



## Mi primer EC2

## Índice

<b>ESCENARIO .....</b>	<b>3</b>
<b>INSTANCIANDO .....</b>	<b>4</b>
<b>AWS CLI .....</b>	<b>12</b>
<b>CATEGORÍAS .....</b>	<b>14</b>
<b>TAREAS .....</b>	<b>15</b>

## Escenario

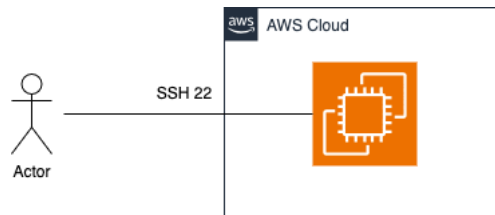
Durante este primer tema vamos a desplegar lo que en programación se dice nuestro “hola mundo”, empezar a familiarizarnos con el panel de control y con nuestro primer servicio.

A lo largo del curso iremos conociendo aquellos servicios que vayamos necesitando. En este primer tema se trata del servicio de EC2, perteneciente a la categoría de computación (al final del documento podréis encontrar un resumen de los servicios y categorías). Si buscamos la definición podríamos encontrar como tal:

*“Amazon EC2 (Elastic Compute Cloud) es un servicio de computación en la nube proporcionado por **AWS (Amazon Web Services)** que permite a los usuarios lanzar y gestionar máquinas virtuales (instancias) de forma escalable y flexible.”*

Traducido podría decirse que son las máquinas virtuales que hemos realizado hasta el momento en clase, con la única diferencia de que estos elementos de computación están expuestos en el mundo real.

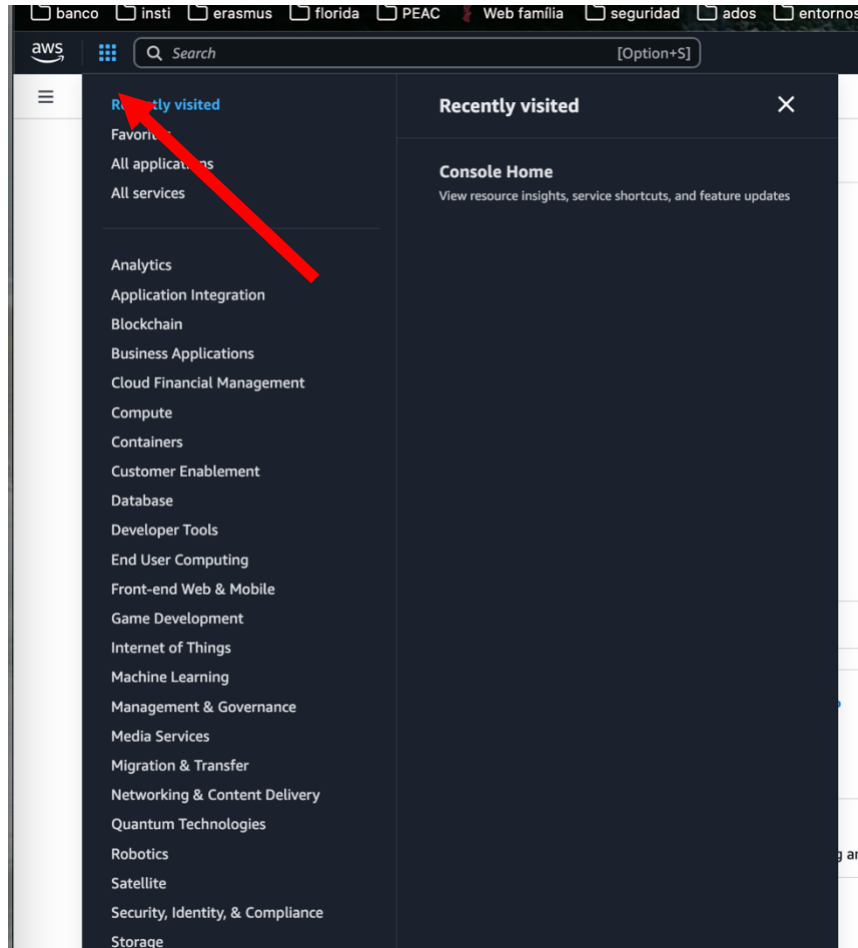
A continuación, muestro el escenario con el que vamos a trabajar durante este primer tema:



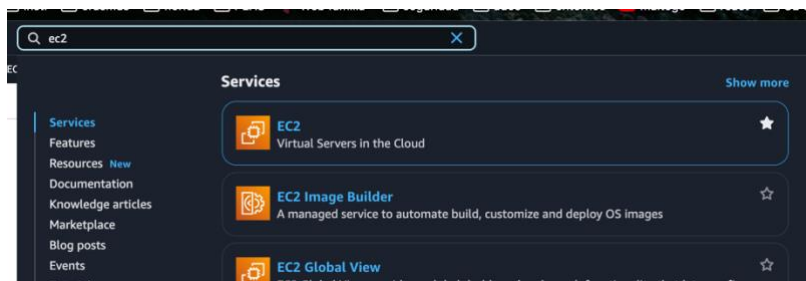
En la nube de AWS es donde vamos a poder desplegar nuestros servicios. Luego éstos serán expuestos al mundo (Internet), debido a que dentro de su estableceremos las reglas de protección. Es importante recalcar que, por defecto TODO está denegado. Si quiero utilizar un puerto, servicio, etc ... deberé autorizarlo explícitamente.

## Instanciando

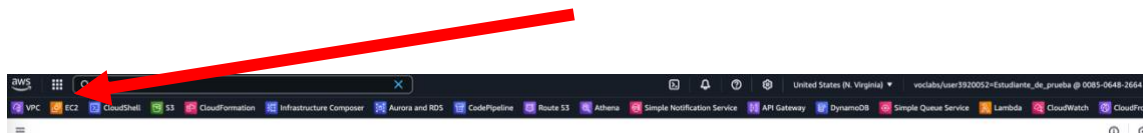
Lo primero es acceder al panel de control. Si hacemos clic en la zona superior izquierda, se despliegan todas las categorías y dentro de ellas los servicios asociados. Os animo a que consultéis la gran cantidad de servicios que hay en algunas categorías.



En la barra de búsqueda teclearemos EC2.



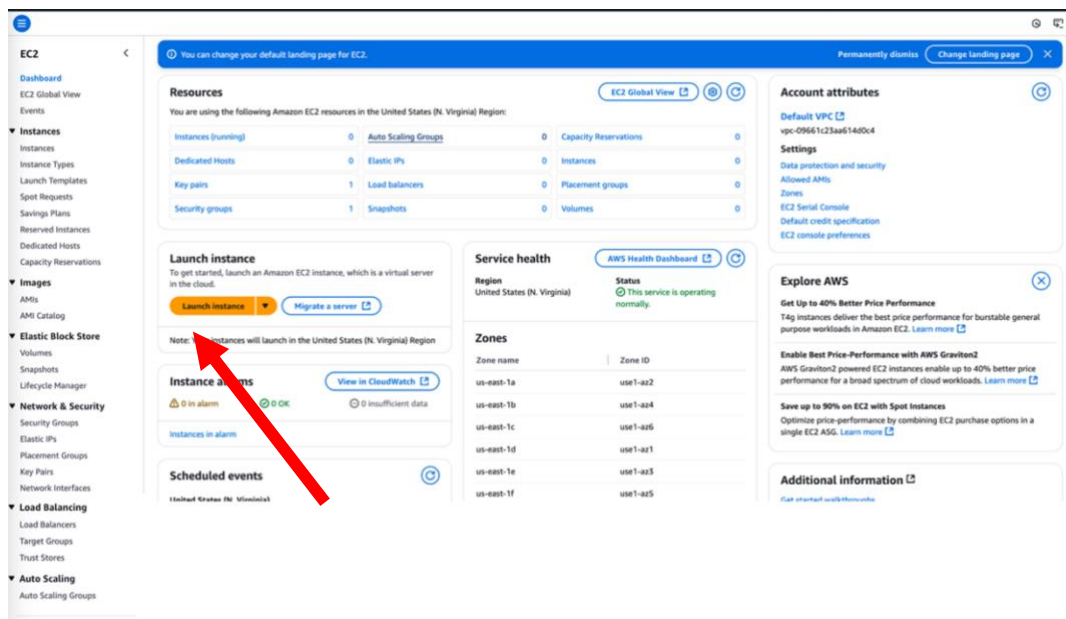
Cuando busquemos el servicio EC2 podremos agregarlo a favoritos y entonces se nos añadirá a la barra de herramientas.



Una vez seleccionado el servicio se nos desplegará el panel de control asociado a éste.

- **Instances:** se muestran las instancias de nuestras AMIs.
- **Images:** las AMI (Amazon Machine Image) son plantillas predefinidas con un sistema operativo y software ya preconfigurado.
- **Elastic Block Store:** es el servicio de almacenamiento que proporciona almacenamiento persistente y que podemos asociar a las instancias EC2.
- **Network & Security:** aquí se definen las reglas de seguridad y un concepto como es el de IP elástica, una IP elástica es una IP pública que podemos asociar/desasociar a cualquier recurso que necesite un IP (conlleva costos asociados).
- **Load Balancing:** servicio de balanceo de carga.
- **Auto Scaling:** servicio de auto escalado.

Para empezar, le daremos a Launch instance



Lo primero es seleccionar la plantilla (AMI) en la cual basaremos nuestro EC2. Ahora seleccionaremos el Sistema Operativo y después, seleccionaremos una AMI entre las que nos ofrece del Sistema Operativo elegido. Para no gastar mucho dinero, sí, hay que estar constantemente pensando cuanto nos vamos a gastar, seleccionaremos una AMI de la capa gratuita (free tier). Aunque también podéis ir probando diferentes configuraciones para ver las ventajas y desventajas de gastar más o menos dinero.

▼ **Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

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

**Quick Start**

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux Debian

**Amazon Machine Image (AMI)**

Amazon Linux 2023 AMI Free tier eligible  
ami-08b5b3a93ed654d19 (64-bit (x86), uefi-preferred) / ami-0eae2a0fc13b15fce (64-bit (Arm), uefi)  
Virtualization: hvm ENA enabled: true Root device type: ebs

Amazon Linux 2023 AMI Free tier eligible ✓  
ami-08b5b3a93ed654d19 (64-bit (x86), uefi-preferred) / ami-0eae2a0fc13b15fce (64-bit (Arm), uefi)  
Virtualization: hvm ENA enabled: true Root device type: ebs

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type Free tier eligible  
ami-04aa00acb1165b32a (64-bit (x86)) / ami-0a90799d0400252c7 (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs

Deep Learning OSS Nvidia Driver AMI GPU PyTorch 2.6 (Amazon Linux 2023)  
ami-06662ef7d02d4e6ed (64-bit (x86)) / ami-0d04ddb7dc152902e (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs

Deep Learning OSS Nvidia Driver AMI GPU TensorFlow 2.16 (Amazon Linux 2)  
ami-02b7f465fe80cb61 (64-bit (x86))  
Virtualization: hvm ENA enabled: true Root device type: ebs

Deep Learning Base OSS Nvidia Driver GPU AMI (Amazon Linux 2023)  
ami-0958bcb162a4e199b (64-bit (x86)) / ami-04745065573a4b195 (64-bit (Arm))

Cuando seleccionemos uno será importante (no para este tema, pero sí para el servicio de Cloudformation y Terraform fijarse en el campo AMI ID). Además, para este tema tenemos el usuario que viene preconfigurado en la máquina.

secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.6.20250303.0 x86\_64 HVM kernel-6.1

Architecture: 64-bit (x86) Boot mode: uefi-preferred AMI ID: ami-08b5b3a93ed654d19 Publish Date: 2025-03-04 Username: ec2-user Verified provider

Ahora, para la AMI elegiremos el tipo de instancia. En este caso, principalmente nos fijamos en las CPUs, RAM y precio por hora (siempre el precio por hora).

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

**Instance type**

t2.micro Free tier eligible  
Family: t2 1 vCPU 1 GiB Memory Current generation: true  
On-Demand Windows base pricing: 0.0162 USD per Hour On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour  
On-Demand SUSE base pricing: 0.0116 USD per Hour On-Demand RHEL base pricing: 0.026 USD per Hour  
On-Demand Linux base pricing: 0.0116 USD per Hour

t2.nano  
Family: t2 1 vCPU 0.5 GiB Memory Current generation: true On-Demand Linux base pricing: 0.0058 USD per Hour  
On-Demand SUSE base pricing: 0.0058 USD per Hour On-Demand Windows base pricing: 0.0081 USD per Hour  
On-Demand Ubuntu Pro base pricing: 0.0076 USD per Hour

t2.micro Free tier eligible ✓  
Family: t2 1 vCPU 1 GiB Memory Current generation: true  
On-Demand Windows base pricing: 0.0162 USD per Hour On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour  
On-Demand SUSE base pricing: 0.0116 USD per Hour On-Demand RHEL base pricing: 0.026 USD per Hour  
On-Demand Linux base pricing: 0.0116 USD per Hour

t2.small  
Family: t2 2 vCPU 2 GiB Memory Current generation: true On-Demand Windows base pricing: 0.032 USD per Hour  
On-Demand Linux base pricing: 0.023 USD per Hour On-Demand RHEL base pricing: 0.0376 USD per Hour  
On-Demand SUSE base pricing: 0.053 USD per Hour On-Demand Ubuntu Pro base pricing: 0.025 USD per Hour

t2.medium  
Family: t2 2 vCPU 4 GiB Memory Current generation: true On-Demand Ubuntu Pro base pricing: 0.0499 USD per Hour  
On-Demand Linux base pricing: 0.0464 USD per Hour On-Demand RHEL base pricing: 0.0752 USD per Hour  
On-Demand Windows base pricing: 0.0644 USD per Hour On-Demand SUSE base pricing: 0.1464 USD per Hour

El siguiente paso es generar una clave o reutilizar una. De forma predeterminada, en la creación del laboratorio ya tenemos el famoso vockey. Esto no es más que el .pem que

encontramos en el apartado de AWS Details en la ventana de creación del lab. Este fichero .pem es la parte pública que necesita el cliente, ya que la privada se encuentra instalada en la instancia de la AMI.

Cuando hagamos clic en *Download PEM*, se descargará el archivo labuser.pem que más adelante ya configuraremos y usaremos.

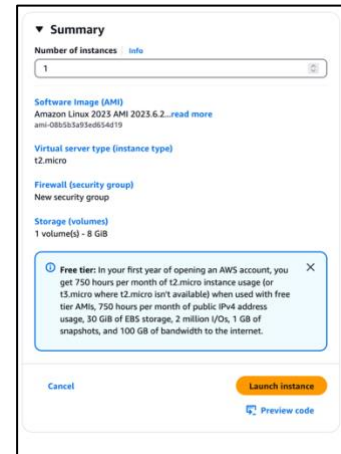
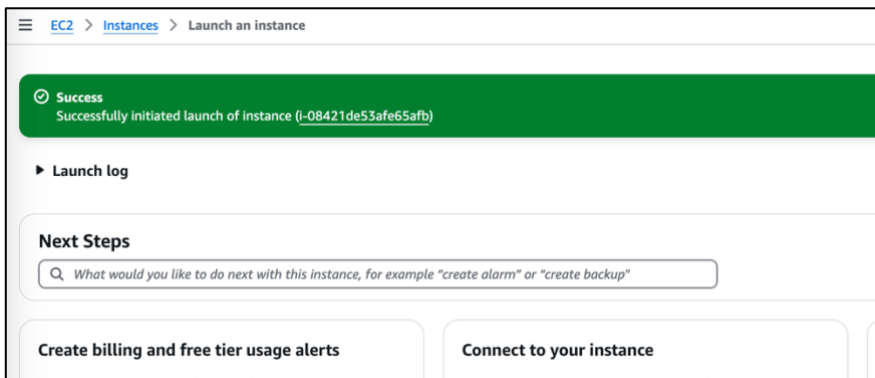
A continuación, empezamos con un aspecto fundamental, que es la parte de la configuración de seguridad. Se lleva a cabo a través de los grupos de seguridad. Un grupo de seguridad no es más que un firewall por detrás en el que, de una forma gráfica, podremos configurar las reglas de entrada y de salida.

Aquí podremos ver que aparece el concepto de VPC y subred, pero eso ya lo dejaremos para el tema siguiente, ahora estamos con el “hola mundo”, y no vamos a complicarlo mucho. Al tener marcada la opción *Allow SSH traffic* from y luego seleccionado *Anywhere*, ya podremos permitir las conexiones SSH. También se puede configurar un acceso más restrictivo, especificando mejor desde dónde se puede acceder, pero eso ya lo iremos configurando cada vez más en detalle.

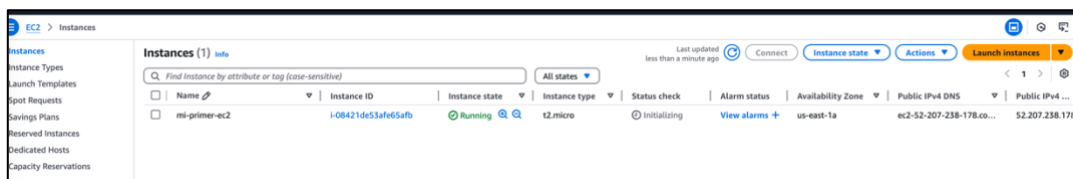
El siguiente paso es el almacenamiento que por defecto viene con un storage de 8GB que, para lo que vamos a trabajar, es más que suficiente.

A la derecha, nos aparece el número de instancias que queremos lanzar con esa configuración, por defecto viene un 1, pero para ciertas situaciones a lo mejor queremos lanzar tres o cuatro instancias a la vez.

Una vez configurada ya podremos lanzar la instancia, y si todo ha ido correcto saldrá un mensaje indicándolo.



Ahora si hacemos clic en **Instances**, podremos ir directamente al panel de control para ver nuestras instancias.





Una vez ya tenemos la instancia en ejecución, necesitaremos ver su configuración porque vamos a probar la conexión a la misma, a ver si realmente tenemos acceso.

The screenshot shows the AWS Management Console interface for an EC2 instance. The instance is named 'mi-primer-ec2' and is in the 'Running' state. The 'Public IPv4 address' is highlighted with a red box, showing the value '52.207.238.178'. The 'Public IPv4 DNS' is also highlighted with a red box, showing the value 'ec2-52-207-238-178.compute-1.amazonaws.com'. Other details visible include the instance ID 'i-08421de53afe65afb', the instance type 't2.micro', and the VPC ID 'vpc-09661c23aa614d0c4'.

Respecto a los grupos de seguridad, podemos comprobar el grupo creado y sus opciones.

The screenshot shows the AWS Management Console interface for a security group named 'launch-wizard-1'. The security group is in the 'default' VPC. The 'Rules of entry' tab is selected, showing a single rule for SSH access (port 22) from any IP address (0.0.0.0/0). A red arrow points from the 'Launch instance' button in the left sidebar to the security group details page.

Grupos de seguridad (1/2) <a href="#">Información</a>					
<input type="text" value="Find resources by attribute or tag"/>					
<input type="checkbox"/>	Name	ID de grupo de seguridad	Nombre del grupo de seguridad	ID de la VPC	Descripción
<input checked="" type="checkbox"/>	-	sg-0f62eca023164db71	launch-wizard-1	vpc-09661c23aa614d0c4 <a href="#">🔗</a>	launch-wizard-1 created 2025-03
<input type="checkbox"/>	-	sg-0e193835542b2a3c6	default	vpc-09661c23aa614d0c4 <a href="#">🔗</a>	default VPC security group

```
chmod 400 labuser.pem
```

[illegible]

Y ahora ya desde aquí podemos comprobar la ip privada a ver si es realmente como la que nos da el panel de control aws.

The screenshot shows a terminal window with the following content:

```

jorge@mjolnir Downloads % chmod 400 labsuser.pem
jorge@mjolnir Downloads % ssh -i labsuser.pem ec2-user@52.207.238.178
The authenticity of host '52.207.238.178 (52.207.238.178)' can't be established.
ED25519 key fingerprint is SHA256:shYkCdapRrpe8tCCeY2SP47kg9tI8Mvviczq6P//QCo.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '52.207.238.178' (ED25519) to the list of known hosts.

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-172-31-80-38 ~]$ 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 noprefixroute
        valid_lft forever preferred_lft forever
2: enx0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc fq_codel state UP group default qlen 1000
    link/ether 12:8B:47:99:8C:7F brd ff:ff:ff:ff:ff:ff
    altname eni-0547fe924f243246e
    altname device-number-0.0
    inet 172.31.80.38/20 scope global dynamic enx0
        valid_lft 3499sec preferred_lft 3499sec
    inet6 fe80::1003:47ff:fe99:8c7f/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
[ec2-user@ip-172-31-80-38 ~]$
  
```

The AWS console sidebar on the right shows the instance details for 'us-east-1a' in the 'us-east-1a' region. Under 'Private IPv4 addresses', it lists the IP '172.31.80.38'. Under 'Public IPv4 DNS', it lists 'ec2-52-207-238-178.compute-1.amazonaws.com'.

Ahora una vez ya hemos acabado tenemos varias opciones.

- Parar el laboratorio
- Dejar que se acaben las 4 horas
- Resetear el lab. Esta opción conlleva la eliminación de todos los recursos creados.

## AWS CLI

Ahora que ya hemos practicado con el entorno gráfico, vamos a comenzar a utilizar la terminal. Para ello, tal y como se explicó en el tema 0, vamos a configurar las credenciales en el archivo credentials para poder estar conectados con el lab.

Una parte muy importante a tener en cuenta es la documentación oficial, TODO está allí. El gran problema es lograr encontrar aquello que buscamos, pero con tiempo y paciencia se encuentra siempre.

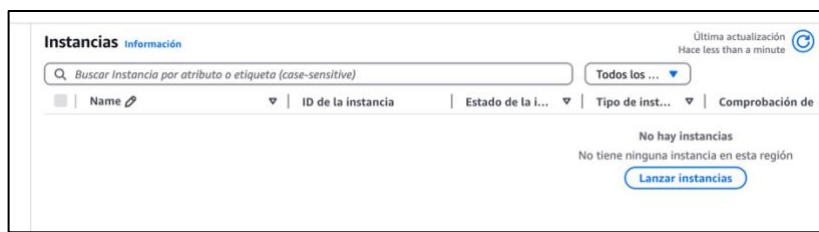
La página oficial de referencia es:

<https://docs.aws.amazon.com/>

Y más concretamente para el servicio EC2

<https://docs.aws.amazon.com/ec2/>

Antes de lanzar el comando que crea mi EC2, compruebo las instancias que hay:



Vemos que no hay ninguna instancia creada.

Ahora el objetivo principal va a ser probar un código de ejemplo para comprobar si realmente funciona. El código que vamos a probar es el siguiente:

```
$ ec2_1.sh x
1  aws ec2 run-instances \
2  --image-id ami-04b4f1a9cf54c11d0 \
3  --count 1 \
4  --instance-type m1.small \
5  --key-name vockey \
6  --region us-east-1
```

Este código se encuentra en el script ec2\_1.sh, que podéis encontrar en Aules.

Ejecuta el script desde el terminal. Antes de ejecutarlo deberás darle permisos de ejecución al archivo. En el momento que se lanza, si no hay errores mostrará algo parecido a la siguiente captura (es la configuración que lanza del ec2).

En la consola, nos devuelve la información del EC2 que ha creado en formato json. Podemos guardarnos esta información para poder acceder a la máquina posteriormente, ya que nos está proporcionando los datos necesarios para poder realizar la conexión (ip y dnis públicos), entre otros.

```
{
  "Architecture": "x86_64",
  "BlockDeviceMappings": [
    {
      "DeviceName": "ephemeral0",
      "Ebs": {
        "AttachTime": "2025-03-14T15:48:48.000Z",
        "ClientToken": "eni-attach-07ab2e78563f890e1",
        "DeleteOnTermination": true,
        "DeviceIndex": 0,
        "Status": "attaching",
        "NetworkCardIndex": 0
      }
    }
  ],
  "Description": "",
  "Groups": [
    {
      "GroupId": "sg-0e193835542b2a3c6",
      "GroupName": "default"
    }
  ],
  "Ipv4Addresses": [
    {
      "MacAddress": "12:5d:58:31:4b:dd",
      "NetworkInterfaceId": "eni-0686131f579762f2b",
      "OwnerId": "0008506482664",
      "PrivateDnsName": "ip-172-31-90-140.ec2.internal",
      "PrivateIpAddress": "172.31.90.140",
      "PrivateIpAddresses": [

```

Otra forma de comprobar que la máquina se ha creado correctamente es acceder al listado de las instancias EC2 desde la consola de AWS, como ya hicimos al crear la primera máquina EC2 desde el entorno gráfico un poco antes, en este mismo tema.

Instancias (1) Información					
<input type="text"/> Buscar instancia por atributo o etiqueta (case-sensitive) <span>Todos los ...</span>					
	Name	ID de la instancia	Estado de la i...	Tipo de inst...	Comprobación de
<input type="checkbox"/>		i-00acf115cbfea80d3	En ejecución	m1.small	Inicializando

Y, haciendo clic en la id de la instancia, podríamos obtener toda la información relacionada con la máquina que acabamos de crear.

## Categorías

 Análisis	 Integración de aplicaciones	 RA y RV	 Blockchain	 Aplicaciones empresariales	 Cómputo
 Administración de costos	 Interacción con clientes	 Base de datos	 Herramientas para desarrolladores	 Informática para usuarios finales	 Tecnología para juegos
 Internet de las cosas	 Machine learning	 Administración y gobernanza	 Servicios multimedia	 Migración y transferencia	 Móvil
 Redes y entrega de contenido	 Robótica	 Satélite	 Identidad de seguridad y cumplimiento	 Almacenamiento	

## Tareas

En un archivo pdf con el formato tarea1\_nombre-apellidos.pdf adjunta:

1. Captura de la versión AWS CLI instalada en el tema 0.
2. Captura del identity realizada en el tema 0.
3. Replica la creación del ec2, pero en vez de conectarte a un AMI Linux por ssh hazlo a una máquina Ubuntu, adjunta sólo la captura final de conexión realizada.
4. Replica la creación del ec2 usando la terminal (AWS CLI) mediante script, pero en vez de un Ubuntu coge la ami de un Debian.