

**Success**  
Successfully initiated launch of instance (i-068ea8530ea8dd5b7)

**▼ Launch log**

Initializing requests	Succeeded
Creating security groups	Succeeded
Creating security group rules	Succeeded
Launch initiation	Succeeded

Si volvemos al panel de control de instancias de EC2 y hacemos clic en el token azul de la instancia que acabamos de crear, podemos ver un resumen de la configuración:

**Instance summary for i-068ea8530ea8dd5b7 (Ubuntu\_WebServer)** [Info](#)

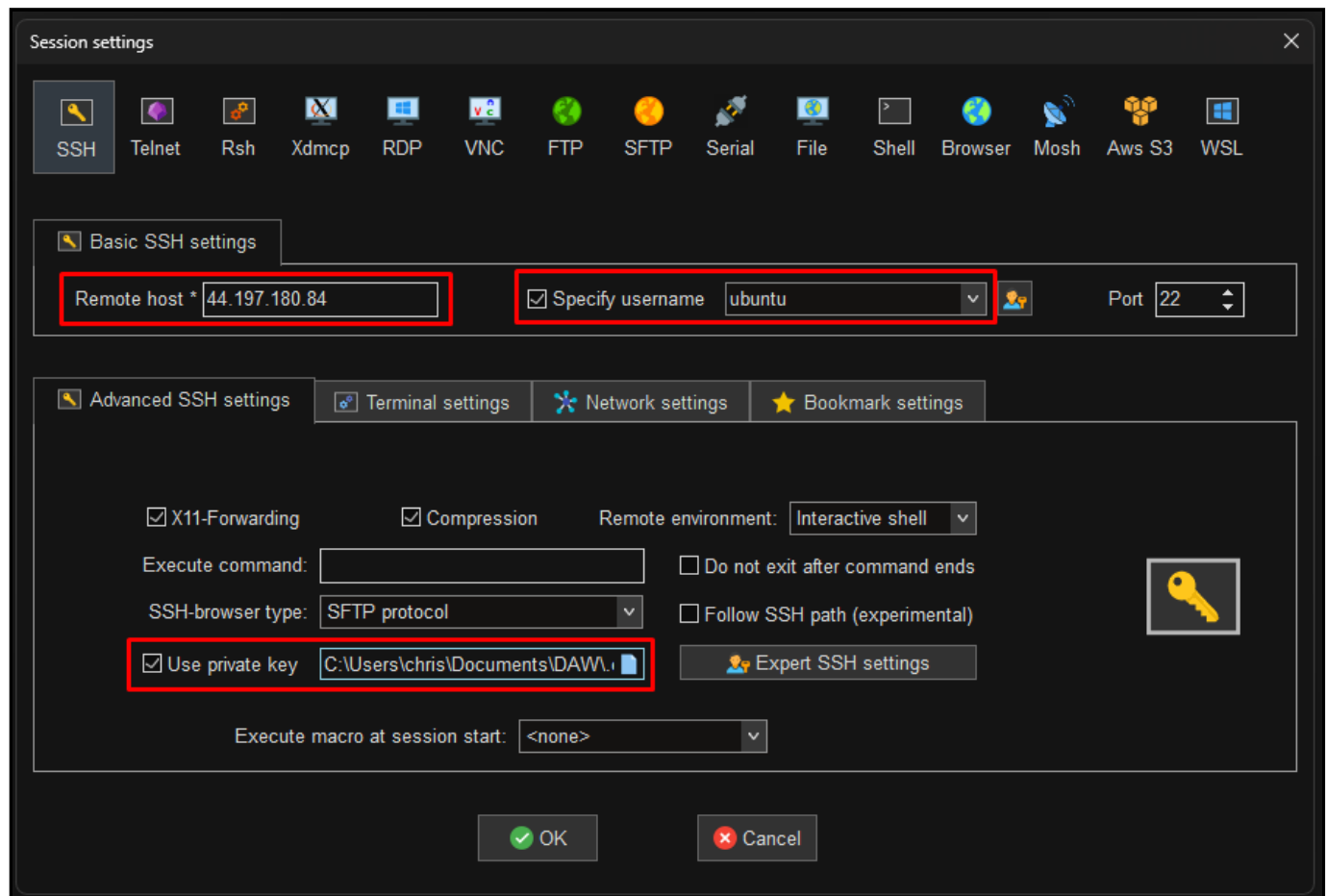
Updated less than a minute ago

Instance ID i-068ea8530ea8dd5b7 <a href="#">[Ubuntu_WebServer]</a>	Public IPv4 address 44.197.180.84   <a href="#">open address</a>	Private IPv4 addresses 10.0.1.10
IPv6 address -	Instance state Running	Public IPv4 DNS -
Hostname type IP name: ip-10-0-1-10.ec2.internal	Private IP DNS name (IPv4 only) ip-10-0-1-10.ec2.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	AWS Compute Optimizer finding <a href="#">Opt-in to AWS Compute Optimizer for recommendations.</a>   <a href="#">Learn more</a>
Auto-assigned IP address 44.197.180.84 (Public IP)	VPC ID vpc-09555c98614ee63ff (vpc_cms)	Auto Scaling Group name -
IAM Role -	Subnet ID subnet-01549b01ed8cf782c (vpc_publica_test)	

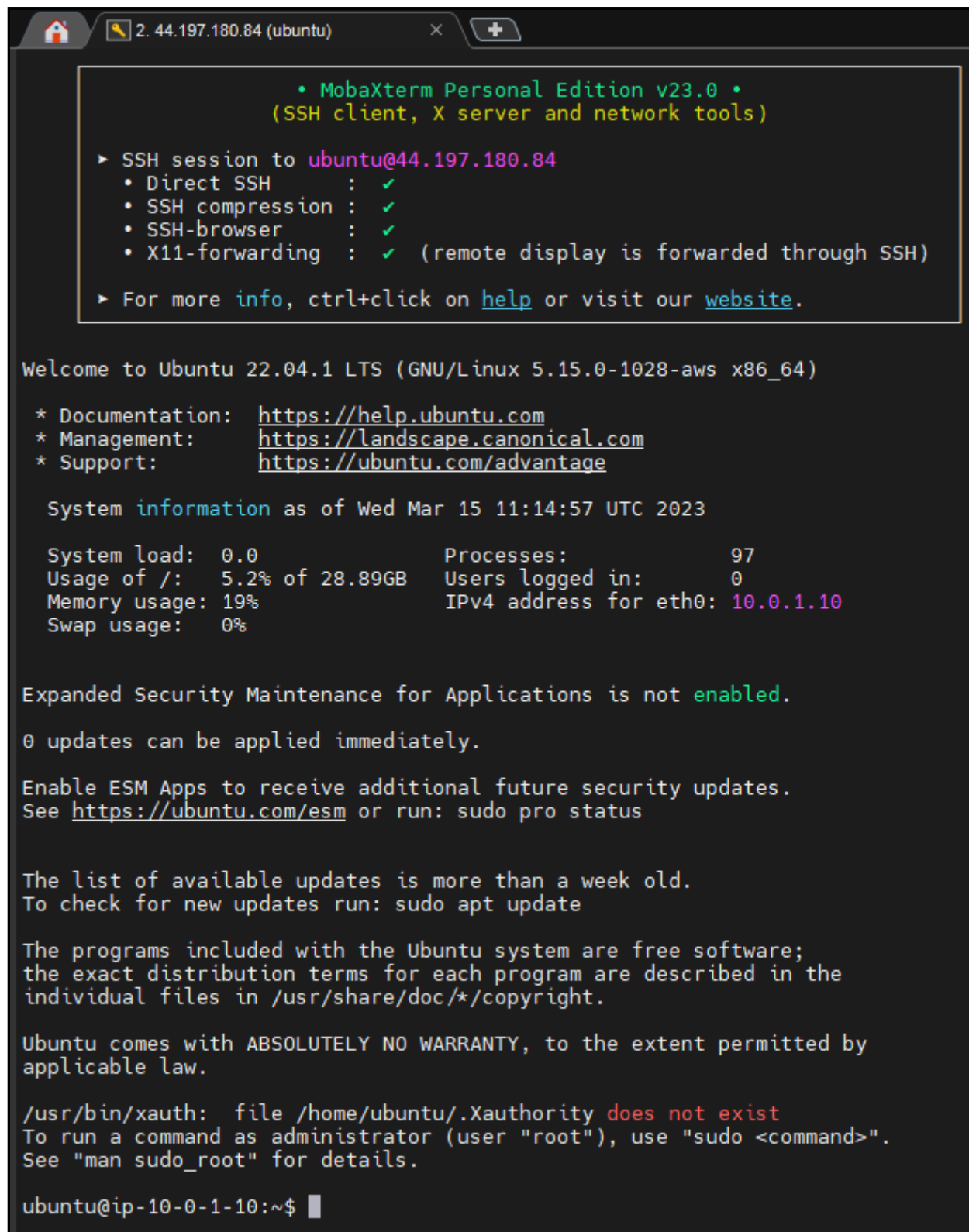
La siguiente parte de este ejercicio requiere que establezcamos conexión con la máquina para poder interactuar con ella.

Para ello, podemos utilizar clientes SSH como Putty o MobaXterm, por ejemplo.

En mi caso, utilizo MobaXterm (el proceso es muy similar y parecido en Putty). Creamos un nuevo perfil de conexión, indicamos el host al que queremos conectarnos (IP pública de la instancia, que aparece en el resumen que acabamos de ver), el usuario con el que se inicia sesión en la máquina (de forma predeterminada, Ubuntu crea el usuario "ubuntu") y la clave vockey con extensión "ppk" que autentifica la conexión



Una vez se autentifica todo sin problema, ya tenemos acceso al terminal de comandos de la máquina de Ubuntu.



```
• MobaXterm Personal Edition v23.0 •
(SSH client, X server and network tools)

► SSH session to ubuntu@44.197.180.84
  • Direct SSH      : ✓
  • SSH compression : ✓
  • SSH-browser     : ✓
  • X11-forwarding  : ✓ (remote display is forwarded through SSH)

► For more info, ctrl+click on help or visit our website.

Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1028-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Wed Mar 15 11:14:57 UTC 2023

System load:  0.0               Processes:           97
Usage of /:   5.2% of 28.89GB   Users logged in:    0
Memory usage: 19%              IPv4 address for eth0: 10.0.1.10
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

/usr/bin/xauth:  file /home/ubuntu/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-1-10:~$
```

Lo primero y más recomendable es ejecutar el siguiente comando nada más iniciar la máquina:

```
sudo apt update -y && sudo apt upgrade -y && sudo apt auto-remove -y
```

Este comando "pone al día" a la máquina, actualizando sus dependencias y llevando el sistema operativo a la última versión.

Si aparece algún cuadro con opciones, etc., simplemente se le da a ENTER (OK) sin tocar nada hasta que termine la actualización y volvamos a tener el control del terminal de forma normal.

Adicionalmente, con el comando "**ip a**" podemos ver la IP fija establecida en la configuración de la instancia antes de crearla.

```
ubuntu@ip-10-0-1-10:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc fq_codel state UP group default qlen 1000
    link/ether 02:33:17:ef:93:cb brd ff:ff:ff:ff:ff:ff
    inet 10.0.1.10/24 metric 100 brd 10.0.1.255 scope global dynamic eth0
        valid_lft 3473sec preferred_lft 3473sec
    inet6 fe80::33:17ff:feef:93cb/64 scope link
        valid_lft forever preferred_lft forever
```

Como especifica este ejercicio, el usuario a utilizar debe llamarse "< tunombre >", por lo que vamos a crearlo.

Para ello, debemos utilizar el siguiente comando:

```
sudo adduser < username >
```

```
ubuntu@ip-10-0-1-10:~$ sudo adduser christianmillan
Adding user `christianmillan' ...
Adding new group `christianmillan' (1001) ...
Adding new user `christianmillan' (1001) with group `christianmillan' ...
Creating home directory `/home/christianmillan' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for christianmillan
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
```

Lo siguiente es instalar Docker en esta máquina, lo cual se hace con el siguiente comando:

```
sudo apt install docker.io -y
```

Igual que antes (y siempre que pase en esta práctica), si aparecen cuadros con una serie de ajustes o advertencias, se hace clic en ENTER (OK) hasta que termine la instalación sin tocar ningún ajuste.

El ejercicio pide que el usuario personal recién creado se incluya en el grupo "docker", lo que permitirá a este usuario ejecutar comandos de Docker sin necesidad de escribir "sudo" delante de cada comando.

El cambio de grupo se hace con el siguiente comando:

```
sudo adduser < username > docker
```

```
ubuntu@ip-10-0-1-10:~$ sudo adduser christianmillan docker
Adding user `christianmillan' to group `docker' ...
Adding user christianmillan to group docker
Done.
```

De ahora en adelante, utilizaremos el usuario personal que ahora sí tiene acceso a los comandos de Docker. Para ello, debemos cambiar de usuario con el siguiente comando:

```
su < username >
```

```
ubuntu@ip-10-0-1-10:~$ su christianmillan
Password:
christianmillan@ip-10-0-1-10:/home/ubuntu$
```

Lo siguiente que pide este ejercicio es que creamos el directorio "miweb" en el directorio personal del usuario que hemos creado. Para hacer esto, ejecutamos el siguiente comando desde el home de nuestro usuario:

```
mkdir miweb
```

```
christianmillan@ip-10-0-1-10:~$ pwd
/home/christianmillan
christianmillan@ip-10-0-1-10:~$ mkdir miweb
```

El siguiente paso es transferir el contenido de una página web con contenido dentro del directorio "miweb", todo esto utilizando el subsistema de Ubuntu instalado en Windows.

Iniciamos el subsistema de Ubuntu en Windows.

```
christianms13@christianpc: ~ X + v
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

This message is shown once a day. To disable it please create the
/home/christianms13/.hushlogin file.
christianms13@christianpc:~$ |
```

Nos dirigimos al directorio donde se encuentra el contenido de la página web, en mi caso con el siguiente comando:

```
cd /mnt/c/Users/chris/Documents/DAW/lenguajes_de_marcas
```

Una vez dentro de este directorio, es necesitamos utilizar la vockey con extensión "pem" para transferir estos archivos. Para hacer esto, primero debemos cambiar los permisos del archivo "vockey.pem".

Lo primero es copiar el archivo desde el directorio donde lo tengamos guardado al directorio home del usuario del subsistema de Ubuntu. En mi caso el comando es el siguiente:

```
cp /mnt/c/Users/chris/Documents/DAW/.extra/clave\ aws/vockey.pem
/home/christianms13
```

Una vez tenemos el archivo en el home de Ubuntu, le cambiamos los permisos:

```
chmod 400 /home/christianms13/vockey.pem
```

```
christianms13@christianpc: /mnt/c/Users/chris/Documents/DAW/lenguajes_de_marcas/fin_1er_trimestre$ ls -la /home/christianms13/
total 32
drwxr-x--- 3 christianms13 christianms13 4096 Mar 15 23:47 .
drwxr-xr-x 3 root          root          4096 Mar 13 20:32 ..
-rw----- 1 christianms13 christianms13 861 Mar 15 19:37 .bash_history
-rw-r--r-- 1 christianms13 christianms13 220 Mar 13 20:32 .bash_logout
-rw-r--r-- 1 christianms13 christianms13 3771 Mar 13 20:32 .bashrc
-rw-r--r-- 1 christianms13 christianms13 0 Mar 15 19:07 .motd_shown
-rw-r--r-- 1 christianms13 christianms13 807 Mar 13 20:32 .profile
drwx----- 2 christianms13 christianms13 4096 Mar 15 19:25 .ssh
-rw-r--r-- 1 christianms13 christianms13 0 Mar 13 20:33 .sudo_as_admin_successful
-r----- 1 christianms13 christianms13 1678 Mar 15 23:47 vockey.pem
```

Lo suyo sería utilizar el comando scp para transferir los archivos directamente al directorio "miweb" en el home del usuario personal creado anteriormente en la instancia, pero después de intentarlo de muchas maneras siempre me da un error de public key.

La forma que he decidido utilizar es la siguiente:

Transferimos los archivos al directorio home del usuario "ubuntu" (usuario que se crea de forma predeterminada con la máquina) y después se mueven esos archivos al directorio "miweb" del usuario con nombre.

Para ello, primero ejecutamos la transferencia de archivos. En mi caso el comando es el siguiente:

```
scp -r -i "/home/christianms13/vockey.pem" fin_1er_trimestre/*
ubuntu@75.101.247.212:/home/ubuntu/
```

```
christianms13@christianpc: /mnt/c/Users/chris/Documents/DAW/lenguajes_de_marcas$ scp -r -i "/home/christianms13/vockey.pem" fin_1er_trimestre/* ubuntu@3.210.185.30:/home/ubuntu/
characters_style.css 100% 14KB 77.1KB/s 00:00
main_style.css       100% 38KB 266.0KB/s 00:00
media_style.css      100% 16KB 172.3KB/s 00:00
planets_style.css    100% 12KB 132.3KB/s 00:00
anakin.css           100% 12KB 128.3KB/s 00:00
finn.css              100% 12KB 127.8KB/s 00:00
han.css               100% 12KB 126.8KB/s 00:00
```

Y ahora desde el cliente SSH (MobaXterm, Putty, etc.), ejecutamos el comando "exit" para volver al usuario "ubuntu".

Por último, ya desde este usuario, ejecutamos el comando "mv" para trasladar todo el contenido transferido por scp al directorio "miweb". Mi comando se ve así:



```
sudo mv css/ font/ img/ index.html js/ pages/ readme/ /home/christianmillan/miweb/
```

```
ubuntu@ip-10-0-1-10:~$ ls -la
total 92
drwxr-x--- 11 ubuntu ubuntu 4096 Mar 15 23:48 .
drwxr-xr-x  4 root   root   4096 Mar 15 11:32 ..
-rw-r----- 1 ubuntu ubuntu  58 Mar 15 23:37 .Xauthority
-rw-r----- 1 ubuntu ubuntu 687 Mar 15 23:34 .bash_history
-rw-r--r--  1 ubuntu ubuntu 220 Jan  6 2022 .bash_logout
-rw-r--r--  1 ubuntu ubuntu 3771 Jan  6 2022 .bashrc
drwx----- 2 ubuntu ubuntu 4096 Mar 15 11:14 .cache
drwxrwxr-x  3 ubuntu ubuntu 4096 Mar 15 23:25 .local
-rw-r--r--  1 ubuntu ubuntu 807 Jan  6 2022 .profile
drwx----- 2 ubuntu ubuntu 4096 Mar 15 11:08 .ssh
-rw-r--r--  1 ubuntu ubuntu  0 Mar 15 11:19 .sudo_as_admin_successful
drwxrwxr-x  3 ubuntu ubuntu 4096 Mar 15 23:46 css
drwxrwxr-x 11 ubuntu ubuntu 4096 Mar 15 23:47 font
drwxrwxr-x 12 ubuntu ubuntu 4096 Mar 15 23:47 img
-rwxrwxr-x  1 ubuntu ubuntu 24758 Mar 15 23:47 index.html
drwxrwxr-x  2 ubuntu ubuntu 4096 Mar 15 23:47 js
drwxrwxr-x  3 ubuntu ubuntu 4096 Mar 15 23:47 pages
drwxrwxr-x  3 ubuntu ubuntu 4096 Mar 15 23:48 readme
ubuntu@ip-10-0-1-10:~$ sudo mv css/ font/ img/ index.html js/ pages/ readme/ /home/christianmillan/miweb/
ubuntu@ip-10-0-1-10:~$ sudo ls -la /home/christianmillan/miweb
total 60
drwxr-xr-x  8 root      root      4096 Mar 15 23:51 .
drwxr-x---  3 christianmillan christianmillan 4096 Mar 15 23:41 ..
drwxrwxr-x  3 ubuntu   ubuntu   4096 Mar 15 23:46 css
drwxrwxr-x 11 ubuntu   ubuntu   4096 Mar 15 23:47 font
drwxrwxr-x 12 ubuntu   ubuntu   4096 Mar 15 23:47 img
-rwxrwxr-x  1 ubuntu   ubuntu   24758 Mar 15 23:47 index.html
drwxrwxr-x  2 ubuntu   ubuntu   4096 Mar 15 23:47 js
drwxrwxr-x  3 ubuntu   ubuntu   4096 Mar 15 23:47 pages
drwxrwxr-x  3 ubuntu   ubuntu   4096 Mar 15 23:48 readme
```

Para seguir con el ejercicio, debemos utilizar la instrucción "**docker run**" para crear un contenedor nuevo que ponga en contacto el directorio "miweb" con el directorio de alojamiento de páginas web de nginx.

Para empezar, volvemos a iniciar sesión como el usuario personal en la instancia:

```
su < username >
```

```
ubuntu@ip-10-0-1-10:~$ su christianmillan
Password:
christianmillan@ip-10-0-1-10:/home/ubuntu$
```

Ahora ejecutamos el siguiente comando para crear el nuevo contenedor y alojar la página web directamente:

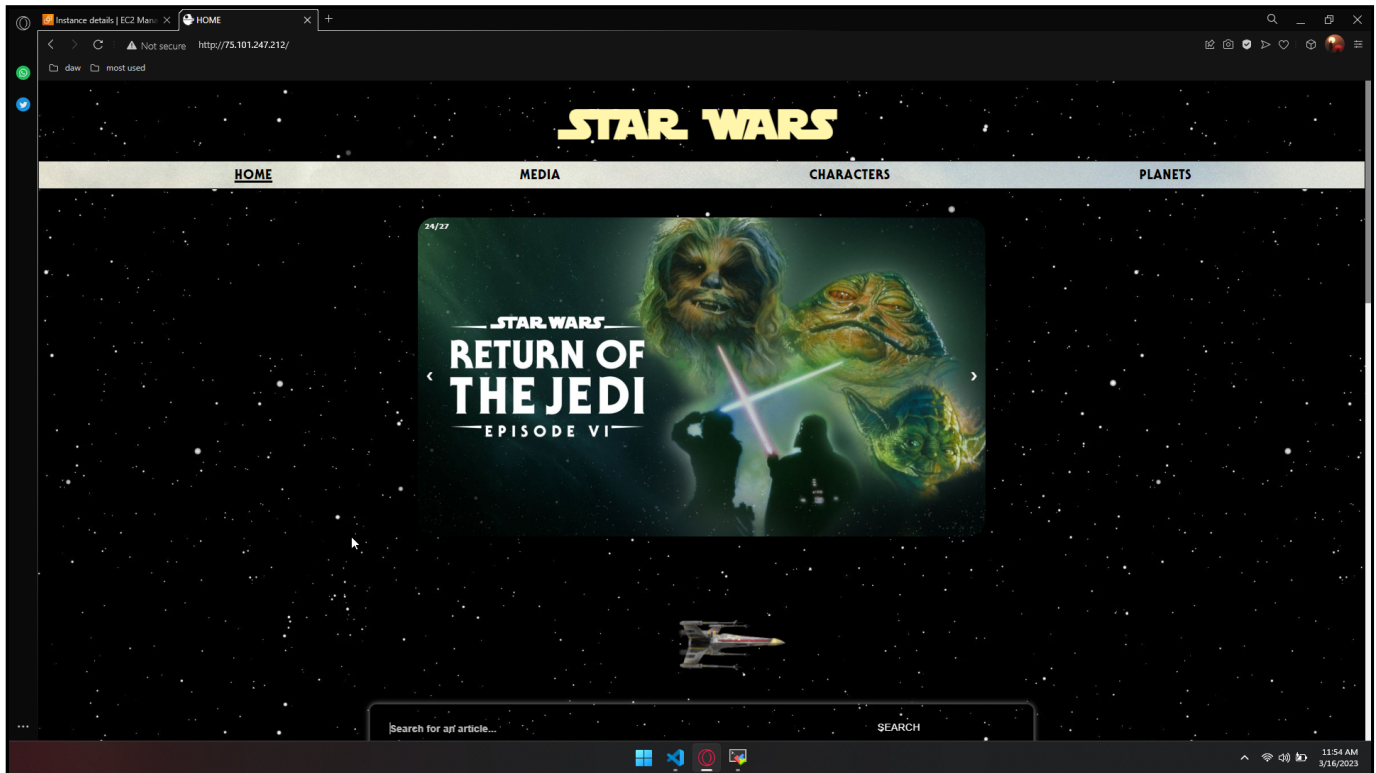
```
docker run --name webserver -d -p 80:80 -v ~/miweb:/usr/share/nginx/html
nginx:1.23.3-alpine-slim
```

```
christianmillan@ip-10-0-1-10:~$ docker run --name webserver -d -p 80:80 -v ~/miweb:/usr/share/nginx/html nginx:1.23.3-alpine-slim
Unable to find image 'nginx:1.23.3-alpine-slim' locally
1.23.3-alpine-slim: Pulling from library/nginx
63b65145d645: Pull complete
8c7e1fd96380: Pull complete
86c5246c96db: Pull complete
b874033c43fb: Pull complete
dbe1551bd73f: Pull complete
0d4f6b3f3de6: Pull complete
Digest: sha256:294dc03dddc4e1ae4062fa379b977ab49cf4f022dd4af371cde9015a0abb9c28
Status: Downloaded newer image for nginx:1.23.3-alpine-slim
2278b7cdf63387ce2b4933c766f6f939e0e4617f8b124900c718115c6ceaabcc
```

Y aquí podemos ver la lista de los contenedores:

```
christianmillan@ip-10-0-1-10:~$ docker container ls
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
2278b7cdf633   nginx:1.23.3-alpine-slim  "/docker-entrypoint..."  36 seconds ago  Up 35 seconds  0.0.0.0:80→80/tcp, :::80→80/tcp  webserver
```

Debido a que en la configuración de la instancia habilitamos el tráfico mediante protocolos HTTP, podemos visitar la página web que acabamos de alojar en un contenedor con nginx ingresando la url "http://<ippublicaserver>:80". En mi caso la url es "http://75.101.247.212:80" (en el momento de hacer este ejercicio, ya que cada vez que el Learner Lab se reinicia, se cambia la IP pública de las instancias), y se ve así:



Para comprobar que cualquier cambio que se realice en la página web se ve reflejado en la página en el navegador, debemos hacer lo siguiente:

Lo primero es realizar un cambio en el código de la página web. En mi caso, voy a cambiar el título que aparece en la página, el cual dice "STAR WARS" ahora mismo, para que ponga "EJERCICIO AWS Y DOCKER". Aquí está el código actualizado:

```

1  <!--
2  @author: Christian Millán Soria
3  description: end of first trimester project
4  -->
5
6  <html>
7  <head>
8  <title>HOME</title>
9  <meta charset="utf-8">
10 <meta name="viewport" content="width=device-width, initial-scale=1.0">
11
12 <!-- links -->
13 <!-- css -->
14 <link rel="stylesheet" href="css/main_style.css">
15 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/4.1.1/animate.min.css">
16
17 <!-- favicon -->
18 <link rel="icon" type="image/x-icon" href="https://icons.iconarchive.com/icons/sensibleworld/starwars/1024/Death-Star-icon.png">
19
20 <!-- fonts -->
21 <link rel="preconnect" href="https://fonts.googleapis.com">
22 <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
23 <link href="https://fonts.googleapis.com/css2?family=Bebas+Neue&display=swap" rel="stylesheet">
24 <link rel="preconnect" href="https://fonts.googleapis.com">
25 <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
26 <link href="https://fonts.googleapis.com/css2?family=Righteous&display=swap" rel="stylesheet">
27 <link rel="preconnect" href="https://fonts.googleapis.com">
28 <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
29 <link href="https://fonts.googleapis.com/css2?family=Rajdhani:wght@700&display=swap" rel="stylesheet">
30 <link rel="preconnect" href="https://fonts.googleapis.com">
31 <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
32 <link href="https://fonts.googleapis.com/css2?family=Russo+One&display=swap" rel="stylesheet">
33
34 <!-- script -->
35 <script type="text/javascript" defer src="js/main.js"></script>
36 <script type="text/javascript" defer src="js/search.js"></script>
37 <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
38
39 <!-- imported -->
40 <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
41 </head>
42
43 <body>
44 <!-- title -->
45 <h1 id="title">EJERCICIO AWS Y DOCKER</h1>
46
47 <!--
48 You, 2 months ago * cms ...

```

Ahora volvemos a ejecutar el comando scp para subir los cambios al servidor Ubuntu (esta vez solo el archivo "index.html").

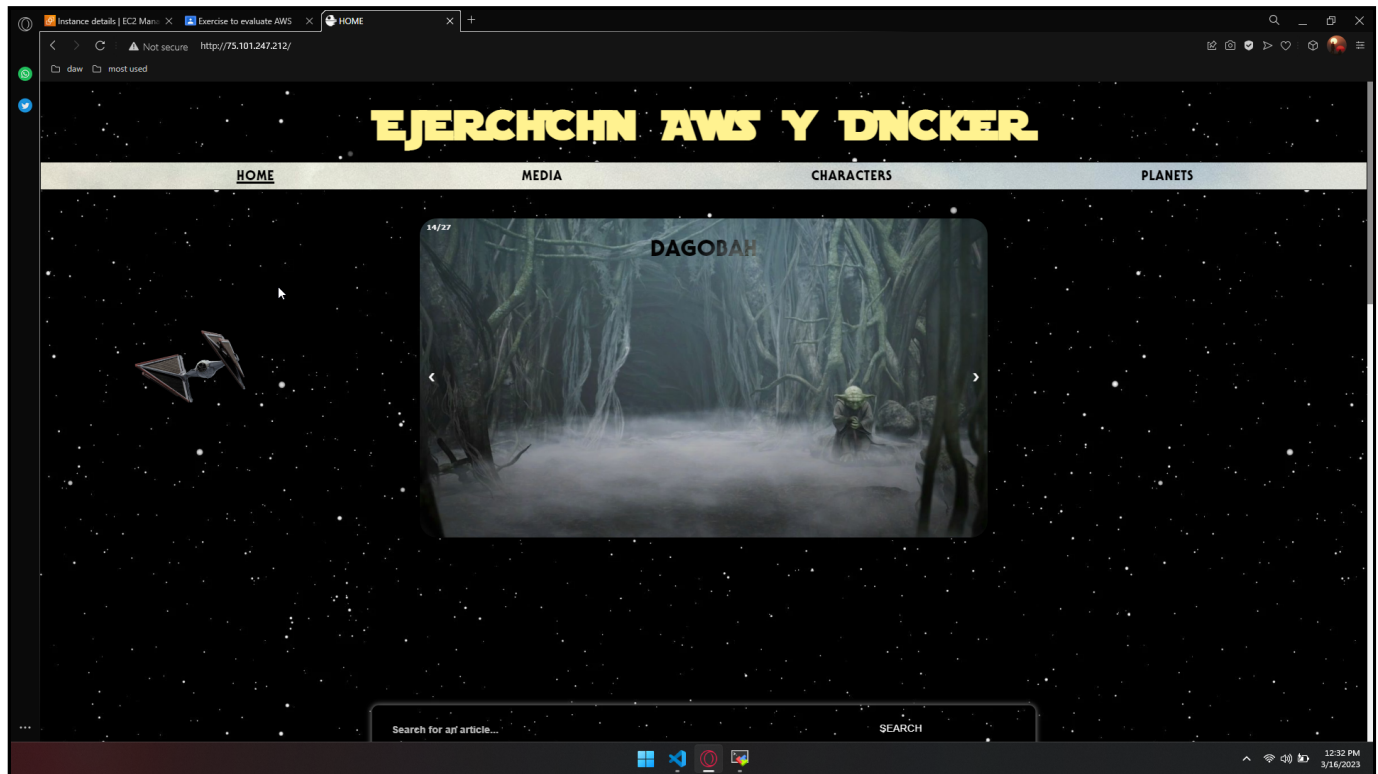
```
scp -r -i "/home/christianms13/vockey.pem" fin_1er_trimestre/index.html
ubuntu@75.101.247.212:/home/ubuntu/
```

Desde la instancia ejecutamos el comando "exit" para volver al usuario "ubuntu". Primero elimino el archivo "index.html" del directorio "miweb" y después muevo el recién modificado index a dicho directorio de nuevo.

```
sudo rm /home/christianmillan/miweb/index.html
```

```
sudo mv index.html /home/christianmillan/miweb/
```

Como se puede ver a continuación, los cambios se han realizado y actualizado con éxito:



## 2.c. Buckets en AWS S3:

Desde la consola del "UbuntuWebServer", lanza las instrucciones "aws s3" necesarias para crear un bucket S3 donde sincronizar tu workdir es este bucket llamado "< misiniciales >backup".

Si no estamos en el usuario "ubuntu", ejecutamos el comando "exit" para salir del usuario personal, ya que no tiene permisos suficientes.

Una vez estamos en el usuario "ubuntu", ejecutamos el siguiente comando para instalar AWS CLI en la instancia:

```
sudo apt install awscli -y
```

De nuevo, pulsamos ENTER (OK) cada vez que sea necesario sin tocar ningún ajuste.

Cuando se termine de instalar, debemos configurar las credenciales de AWS CLI.

En la consola de la instancia, entramos de nuevo en el usuario personal. Mi comando es el siguiente:

```
su christianmillan
```

Desde este usuario tenemos que ejecutar el siguiente comando para crear el directorio de configuración en nuestro workdir:

```
mkdir ~/.aws
```

Dentro de este nuevo directorio deberemos crear 2 archivos de configuración: uno para la región de nuestra cuenta y otro para las credenciales. Empezamos por el de la región.

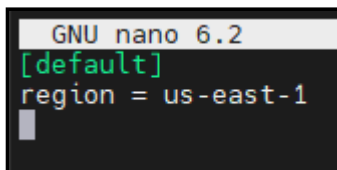
Ejecutamos el siguiente comando para crear el archivo:

```
nano ~/.aws/config
```

Se nos abrirá el editor para incluir contenido dentro del archivo. El contenido es el siguiente:

```
[default]
region = us-east-1
```

El contenido anterior se debe escribir tal cual aparece aquí dentro del archivo que hemos creado.



Para guardar pulsamos primero "CTRL+O", después ENTER y después "CTRL+X".

Ahora pasamos al siguiente archivo, con el siguiente comando:

```
nano ~/.aws/credentials
```

El contenido que debe tener este fichero aparece en la pestaña de "AWS Details" al entrar en los módulos del Learner Lab

03:48 Start Lab End Lab AWS Details Readme Reset

### Cloud Access

**AWS CLI:**  
Copy and paste the following into ~/.aws/credentials

```
[default]
aws_access_key_id=ASIA3YV55J2YCJWLBQUQ
aws_secret_access_key=7u9Qh6QlX0Dc2s6fa4j5Z+gvai06FJxFZcm3/t/V
aws_session_token=FwoGZXIvYXZdEGAADEw4IVwzFUxKPrG7CLCAf/PamX5IbUarAyE7xJ9rCfNKVWPJ3SjybuMLBwKVIn06NFDU48MBLi2agZRMhEMyzOTxh9YuovtQtOXk/k7gngI3gVjo7bnZ6k5/haq/zKBBc3wf0NXYZ8bINHMxiPvFO/JNDAOFVsL7HmnnGVmCO1xZSqUPGD2ZF1VgkGK94pYxLhx17TbPYUWtgI6AOjD7/OUqo7GH5mHZpwuDL5CvEXbxbBdzyxy0tPW7wcs7m9QM1IE0009BB0V/W/u10/iLsLKQ/XzKAGMi3P3hn8SzZkLRuD6WpdgNA5r8UrkUn/maKPj+SVNwHDexQa+2HvS3IGuaJiF8w=
```

### Cloud Labs

Remaining session time: 03:47:58(228 minutes)  
Session started at: 2023-03-16T07:47:09-0700  
Session to end at: 2023-03-16T11:47:09-0700

Accumulated lab time: 2 days 03:06:00 (3066 minutes)

No running instance

SSH key [Show](#) [Download PEM](#) [Download PPK](#)

AWS SSO [Download URL](#)

AWSAccountId	808925875888
Region	us-east-1

Contenido del fichero en mi caso:

```
GNU nano 6.2
[default]
aws_access_key_id=ASIA3YV55J2YCJWLBQUQ
aws_secret_access_key=7u9Qh6QlX0Dc2s6fa4j5Z+gvai06FJxFZcm3/t/V
aws_session_token=FwoGZXIvYXZdEGAADEw4IVwzFUxKPrG7CLCAf/PamX5IbUarAyE7xJ9rCfNKVWP
```

Ahora el comando "aws configure" utilizará estos valores predeterminados para configurar las credenciales con el AWS CLI.

```
christianmillan@ip-10-0-1-10:~$ aws configure
AWS Access Key ID [*****BQUQ]:
AWS Secret Access Key [*****/t/V]:
Default region name [us-east-1]:
Default output format [None]:
```

\*No se introduce nada en ninguno de los campos que se piden, ya que se establecen de forma predeterminada.

Ahora, con el siguiente comando creamos el nuevo bucket. Debido a que el nombre que me corresponde siguiendo las indicaciones del enunciado de este ejercicio ya está cogido, he tenido que cambiar el nombre, pero siguiendo utilizando mis iniciales:

```
aws s3 mb s3://ejfinalcms
```

```
christianmillan@ip-10-0-1-10:~$ aws s3 mb s3://ejfinalcms  
make_bucket: ejfinalcms
```

Y, por último, con el siguiente comando sincronizamos el workdir de nuestro usuario personal de la instancia de AWS con el bucket recién creado:

```
aws s3 sync . s3://ejfinalcms
```

El punto "." hace referencia al directorio actual en el que estamos, por lo que tenemos que asegurarnos que estamos en el directorio home de nuestro usuario personal.