

Instalación de un servidor Ubuntu en AWS


{ Christian Millán Soria }



- Curso: 2023-2024
- Clase: 2º DAW Tarde
- Profesor: David Hormigo Ramírez
- Módulo: Despliegue de Aplicaciones Web
- Fecha: 19/09/2023

<u>1. Creación de una instancia</u>	3
1.1. Inicio de sesión en AWS	3
1.2. Console Home	5
1.3. EC2	6
<u>2. Configuración de la instancia</u>	6
2.1. Nombre y etiquetas	6
2.2. Imágenes de aplicaciones y sistemas operativos (AMI)	7
2.3. Tipo de instancia	8
2.4. Par de claves	8
2.5. Configuraciones de red	9
2.6. Almacenamiento	10
<u>3. Conexión mediante SSH</u>	11
3.1. Obtención de las credenciales	12
3.2. PuTTY	13
3.3. MobaXterm	14
<u>4. Software actualizado</u>	16
<u>5. Extensión AWS</u>	17
5.1. Añadir credenciales	18
5.2. Demostración con bucket	20
<u>6. AWS CLI</u>	21
6.1. Configuración de AWS CLI	22
6.2. Infraestructura EC2 en formato JSON	24

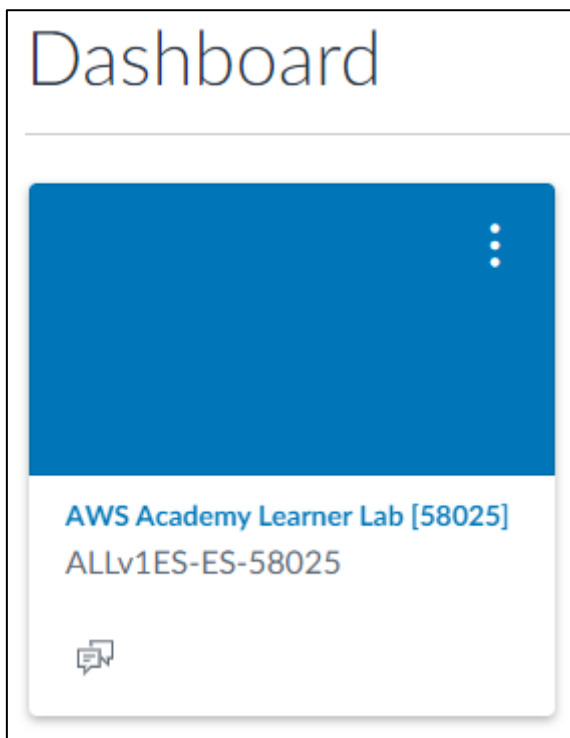
1. Creación de una instancia

Debido a la naturaleza de esta práctica, la creación de la instancia se va a explicar de forma completa: desde como acceder a este proceso en **AWS** , hasta cómo realizar una conexión a la misma mediante “SSH”.

1.1. Inicio de sesión en AWS

El primer paso en esta guía es acceder a **AWS (Amazon Web Services)**. En mi caso, he accedido con mis credenciales proporcionadas por el centro educativo para poder optar al crédito de Amazon y poder trabajar con el laboratorio de AWS.

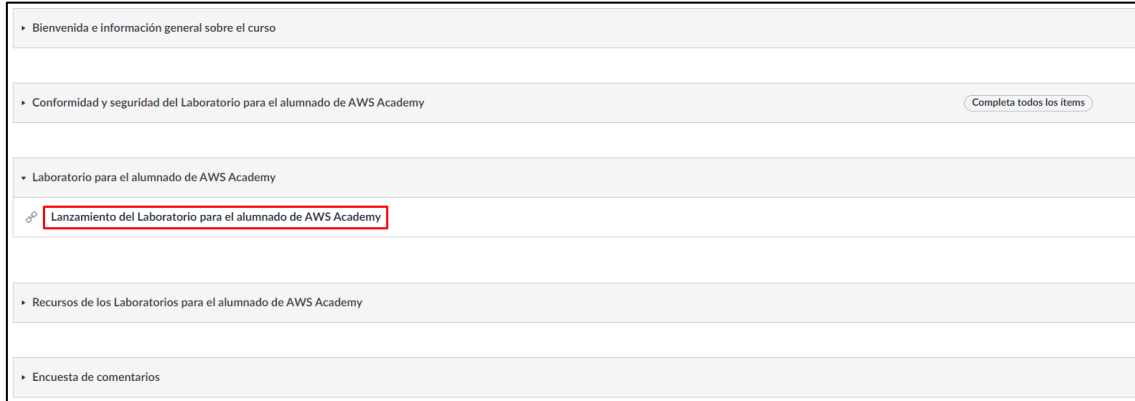
Una vez iniciada la sesión con la cuenta de Amazon, nos dirigimos al “**Panel de Control**” del usuario. En esta zona nos encontramos con un módulo accesible. Hacemos clic en él.



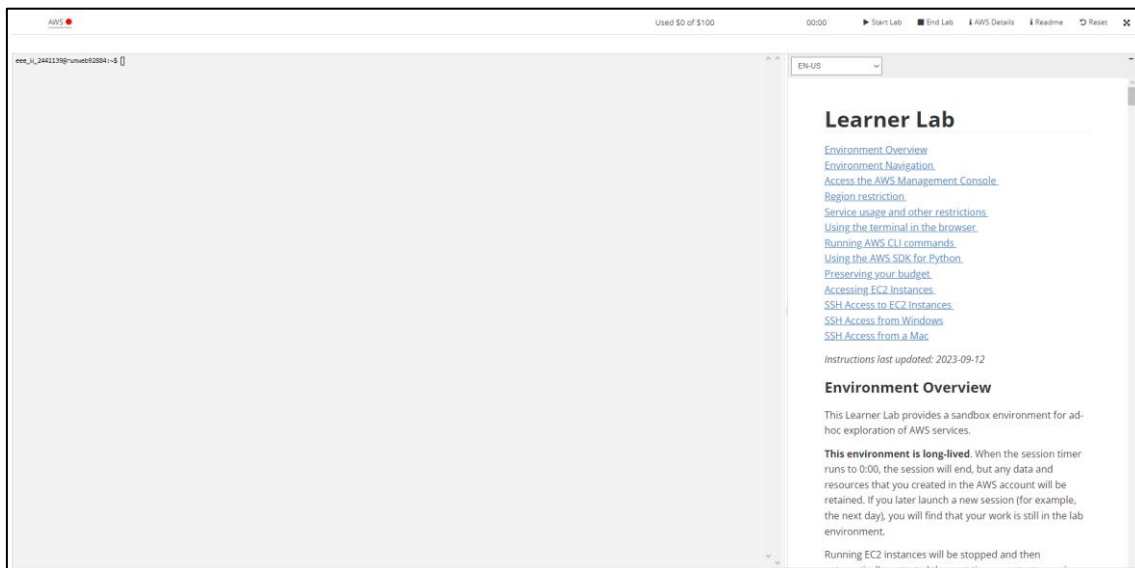
Dentro aparecen tres pestañas: “Página de inicio”, “Contenidos” y “Foros de discusión”. Nos dirigimos a la pestaña de “**Contenidos**”.

Instalación de un servidor Ubuntu en AWS

A su vez, esta pestaña se divide en un total de cinco submenús. Nos centraremos únicamente en el tercero, dedicado al propio laboratorio donde realizaremos gran parte de la práctica.

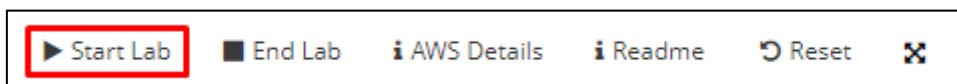


Accedemos al elemento dentro de este submenú. Se nos presenta una vista similar a la siguiente (si no igual):



Podemos ver un terminal de comandos proporcionado por Amazon, una serie de botones con distintas funcionalidades y un panel lateral que muestra el contenido destinado a algunos de esos botones.

Mas adelante veremos para qué funcionan algunos de ellos, pero por ahora nos conformaremos con hacer clic en el botón **“Start Lab”**.



1.2. Console Home

En la parte superior izquierda, antes de hacer clic en el botón “Start Lab”, aparecía lo siguiente:



Este icono en rojo indica que el laboratorio de AWS está apagado, por lo que no consume crédito del saldo proporcionado a nuestra cuenta.

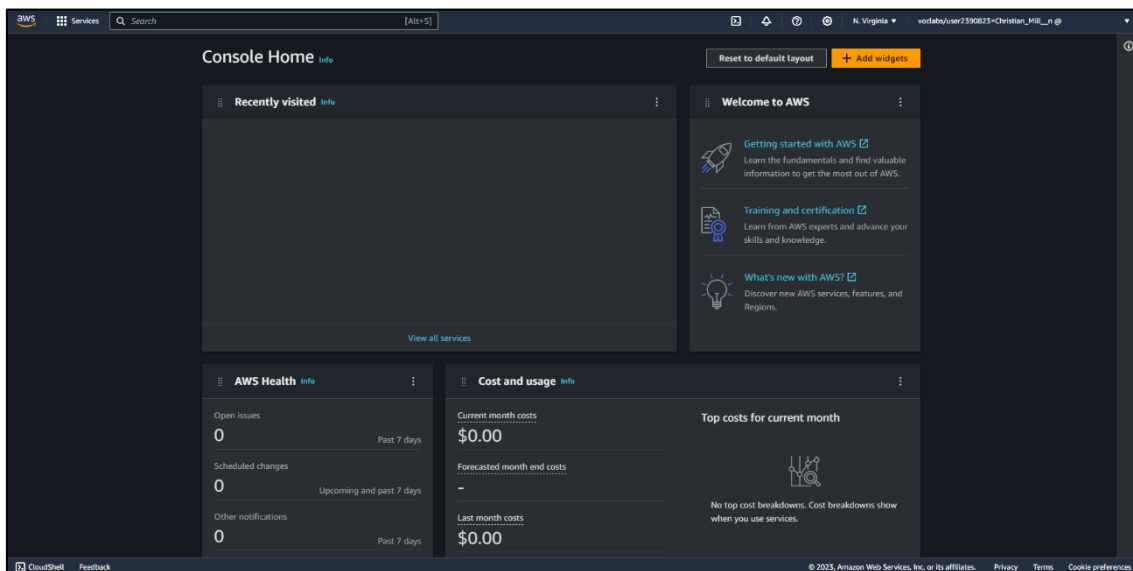
Una vez hemos pulsado el botón, el estado del icono ha cambiado a color amarillo. Esto indica que el laboratorio está en proceso de encenderse. El primer encendido del laboratorio es más lento que los encendidos futuros, debido a que se configura todo por primera vez.



Cuando el laboratorio se ha encendido por completo, el icono cambia a color verde.





Para acceder al laboratorio una vez encendido, clicamos en el icono en verde. Esto abre una nueva pestaña, con un panel de control general llamado “Console Home”.



1.3. EC2

En la esquina superior izquierda de esta página, contamos con una barra de búsqueda.

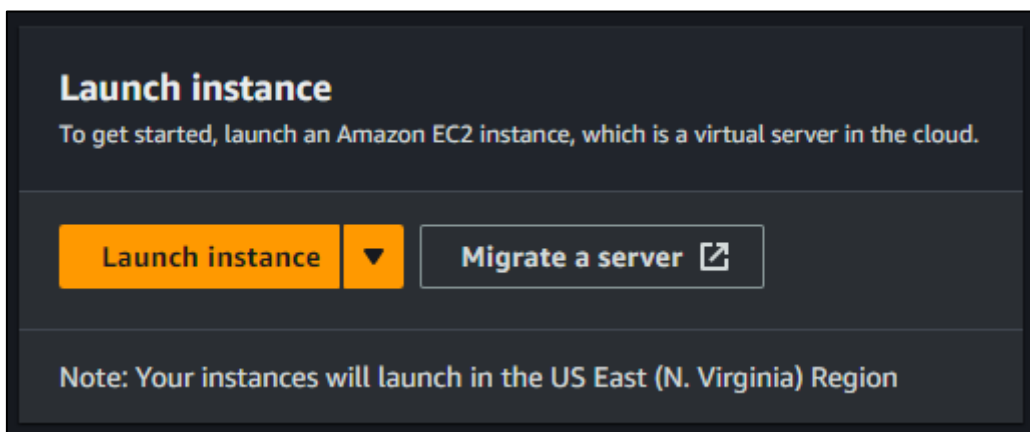


AWS se basa en el ofrecimiento de los llamados “**Servicios**”. Algunos ejemplos de servicios son “**S3**” (**Simple Storage Service**) , para crear objetos de almacenamiento en la nube; o “**EC2**” (**Elastic Compute Cloud**), para la creación de las llamadas “**instancias**”  (servidores en la nube).

Para esta práctica necesitamos utilizar el servicio de “EC2”.

Realizamos la consulta en la barra de búsqueda y entramos al panel de control de “EC2”.

Entre todas las opciones y recuadros que aparecen disponibles, seleccionamos la opción “**Lanzar instancia**” (opción de color naranja).



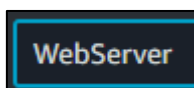
2. Configuración de la instancia

Se nos abre un menú de configuración básica y avanzada para la nueva instancia que vamos a crear.


Vamos a ir por partes.

2.1. Nombre y etiquetas

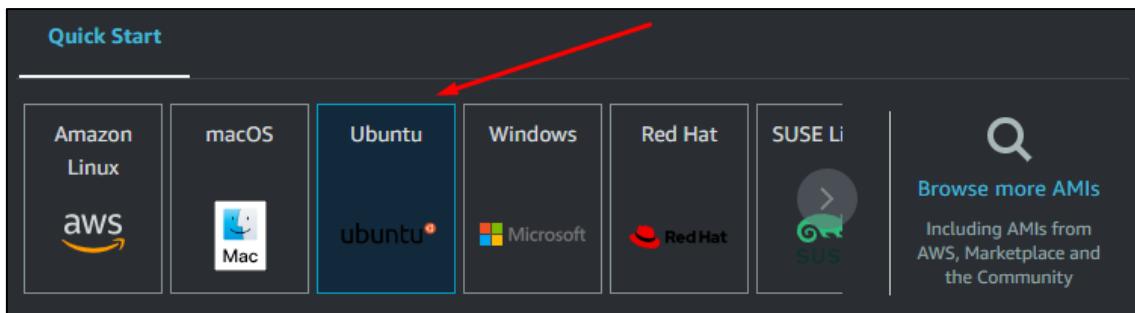
Lo primero es darle un nombre a nuestra nueva instancia. En la documentación de la práctica se especifica que el nombre debe ser “**WebServer**”.



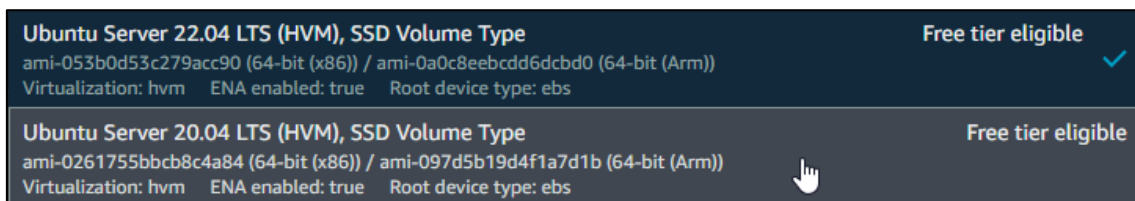
2.2. Imágenes de aplicaciones y sistemas operativos (AMI)

El término “**AMI**” (**A**mazon **M**achine **I**mage)  hace referencia a la plantilla/sistema operativo que AWS instalará en nuestra instancia para que el servidor pueda funcionar.

Para esta práctica instalaremos un servidor Ubuntu, por lo que tenemos que seleccionar el sistema operativo “**Ubuntu**” en el slider de imágenes.

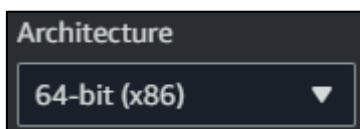


En el desplegable justo debajo de lo anterior, aparece seleccionada de forma predeterminada la versión 22.04 LTS de Ubuntu. Abrimos dicho desplegable y optamos por la versión **20.04 LTS** para nuestro servidor.




La elección de una versión anterior se justifica debido a que las versiones más recientes pueden tener una mayor probabilidad de presentar errores o problemas de rendimiento inesperados, mientras que las versiones más antiguas suelen estar más pulidas gracias a su mayor tiempo de desarrollo.

Para finalizar, encontraremos un menú desplegable que nos permite seleccionar la arquitectura de nuestro servidor, y en este caso, es recomendable dejarlo en la opción predeterminada: **64 bits (x86)**. Optamos por la arquitectura de 64 bits (x86) en lugar de ARM debido a su mayor compatibilidad con una amplia gama de aplicaciones y servicios, lo que facilita la administración y el despliegue en entornos variados.

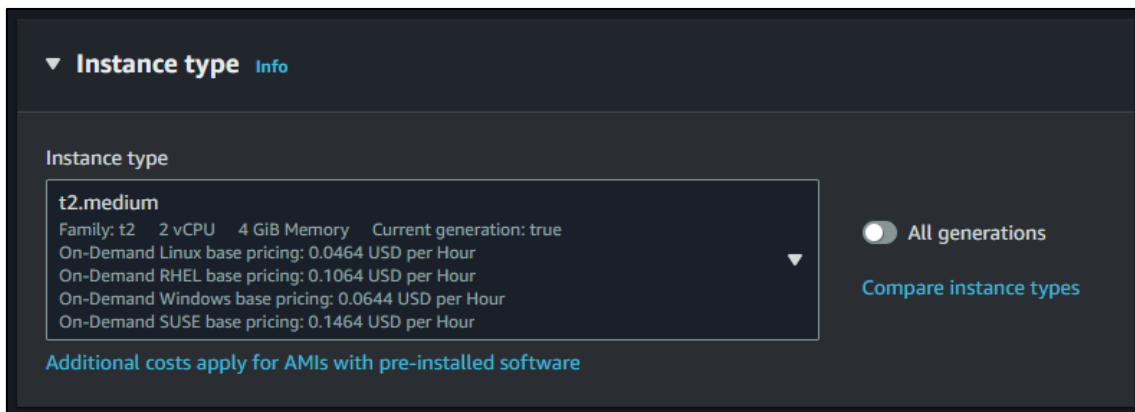


2.3. Tipo de instancia

El siguiente paso es establecer los recursos de los que dispondrá el servidor una vez esté en funcionamiento.

Gracias a que contamos con un plan proporcionado por el centro, tenemos disponibles varios tipos de instancias para nuestro uso. En el caso de esta práctica nos conformaremos con el tipo “**t2.medium**” , que cuenta con 2 vCPU y 4 GiB de memoria, suficientes para el trabajo que realizaremos con este servidor.


Abrimos el desplegable y seleccionamos la opción deseada: “t2.medium”.



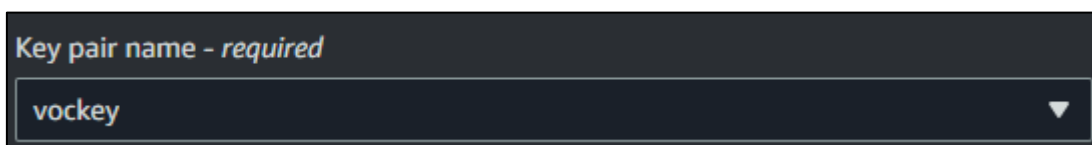
2.4. Par de claves

El inicio de sesión es una parte crucial para nuestra instancia. Necesitamos de un método seguro para poder acceder al control del servidor, de manera directa y sin peligro de que un usuario externo pueda alterar el funcionamiento del mismo sin permiso.

Aquí es donde entra en juego el par de claves de AWS.

Conocido como “**vockey**” , es un fichero único para cada usuario de AWS que permite realizar la conexión a las instancias y demás servicios sin miedo a que nadie más pueda acceder a dicha clave, siempre que no se comparta de forma indebida, claro.

En este caso el proceso es fácil: abrimos el desplegable que aparece en el apartado para el par de claves y seleccionamos la opción “vockey”.

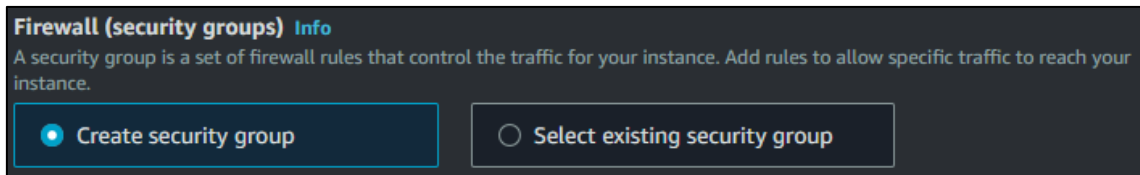


Más adelante en la guía se explicará cómo descargar el fichero necesario para poder realizar las conexiones con “vockey”.

2.5. Configuraciones de red


Este es un apartado delicado del proceso de configuración, ya que cualquier cambio indebido puede resultar en un mal desempeño de las conexiones y desarrollo del servidor.

Lo primero: dejamos la opción marcada de forma predeterminada **“Crear grupo de seguridad”**.




El grupo de seguridad es el que se encargará de establecer las reglas de entrada y salida para las conexiones de la instancia.

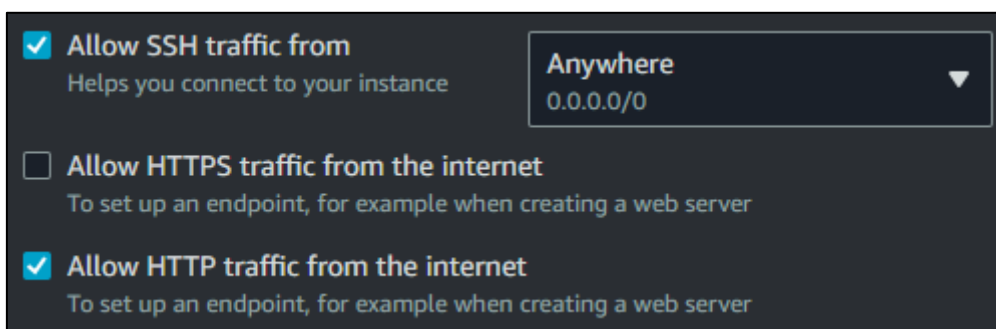
Por último en este apartado, pero no menos importante, viene la configuración del tráfico de conexiones según el protocolo. Nosotros nos centraremos en dos de ellos:

- **“SSH” (Secure Shell)**  : es el protocolo utilizado para acceder de forma segura a un servidor. Proporciona un canal cifrado para la comunicación entre el cliente y el servidor.

Marcaremos la casilla para este protocolo, dejando el recuadro de la derecha en su forma predeterminada (**“Cualquier lugar 0.0.0.0/0”**). El recuadro indica que las conexiones mediante SSH hacia nuestro servidor se podrán realizar desde cualquier parte del mundo y **desde cualquier dirección IP**. Esto es una configuración a tener en cuenta para un servidor con un fin más profesional, pero por ahora nos funcionará sin problemas para esta práctica.

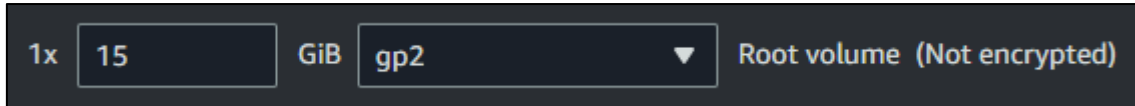
- **“HTTP” (Hypertext Transfer Protocol)**  : “HTTP” es el protocolo utilizado para servir contenido web y permite que los navegadores web se conecten al servidor para acceder a sitios web.


Marcamos también la casilla de este protocolo (no confundirlo con el protocolo “HTTPS”, el cual ofrece una seguridad adicional con respecto al protocolo “HTTP”).



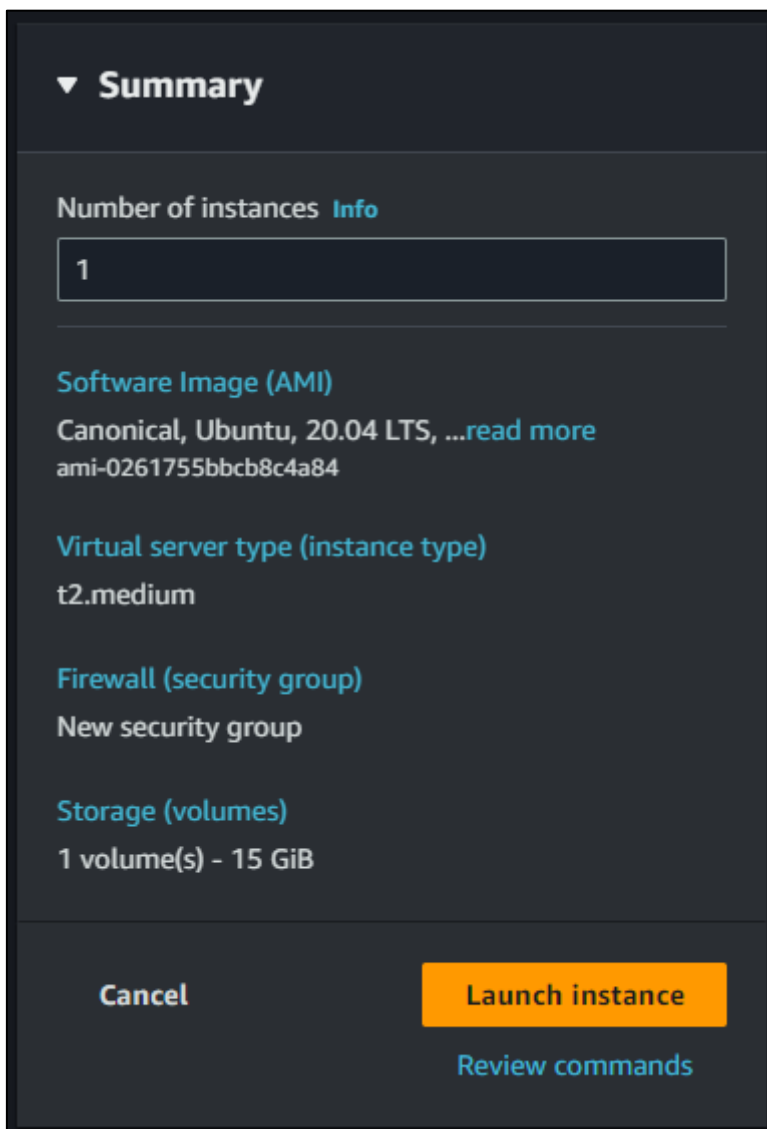
2.6. Almacenamiento

Llegando al final del proceso de configuración de la nueva instancia, debemos establecer el almacenamiento con el que contará nuestro servidor.

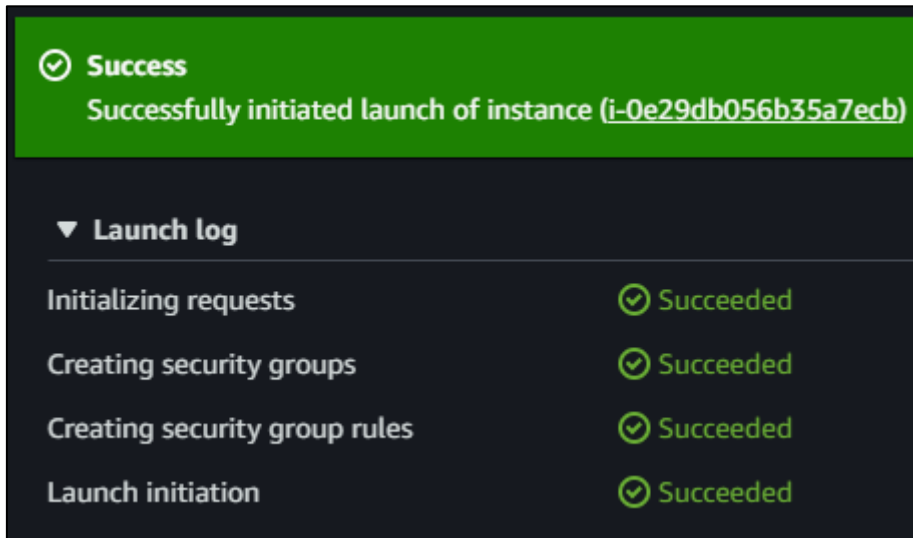
A screenshot of the AWS console showing the storage configuration for a new instance. It features a dark-themed interface with a row of configuration options: a multiplier '1x', a text input field containing '15', the unit 'GiB', a dropdown menu showing 'gp2' with a downward arrow, and the text 'Root volume (Not encrypted)'.

Aumentamos el tamaño del almacenamiento a **15 GiB** y dejamos el tipo en “**gp2**”  debido a su equilibrio ideal entre rendimiento y costo, asegurando así un funcionamiento eficiente y flexible para nuestro servidor en AWS.

Una vez hecho todo esto, clicamos en el botón “**Lanzar instancia**”, que aparece a la derecha de la columna de configuración, en naranja.

A screenshot of the AWS console 'Summary' section for launching an instance. The section is titled 'Summary' with a downward arrow. It contains several configuration details: 'Number of instances' set to '1' with an 'Info' link; 'Software Image (AMI)' as 'Canonical, Ubuntu, 20.04 LTS, ...read more' with the ID 'ami-0261755bbcb8c4a84'; 'Virtual server type (instance type)' as 't2.medium'; 'Firewall (security group)' as 'New security group'; and 'Storage (volumes)' as '1 volume(s) - 15 GiB'. At the bottom, there are three buttons: 'Cancel', 'Launch instance' (highlighted in orange), and 'Review commands'.

Se muestra un mensaje que confirma la finalización del proceso con éxito.



Si volvemos al apartado de “**Instancias**” podemos ver que la lista de instancias de nuestra cuenta se ha actualizado con esta.



Name ▾	Instance ID	Instance state ▾
WebServer	i-0e29db056b35a7ecb	Running


Podemos ver el estado de la instancia (hay que tenerlo en cuenta, ya que una instancia en estado “**En ejecución**” consume crédito de nuestra cuenta de AWS).

3. Conexión mediante SSH

Vamos a proceder a conectarnos al servidor mediante el protocolo “SSH”, mencionado anteriormente.

Para realizar esta conexión podemos optar por una de las siguientes opciones:

- **Terminal del Sistema Operativo:** Se utiliza el terminal nativo del propio sistema operativo en el que se va a realizar la conexión. Con el comando “**ssh**”  y estableciendo una **dirección IP pública** y un **usuario** para acceder al servidor por línea de comando.
- **Entorno gráfico:** Es una alternativa más “amigable” para los usuarios menos experimentados. Requiere de la utilización de un programa de terceros. La opción habitual y más común es “**PuTTY**” , un programa que proporciona un entorno gráfico para el proceso de conexión al servidor, aunque después pasa a ser un terminal de comandos como en la primera opción.

En mi caso, utilizaré el programa “**MobaXterm**”  que, aunque similar a “PuTTY”, ofrece más opciones de personalización y de conexión.

Debido a que el entorno que AWS recomienda es “PuTTY”, se añadirá una guía de conexión en los dos programas.

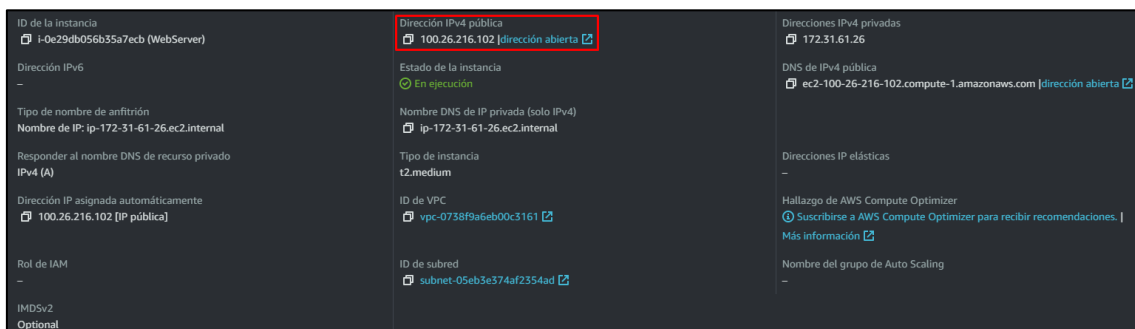
3.1. Obtención de las credenciales

Lo primero, antes de empezar la conexión, es obtener las credenciales necesarias para realizar la misma.

En el menú de “Instancias”, y con la instancia recién creada a la vista, hacemos clic en el elemento “**ID de la instancia**” (de color azul).

Esto abre una página resumen de todo lo que hemos configurado en dicha instancia.

Lo que necesitamos es obtener la dirección IP pública de nuestro servidor, la cual se encuentra en la columna central del resumen, el primer elemento. Hacemos clic en el botón situado a la izquierda de la IP.




Esto ha copiado la dirección al portapapeles de nuestro equipo.

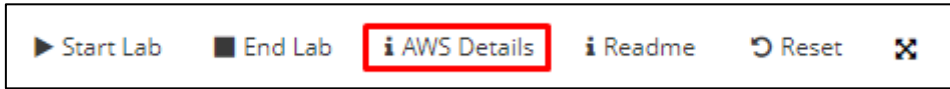
El otro elemento necesario para el correcto desempeño de la conexión por SSH es el usuario con el que se realizará dicha conexión, el cual es “**ubuntu**”, creado de forma automática por AWS al crear la instancia.

Su contraseña es “**ubuntu**”, pero en AWS podemos utilizar el **par de claves**, como ya vimos anteriormente.

El par de claves es un archivo cifrado que AWS ofrece para crear una conexión segura hacia nuestras instancias y demás servicios.

Para descargar dicho archivo, primero debemos dirigirnos al workbench de AWS  donde podíamos encender el laboratorio o apagarlo, entre otras opciones.

Una vez aquí, hacemos clic en el botón “AWS Details”.



Esto abre una sección de información sobre nuestro laboratorio.

En la parte inferior de esta sección podemos ver dos botones: “Download PEM” y “Download PPK”.



Para la realización de nuestra práctica vamos a necesitar hacer clic en el segundo para descargar el archivo con extensión “.ppk”. Esto es porque AWS proporciona este archivo en concreto para las conexiones realizadas con un programa de entorno gráfico, como lo son “PuTTY” o “MobaXterm”.

En cambio, el archivo con extensión “.pem” se utiliza para el otro método mencionado antes, en el cual se utiliza el terminal de comandos para conectarse al servidor por SSH.

Guardamos el archivo “.ppk” en un lugar seguro y localizado en nuestro equipo.

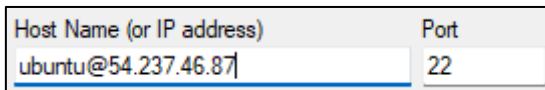
3.2. PuTTY

Se presupone que “PuTTY” ya está instalado.

Al abrir “PuTTY”, nos centramos en el cuadro de texto debajo del título “Host Name (or IP address)”. En este cuadro debemos ingresar primero el usuario con el que se realizará la conexión, que en nuestro caso es el usuario “ubuntu”, seguido de un “@”.

Seguidamente y sin espacios, ingresamos la dirección IP pública que copiamos anteriormente desde el resumen de nuestra instancia.

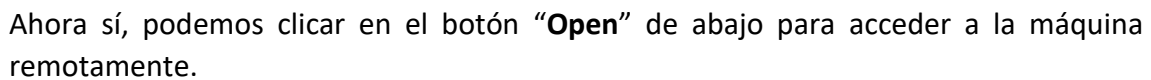
El puerto, que aparece a la derecha, lo dejamos de forma predeterminada con el 22, ya que es el por defecto para el protocolo “SSH”.



Lo usual ahora sería clicar en el botón “Open” de abajo para entrar al terminal de comandos y poder introducir la contraseña del usuario con el que hemos decidido entrar a la máquina remotamente, pero como ya hemos visto, vamos a utilizar el par de claves de AWS.

Para esto, nos dirigimos al menú “Connection”, situado a la izquierda de nuestra interfaz gráfica. Desplegamos este menú si no lo está ya. Dentro vemos un submenú llamado

Aquí vemos varias opciones, pero la que a nosotros nos interesa es la de debajo del título **“Private key file for authentication”**. Clicamos en el botón **“Browse”** y buscamos el archivo con extensión **“.ppk”** que hemos descargado con anterioridad.



```
ubuntu@172-31-61-26:~$
# Using username "ubuntu".
# Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.15.0-104-aws x86_64)

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

System information as of Fri Sep 22 17:37:17 UTC 2023

System load: 0.0          Processes:    109
Usage of /: 12.3% of 14.5GB   Memory: 100%
Memory usage: 6%            IP address for eth0: 172.31.61.26
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 apt update

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Sep 22 17:36:00 2023 from 46.27.76.3
To run a command as administrator (user "root"), use "sudo ".
See "man sudo_root" for details.

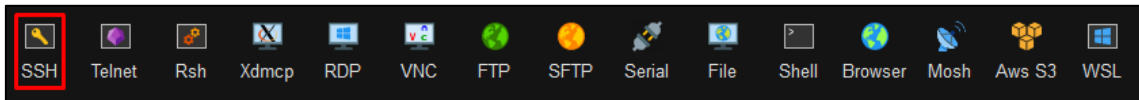
ubuntu@172-31-61-26:~$
```

Se presupone que “MobaXterm” ya está instalado.



Session

14



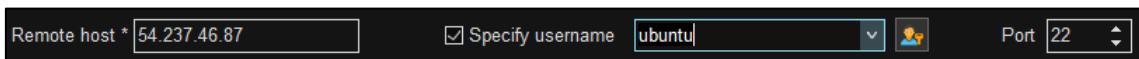
La interfaz se divide en dos zonas clave para nuestra práctica: la primera enfocada en la configuración básica de la conexión a realizar y la segunda donde especificaremos que esta se va a autenticar mediante un par de claves.

En la primera nos aparecen dos recuadros a rellenar.

En el primero ingresamos la dirección IP pública que copiamos anteriormente desde el resumen de nuestra instancia.

En el segundo recuadro marcamos la opción “**Specify username**” y rellenamos el cuadro con el texto “**ubuntu**”, ya que será el usuario que utilizaremos para conectarnos al servidor.

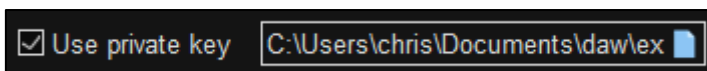
El puerto lo dejamos de forma predeterminada en el 22, ya que es el por defecto para el protocolo “SSH”.



Pasamos a la segunda zona de la división, donde tenemos que entrar en la pestaña “**Advanced SSH settings**”.

Se nos muestran una serie de opciones disponibles para personalizar la conexión, pero nos centraremos en la opción “**Use private key**”, en la esquina inferior izquierda.

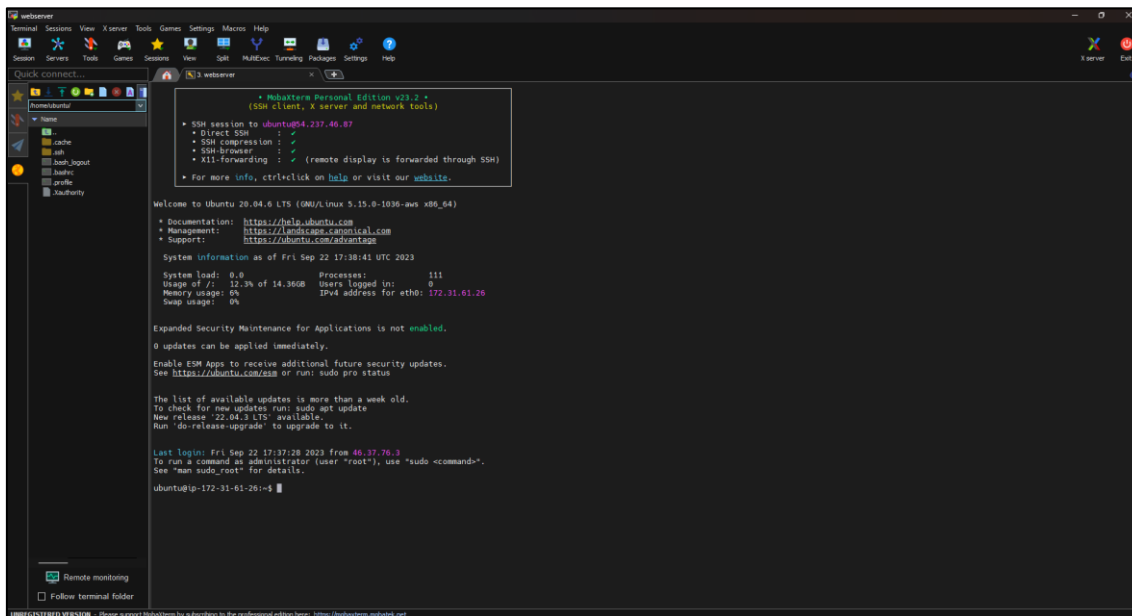
Marcamos esta opción y en el recuadro que aparece a su derecha hacemos clic para poder seleccionar el archivo con extensión “.ppk” que nos descargamos con anterioridad en nuestro equipo.



Una vez hecho todo esto, clicamos en el botón “**OK**” de abajo para realizar la conexión con el servidor.

Instalación de un servidor Ubuntu en AWS

Se muestra un terminal de comandos similar al siguiente:



4. Software actualizado

Lo siguiente es comprobar que el software instalado en la nueva instancia cuenta con la última versión disponible para procurar encontrarnos con el menor número de bugs posible y contar con una mayor estabilidad.


Utilizaremos comandos de Linux, ya que el servidor que hemos creado cuenta con “Ubuntu Server 20.04 LTS”.

Ejecutando el comando “**sudo apt update -y && sudo apt upgrade -y**” actualizamos el sistema a su última versión.


En realidad, esto son dos comandos concatenados. Gracias a los dos “&&” se pueden ejecutar varios comandos de una vez sin tener que esperar a que se ejecute uno para utilizar el siguiente.

El comando “**sudo**” es uno que se utiliza para ejecutar otros comandos con privilegios de administrador o “**superusuario**” en sistemas Linux. El superusuario (también conocido como “**root**”) tiene control total sobre el sistema y puede realizar cambios críticos.


El comando “**apt update**” en Linux actualiza la lista de repositorios de software disponibles en el sistema, permitiendo que este conozca las últimas versiones de software y su ubicación en línea.

Por otro lado, "**apt upgrade**"  se utiliza para actualizar el software instalado a las versiones más recientes disponibles en los repositorios, asegurando que todas las aplicaciones en el sistema estén actualizadas sin instalar nuevo software.

```
ubuntu@ip-172-31-61-26:~$ sudo apt update -y && sudo apt upgrade -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-61-26:~$
```

Los operadores "**-y**"  se utilizan para aceptar la ejecución de estos comandos sin tener que pararnos a aceptar todo.

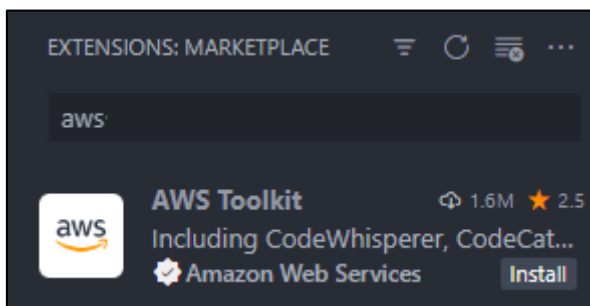
5. Extensión AWS

Esta parte de la práctica se basa en mostrar cómo se ha descargado la extensión de AWS en Visual Studio Code  y comprobar su funcionamiento.

Se presupone que Visual Studio Code ya está instalado.

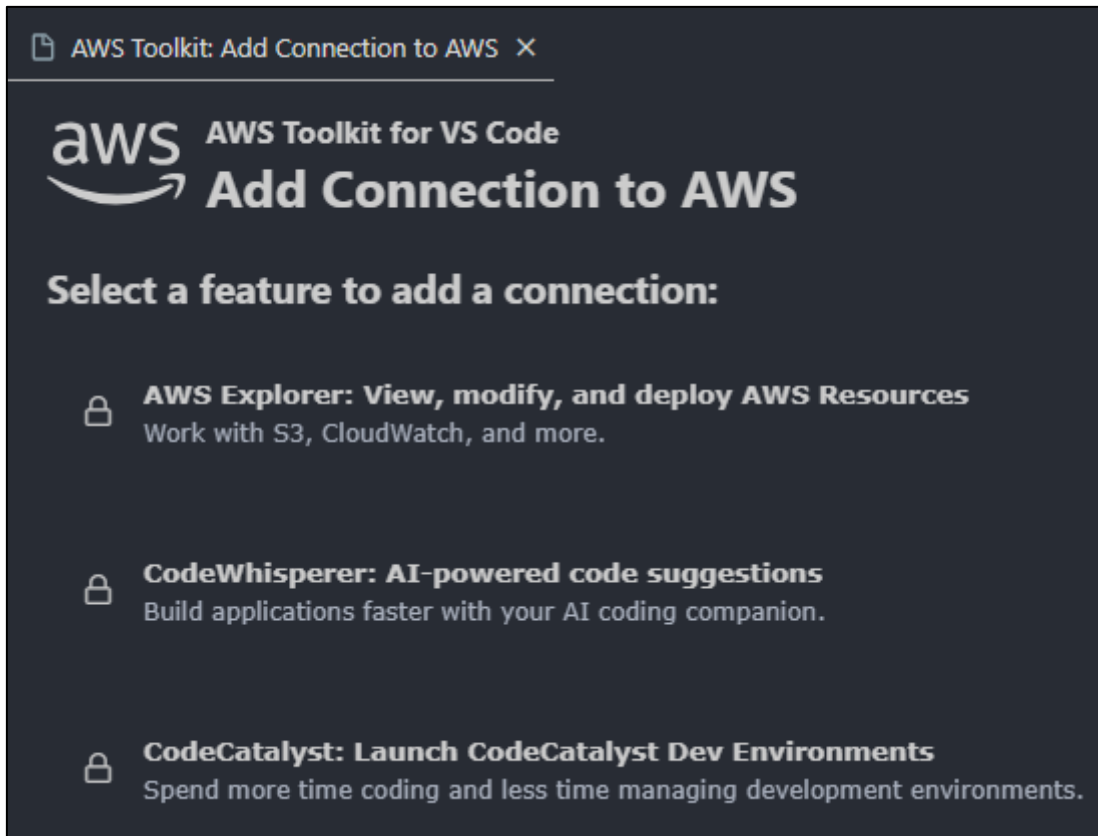
El primer paso es, obviamente, abrir el programa. Una vez dentro, utilizamos el shortcut de teclado "**Ctrl + Shift + X**" para acceder al menú de extensiones del mismo (si no se ha cambiado el shortcut manualmente).

Ingresamos la palabra "**aws**" en el buscador. Clicamos en "**Install**" en la extensión llamada "**AWS Toolkit**".



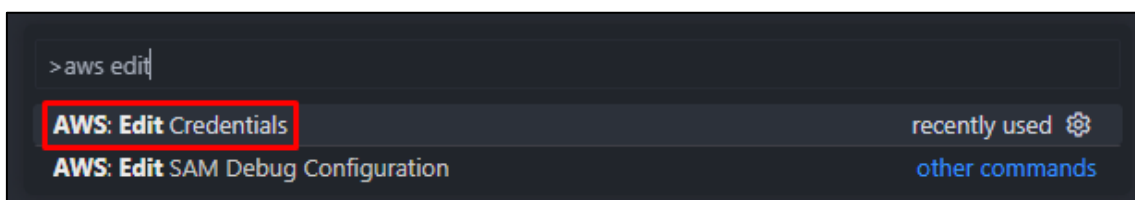
5.1. Añadir credenciales

Una vez instalada, se mostrará un contenido como el siguiente:



Lo primero que debemos hacer es iniciar sesión con nuestra cuenta de AWS. Para ello, utilizamos el atajo de teclado **“Ctrl + Shift + P”**, lo que abre la paleta de comandos de VS Code.


Aquí seleccionamos la opción **“AWS: Edit Credentials”**.

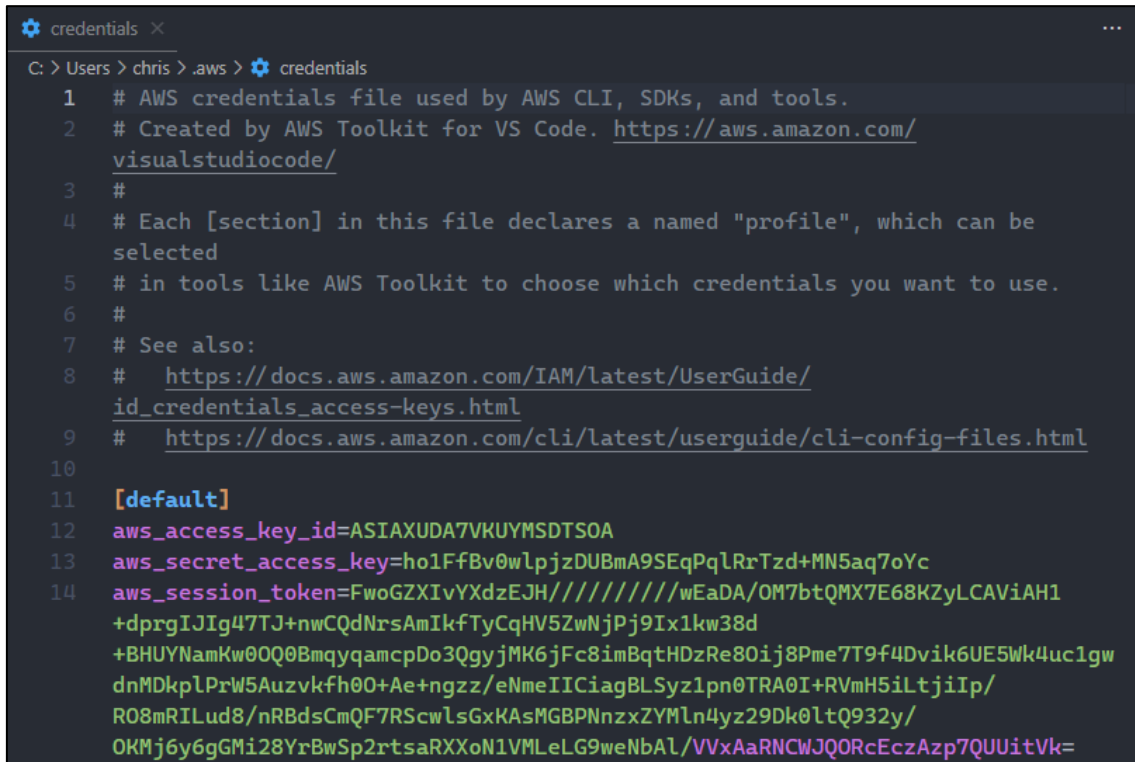


Esto abre el archivo **“credentials”**, donde se configuran las credenciales para las conexiones de **“AWS Toolkit”** con cuentas de AWS.

Nos dirigimos al final del archivo e introducimos las credenciales de nuestro usuario de AWS.

Instalación de un servidor Ubuntu en AWS

Las credenciales de nuestro usuario para “AWS CLI” las podemos sacar del workbench de AWS . En el botón “**AWS Details**”, podemos ver un botón “Show”, donde encontraremos todas nuestras credenciales. Las copiamos y las pegamos en el archivo “credentials”.



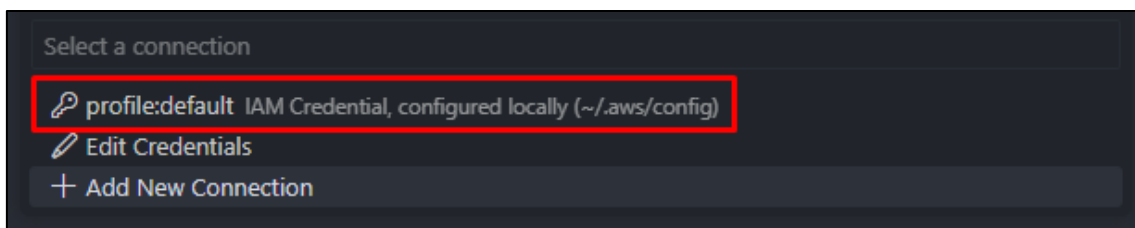
```

1 # AWS credentials file used by AWS CLI, SDKs, and tools.
2 # Created by AWS Toolkit for VS Code. https://aws.amazon.com/visualstudiocode/
3 #
4 # Each [section] in this file declares a named "profile", which can be
5 # in tools like AWS Toolkit to choose which credentials you want to use.
6 #
7 # See also:
8 # https://docs.aws.amazon.com/IAM/latest/UserGuide/id\_credentials\_access-keys.html
9 # https://docs.aws.amazon.com/cli/latest/userguide/cli-config-files.html
10
11 [default]
12 aws_access_key_id=ASIAxUDA7VKUyMSDTSOA
13 aws_secret_access_key=ho1FfBv0wlpjzDUBmA9SEqPqLRrTzd+MN5aq7oYc
14 aws_session_token=FwoGZXIvYXdzEjH////////wEaDA/OM7btQMX7E68KZyLCAViAH1
+dprgIJIg47TJ+nwCQdNrsAmIkfTyCqHV5ZwNjPj9Ix1kw38d
+BUHYNamKw00Q0BmqyqamcpDo3QgyjMK6jFc8imBqtHDzRe80ij8Pme7T9f4Dvik6UE5Wk4uc1gw
dnMDkplPrW5Auzvkfh00+Ae+ngzz/eNmeIICiagBLSyz1pn0TRA0I+RVmH5iLtjiIp/
R08mRILud8/nRBdsCmQF7RScwlsGxKAsMGBPNnzxZYMLn4yz29Dk0ltQ932y/
OKMj6y6gGMi28YrBwSp2rtsaRXXoN1VMLeLG9weNbAl/VVxAaRNCWJQORcEczAzp7QUUitVk=

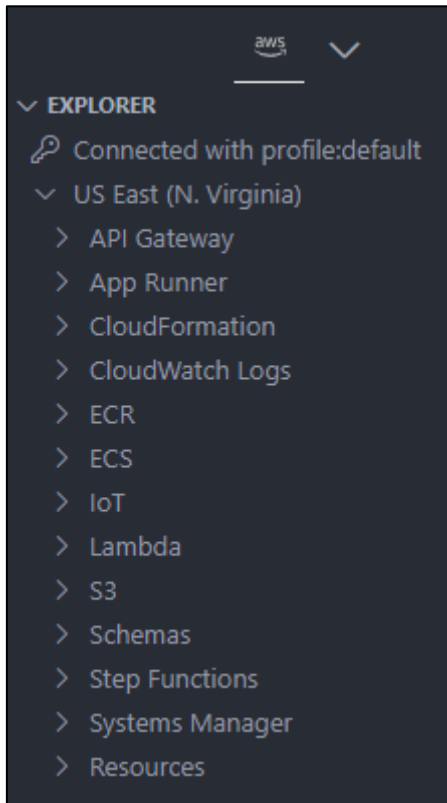
```

Ahora volvemos a utilizar el comando “Ctrl + Shift + P” y esta vez buscamos “**aws connect**”. Seleccionamos la opción “**AWS: Connect to AWS**”.

Aquí seleccionamos el perfil “**default**”, que acabamos de crear al ingresar nuestras credenciales.



Ahora nos aparecen todas las opciones de AWS en la barra izquierda de nuestro IDE.

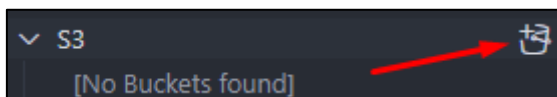


5.2. Demostración con bucket

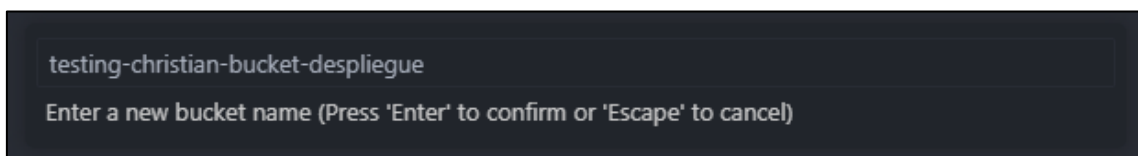
Vamos a probar que la conexión funciona y se puede interactuar con nuestra infraestructura de AWS desde el IDE.

Para ello, seleccionamos el desplegable “S3” de la barra lateral izquierda de la extensión “AWS Toolkit”.

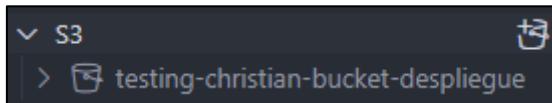
Hacemos clic en el botón “Create Bucket” que aparece a su derecha.



Se nos abre un cuadro de texto para ingresar el nombre para el nuevo bucket.

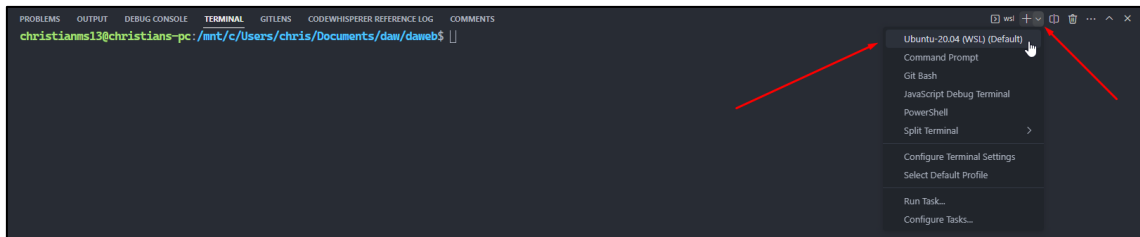


Le damos a Enter y ahora el nuevo bucket aparece en el menú lateral.



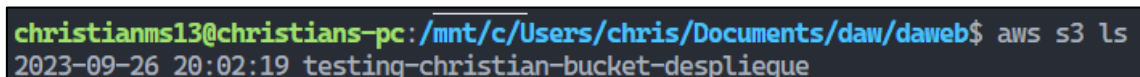
Comprobamos que los comandos de “AWS CLI” funcionan desde el terminal.

Pulsamos “Ctrl + Ñ” en VS Code, lo que abre el terminal del IDE. Seleccionamos el perfil de Ubuntu (se presupone de el subsistema de Ubuntu ya está instalado en la máquina local).



Si no lo hemos hecho antes, ejecutamos aquí el comando “**sudo apt install awscli -y**”, que se encargará de instalar “AWS CLI” en el subsistema de Ubuntu local.

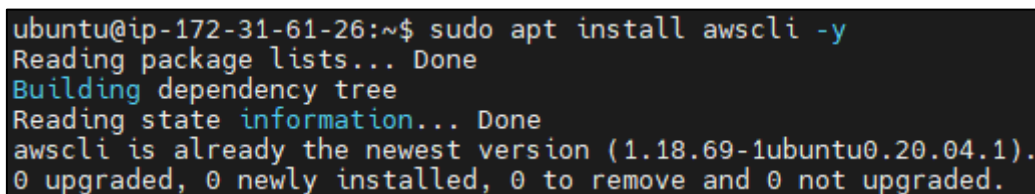
Ahora sí, ejecutamos el comando “**aws s3 ls**”, que nos mostrará una lista de los buckets asociados a nuestra cuenta de AWS.



6. AWS CLI

Para terminar, vamos a instalar “**AWS CLI**” en nuestra máquina remota.

Para ello, volvemos al entorno de trabajo donde tenemos la conexión mediante “SSH”. Desde la línea de comandos del servidor ejecutamos el siguiente comando para instalar “AWS CLI”: “**sudo apt install awscli -y**”.



El comando mencionado instala el paquete “**awscli**” en la máquina.

Para comprobar que el paquete se ha instalado correctamente, podemos utilizar el comando “**aws --version**”, el cual debe generar una respuesta similar a la siguiente:

```
ubuntu@ip-172-31-61-26:~$ aws --version  
aws-cli/1.18.69 Python/3.8.10 Linux/5.15.0-1045-aws botocore/1.16.19
```

Por último, utilizaremos el comando **“aws ec2 describe-instances --output json”** para generar una lista de las instancias “EC2” de nuestra infraestructura en formato “JSON”.

6.1. Configuración de AWS CLI

Pero antes, necesitamos configurar “AWS CLI” para que se conecte con nuestra cuenta de AWS. Para ello, ejecutamos primero el comando **“aws configure”**. Esto abre la configuración del mismo. Se nos solicitan una serie de datos, los cuales obtendremos, de nuevo, desde el workbench de AWS.

Entraremos de nuevo en el botón “AWS Details”, y esta vez nos centraremos en la primera opción que aparece: **“AWS CLI”**. Hacemos clic en el botón “Show”. Aquí tenemos los datos necesarios para conectar “AWS CLI” con nuestra cuenta de AWS.

Lo primero que se nos solicita es el **“Access Key ID”**, el cual se encuentra en la segunda línea del cuadro abierto en el workbench de AWS. Copiamos el contenido de esta línea, situado justo después del “=” (no hay que copiar el texto “aws_access_key_id=”, sino el contenido que aparece después en esa misma línea).

Le damos a Enter.

Lo siguiente es ingresar el **“Secret Access Key”**. Repetimos el mismo método que con el “Access Key ID”, pero esta vez con el contenido de la línea de justo debajo.

Le damos a Enter de nuevo.

Ahora se nos pide la región de nuestra cuenta, que en nuestro caso es la **“us-east-1”**.

Para finalizar, debemos establecer un formato por defecto en el que se mostrarán los datos solicitados, por lo que ingresamos el texto **“json”**.

```
ubuntu@ip-172-31-61-26:~$ aws configure  
AWS Access Key ID [None]: ASIAXUDA7VKU6JZKUVAG  
AWS Secret Access Key [None]: rsl8Y1joYhAh4oP4x7DCDLApgZ1Ua10L0eQ5jtJ6  
Default region name [None]: us-east-1  
Default output format [None]: json
```

Si al ejecutar el comando **“aws ec2 describe-instances --output json”** se produce un error, seguramente es debido a que las credenciales para la conexión no se han configurado de forma correcta.

Otro método es editar el archivo **“credentials”** de “AWS CLI” directamente, a mano.

Para hacer esto, ejecutamos el comando **“nano ~/.aws/credentials”**.

Primero borramos todo el contenido que haya en este archivo.

Lo siguiente es volver a dirigirnos al workbench de AWS, donde copiaremos esta vez absolutamente todo el contenido dentro del cuadro de la opción “AWS CLI” en “AWS Details”.

Esta vez pegaremos todo ese contenido en el archivo “credentials” directamente.

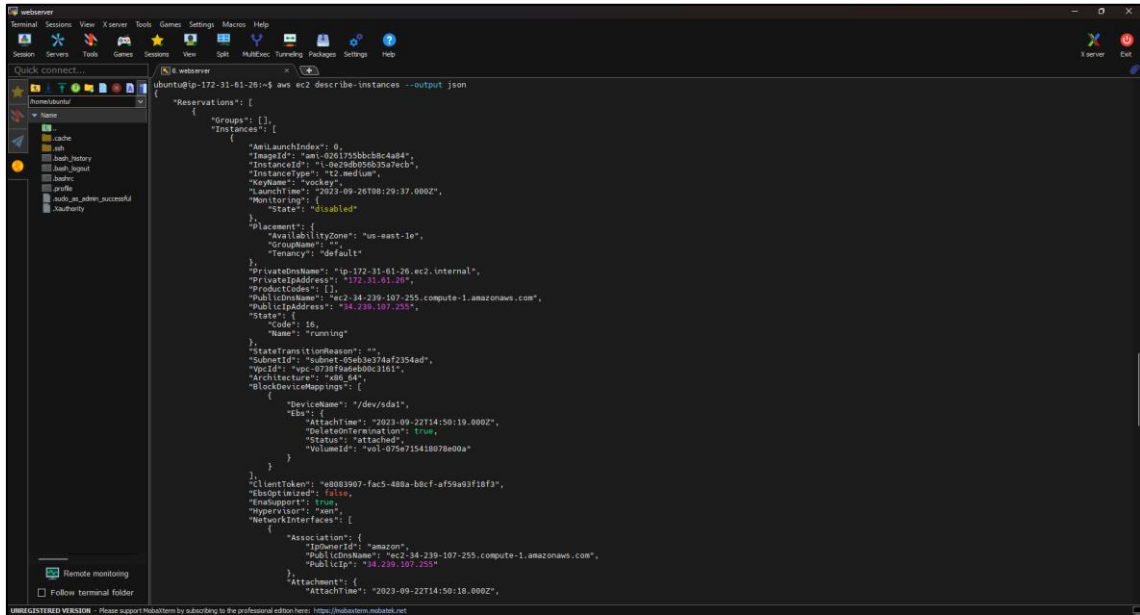
```
GNU nano 4.8 /home/ubuntu/.aws/credentials
[Default]
aws_access_key_id=ASIAXUBA7YKUGJ2KUVAG
aws_secret_access_key=IUV1j0HhAh4p4z7DCDLAppZ1Ua1dL0e051tJ6
aws_session_token=Pw0ZK1VYXdE1//////////wEa0Br+3qFaGJlNFJTNISLCAe+8B3MK/fokd10noCnkBxbMqMs1SMq1hu3sFyg5EZ5AGLveVGKzFo0RSkT9FLuLvuYzzY1wQ13PyuRILwht8DvckmDe+uxnERSdXqpAUqyPp29tksBvxspV/UPrraHBr4J1M0s
```

Guardamos los cambios con “**Ctrl + O -> Enter -> Ctrl + X**”.

6.2. Infraestructura EC2 en formato JSON

Una vez configurada la conexión con AWS, podemos ejecutar el comando “aws ec2 describe-instances --output json”, mencionado anteriormente.

Se muestra la estructura de instancias asociada a nuestra cuenta de AWS en formato “JSON”.

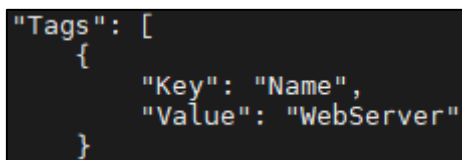


```

ubuntu@ip-172-31-61-26:~$ aws ec2 describe-instances --output json
{
  "Reservations": [
    {
      "Groups": [],
      "Instances": [
        {
          "AmiLaunchIndex": 0,
          "ImageId": "ami-0261555bcb8c4884",
          "InstanceId": "i-0e79d8056b25a7ebc",
          "InstanceType": "m2.medium",
          "KeyName": "vockey",
          "LaunchTime": "2023-09-20T08:29:37.000Z",
          "Monitoring": {
            "State": "disabled"
          },
          "Placement": {
            "AvailabilityZone": "us-east-1a",
            "GroupName": "",
            "Tenancy": "default"
          },
          "PrivateIpAddress": "ip-172-31-61-26.ec2.internal",
          "PrivateIpAddress": "172.31.61.26",
          "ProductCodes": [],
          "PublicIpAddress": "ec2-34-239-107-255.compute-1.amazonaws.com",
          "PublicIpAddress": "34.239.107.255",
          "State": {
            "Code": 16,
            "Name": "running"
          },
          "StateTransitionReason": "",
          "SubnetId": "subnet-95b3e374af2354ad",
          "VpcId": "vpc-0738f9a6b00c161",
          "Architecture": "x86_64",
          "BlockDeviceMappings": [
            {
              "DeviceName": "/dev/sda1",
              "Ebs": {
                "AttachTime": "2023-09-22T14:56:19.000Z",
                "DeleteOnTermination": true,
                "Status": "attached",
                "VolumeId": "vol-075e71541807b00a"
              }
            }
          ],
          "CpuCredits": "standard",
          "EnclaveOptions": {
            "Enabled": false
          },
          "EnclaveSupport": true,
          "Hypervisor": "xen",
          "NetworkInterfaces": [
            {
              "Association": {
                "IpOwnerId": "amazon",
                "PublicIp": "ec2-34-239-107-255.compute-1.amazonaws.com",
                "PublicIp": "34.239.107.255"
              },
              "Attachment": {
                "AttachTime": "2023-09-22T14:56:18.000Z",

```

Aquí podemos ver, entre otros muchos datos interesantes, el nombre de la instancia que hemos creado o su dirección IP.



```

"Tags": [
  {
    "Key": "Name",
    "Value": "WebServer"
  }
]

```

Este contenido se está mostrando directamente en el terminal de comandos, pero podríamos generar un archivo dedicado a almacenar la información que acabamos de recibir para poder trabajar con ella de una forma más concisa.