

Manual de Instalación

Sistema MotionParts

Tabla de Contenidos

1. [Introducción](#)
 2. [Requisitos Previos](#)
 3. [Preparar Instancia en AWS](#)
 4. [Instalación de Docker en Ubuntu](#)
 5. [Configuración del Proyecto](#)
 6. [Certificado SSL](#)
 7. [Acceso y Verificación](#)
-

Introducción

Este manual describe el proceso completo de instalación y configuración del sistema MotionParts en un servidor Ubuntu utilizando Docker como plataforma de contenedores y Caddy como servidor web con SSL automático.

El sistema consta de tres componentes principales:

- **Base de datos:** MongoDB
 - **Backend:** API REST en Spring Boot (Puerto 8080)
 - **Frontend:** Aplicación Angular (Puerto 4200)
 - **Portainer:** Herramienta de monitoreo de contenedores (Puerto 9000)
-

Requisitos

Antes de comenzar la instalación, asegúrese de contar con:

- Servidor Ubuntu 20.04 o superior
- Acceso root o usuario con privilegios sudo
- Conexión a Internet estable
- Dominio configurado (por ejemplo: dominio.org o servicio DuckDNS)

- Puertos abiertos: 80, 443, 4200, 8080, 9000

Preparar Instancia en Canvas - AWS

1. Iniciar sesión en Canvas

1. Inicie sesión en Canvas con las credenciales otorgadas y dirígete al aula, te debe aparecer esta pantalla a continuación:

AWS Academy Learner Lab proporciona un entorno de pruebas prolongado para la exploración a medida de los servicios de AWS. En esta clase, los alumnos tendrán acceso a un conjunto restringido de servicios de AWS. No todos los laboratorios de maestros o tutoriales de documentación de AWS que operan en la cuenta de producción de AWS funcionarán en el entorno del laboratorio de aprendizaje. Conservará el acceso a los recursos de AWS configurados en este entorno mientras dure este curso. Limitamos su presupuesto (50 USD), por lo que debe tener cuidado a la hora de hacer gastos que lo agoten muy rápidamente. Si se pasa del presupuesto, perderá el acceso a su entorno y todo su trabajo.

Todas las sesiones duran 4 horas de forma predeterminada, aunque puede extenderlas haciendo clic en el botón iniciar para reiniciar el temporizador de la sesión. Al final de cada sesión, cualquier recurso que haya creado se guardará. Sin embargo, las instancias EC2 se apagan de forma automática. Otros recursos, como las instancias de Relational Database Service (RDS), se siguen ejecutando. Tenga en cuenta que algunas funciones de AWS no se detienen, por lo que pueden generar gastos entre una sesión y la siguiente. Por ejemplo, un Elastic Load Balancer o una NAT. Es posible que prefiera borrar estos tipos de recursos y volver a crearlos a medida que los necesite para poner a prueba su trabajo durante una sesión. Tendrá acceso a este entorno mientras dure la clase en la que se ha inscrito. Cuando la clase termine, el acceso al laboratorio de aprendizaje también terminará.

Solo para el educador/profesor

Si usted es un educador que utiliza un laboratorio de aprendizaje en su curso, consulte el área Recursos de la página de inicio del portal de AWS Academy para acceder a la lista de servicios compatibles para cada clase del laboratorio de aprendizaje. Este entorno de pruebas está orientado para trabajo de proyecto, ejercicios de laboratorio o prácticas diseñados por un educador que se crean y prueban dentro de un laboratorio de aprendizaje.

Empezar

2. Seleccione la sección de módulos y dale click a “Laboratorio de aprendizaje de AWS academy”

The screenshot shows the AWS Academy Learning Lab interface. On the left, there's a sidebar with navigation links: Página de Inicio, Módulos, Foros de discusión, Clasificaciones, Lucid (pizarra), Calendario, Bandeja de entrada, Historial, and Ayuda. The main content area displays the 'Guía del estudiante del Laboratorio de aprendizaje de AWS Academy' page. It includes sections like 'Laboratorio de aprendizaje de AWS Academy: cumplimiento y seguridad', 'Cómo usar de manera eficaz el Laboratorio de aprendizaje de Academy', 'Evaluación de conocimientos del módulo', 'Laboratorio de aprendizaje de AWS Academy', 'Iniciar el Laboratorio de aprendizaje de AWS Academy', 'Recursos de los laboratorios de aprendizaje de AWS Academy', 'Demostración: cómo acceder al Laboratorio de aprendizaje', 'Demostración: sugerencias para la solución de problemas generales', 'Demostración: cómo iniciar servicios a través de la Consola de AWS', and 'Actividad de laboratorio para alumnos: desarrollador de Amazon Q'. At the bottom, there's an 'Encuesta de comentarios' section.

3. Dentro del laboratorio, debes darle click “Start lab”

The screenshot shows the AWS Academy Learner Lab environment. The top bar indicates 'Used \$15.2 of \$50' and a timer '00:00'. There are buttons for 'Start Lab', 'End Lab', 'AWS Details', 'Readme', and 'Reset'. The main area is titled 'Learner Lab' and contains sections for 'Environment Overview', 'Environment Navigation', 'Access the AWS Management Console', 'Region restriction', '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'. A note at the bottom says 'Instructions last updated: 2025-06-24'. Below this, the 'Environment Overview' section states: 'This Learner Lab provides a sandbox environment for ad-hoc exploration of AWS services.' and 'This environment is long-lived. When the session timer runs to 0:00, the session will end, but any data'.

El botón de color rojo “AWS” debe cambiar a color verde, el laboratorio ya está activo, a continuación, haga click en el botón verde “AWS”, esto lo redireccionara a AWS

This screenshot is identical to the previous one, showing the 'Learner Lab' interface. The difference is that the 'AWS' button at the top has turned green, indicating the lab is now active. The rest of the interface remains the same, displaying the environment overview and usage details.

4. Dentro de AWS, en el buscador debe buscar el servicio “EC2”

The screenshot shows the AWS search interface for 'EC2'. The search bar at the top contains 'EC2'. Below it, there are three main sections:

- Servicios (13)**: Includes EC2, EC2 Image Builder, and AWS Global View.
- Características (58)**: Includes EC2 Instances, Panel, and AMI.
- Publicaciones de blog (47)**: Includes 'Migración de máquinas virtuales (VMs) de VMWare a Amazon EC2 con el Agente de Réplica de AWS Application Migration Service' and 'Prácticas recomendadas de almacenamiento para cargas de trabajo de producción en Amazon RDS o Amazon EC2'.

At the bottom left, there's a link '¿Tienes más resultados?' with buttons 'Si' and 'No'.

Al darle click, aparecerá esta pantalla a continuación:

This screenshot shows the 'Lanzar la instancia' (Launch instance) step of the EC2 instance creation wizard. The left sidebar shows navigation links like 'Panel', 'Instancias', 'Imagenes', etc. The main area has the following sections:

- Recursos**: Shows 1 instance (en ejecución), 2 load balancers, 0 capacity reservations, 2 security groups, 1 instance, and 1 volume.
- Lanzar la instancia**: Contains a 'Lanzar la instancia' button and a 'Migrar un servidor' link.
- Alarms de instancia**: Shows 0 alarms and 0 data insufficient.
- Eventos programados**: Shows 'Estados Unidos (Norte de Virginia)' and 'No hay eventos programados'.
- Estado del servicio**: Shows the service is functioning normally.
- Atributos de la cuenta**: Includes VPC predefinida (vpc-095f71b7bae9e26d503), Configuración (Protección y seguridad de datos, AMI permitida, Zonas, Especificación de crédito predefinida, Preferencias de la consola de EC2), and Panel de AWS Health.
- Información adicional**: Includes 'Este servicio funciona con normalidad.', 'Instrucciones para comenzar', 'Guía de introducción', 'Documentación', 'Todos los recursos de EC2', 'Foros', 'Precios', and 'Póngase en contacto con nosotros'.

En la sección “Lanzar la instancia”, dar click al botón naranja “Lanzar la instancia”

2. Configuración de la instancia

1. Asignar un nombre a la instancia, en la sección “Imagenes de aplicaciones y sistemas operativos (imagen de maquina de Amazon)” seleccionar Ubuntu

Lanzar una instancia Información

Amazon EC2 le permite crear máquinas virtuales, o instancias, que se ejecutan en la nube de AWS. Comience rápidamente siguiendo los sencillos pasos que se indican a continuación.

Nombre y etiquetas Información

Nombre: MotionParts [Agregar etiquetas adicionales](#)

Imagenes de aplicaciones y sistemas operativos (Imagen de máquina de Amazon) Información

Una AMI posee el sistema operativo, el servidor de aplicaciones y las aplicaciones de la instancia. Si a continuación no ve una AMI adecuada, utilice el campo de búsqueda o elija [Buscar más AMI](#).

Busque en nuestro catálogo completo que incluye miles de imágenes de sistemas operativos y aplicaciones

Recientes | **Inicio rápido**

Amazon Linux	macOS	Ubuntu	Windows	Red Hat	SUSE Linux	Debian

Imagenes de máquina de Amazon (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
ami-0dec6295f9fb5d549 (64 bits x86) / ami-01991f170-43725de (64 bits (AMI))
Virtualización: hvm Activado para EBS: true Tipo de dispositivo: raid-0

Descripción

Ubuntu Server 24.04 LTS (HVM)EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Resumen

Número de instancias: 1

Imagen de software (AMI)
Canonical, Ubuntu, 24.04, amd64...más información
ami-0dec6295f9fb5d549

Tipo de servidor virtual (tipo de instancia)
t3.micro

Firewall (grupo de seguridad)
Nuevo grupo de seguridad

Almacenamiento (volúmenes)
Volumenes: 1 (8 GiB)

[Cancelar](#) Lanzar instancia [Código de versión preliminar](#)

Más abajo se encontrará con la sección “Tipo de instancia”, se debe seleccionar “t2.large”

Tipo de instancia Información | Obtener asesoramiento

Tipo de instancia

t2.large
Familia: t2 2 vCPU 8 GiB Memoria Generación actual: true
Bajo demanda Windows base precios: 0.1208 USD por hora Bajo demanda RHEL base precios: 0.1216 USD por hora
Bajo demanda SUSE base precios: 0.1928 USD por hora Bajo demanda Ubuntu Pro base precios: 0.0963 USD por hora
Bajo demanda Linux base precios: 0.0928 USD por hora

Se aplican costos adicionales a las AMI con software preinstalado

Todas las generaciones

[Comparar tipos de instancias](#)

En la sección “Par de claves”, se deben crear un nuevo par de claves en el botón “Crear un nuevo par de claves”, es importante para poder acceder a la máquina

Tipos de instancia

t2.large
Familia: t2 2 vCPU 8 GiB Memoria Generación actual: true
Bajo demanda Windows base precios: 0.1208 USD por hora Bajo demanda RHEL base precios: 0.1216 USD por hora
Bajo demanda SUSE base precios: 0.1928 USD por hora Bajo demanda Ubuntu Pro base precios: 0.0963 USD por hora
Bajo demanda Linux base precios: 0.0928 USD por hora

Par de claves (inicio de sesión) Información

Puede utilizar un par de claves para conectarla de forma segura a la instancia. Asegúrese de que tiene acceso al par de claves seleccionado antes de lanzar la instancia.

Nombre del par de claves - obligatorio
 Seleccionar [Crear un nuevo par de claves](#)

Configuraciones de red Información

Red Información
vpc-095717bc9ae26d503

Subred Información
Sin preferencias (subred predeterminada en cualquier zona de disponibilidad)

Asignar automáticamente la IP pública Información

Habilitar

Firewall (grupos de seguridad) Información

Un grupo de seguridad es un conjunto de reglas de firewall que controlan el tráfico de la instancia. Agregue reglas para permitir que un tráfico específico llegue a la instancia.

Crear grupo de seguridad Seleccionar un grupo de seguridad existente

Crearemos un nuevo grupo de seguridad denominado "launch-wizard-2" con las siguientes reglas:

Permitir el tráfico de SSH desde Anywhere a establecer conexión con la instancia

Anywhere a establecer conexión con la instancia

Resumen

Número de Instancias: 1

Imagen de software (AMI)
Canonical, Ubuntu, 24.04, amd64...más información
ami-0dec6295f9fb5d549

Tipo de servidor virtual (tipo de instancia)
t2.large

Firewall (grupo de seguridad)
Nuevo grupo de seguridad

Almacenamiento (volúmenes)
Volumenes: 1 (8 GiB)

[Cancelar](#) Lanzar instancia [Código de versión preliminar](#)

se asigna un nombre con la siguiente configuración:

Crear par de claves

Nombre del par de claves
Con los pares de claves es posible conectarse a la instancia de forma segura.

MotionParts|
El nombre puede incluir hasta 255 caracteres ASCII. No puede incluir espacios al principio ni al final.

Tipo de par de claves

- RSA
Par de claves pública y privada cifradas mediante RSA
- ED25519
Par de claves privadas y públicas cifradas ED25519

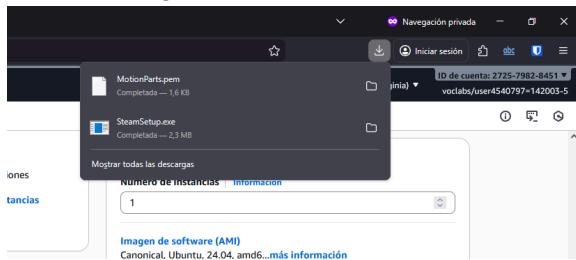
Formato de archivo de clave privada

- .pem
Para usar con OpenSSH
- .ppk
Para usar con PuTTY

Advertencia: Cuando se le solicite, almacene la clave privada en un lugar seguro y accesible del equipo. Lo necesitará más adelante para conectarse a la instancia. [Más información ↗](#)

Cancelar **Crear par de claves**

Esto descargara un archivo .pem, este es nuestro par de claves



Nota: Resguardar y proteger muy bien el archivo, en caso de pérdida o eliminación del archivo, va a ser imposible acceder a la instancia y es necesario crear una nueva instancia con claves nuevas

En la sección “configurar almacenamiento” se debe dejar como lo muestra la imagen a continuación:

▼ Configurar almacenamiento [Información](#) [Avanzado](#)

1x GiB Volumen raíz, 3000 IOPS, No cifrado

Agregar un nuevo volumen

La AMI seleccionada contiene volúmenes de almacenamiento de instancias; sin embargo, la instancia no permite dichos volúmenes. Por lo que no se podrá obtener acceso a ninguno de estos volúmenes de la AMI desde la instancia

Haga clic en actualizar para ver la información de la copia de seguridad [C](#)
Las etiquetas que asigne determinan si alguna política de Data Lifecycle Manager realizará una copia de seguridad de la instancia.

0 x sistemas de archivos [Editar](#)

Antes de lanzar la instancia, verificamos el resumen para asegurarnos de que todo coincide perfectamente con lo hecho en los pasos anteriores:

▼ Resumen

Número de instancias | [Información](#)

1

Imagen de software (AMI)
Canonical, Ubuntu, 24.04, amd64...[más información](#)
ami-0ecb62995f68bb549

Tipo de servidor virtual (tipo de instancia)
t2.large

Firewall (grupo de seguridad)
Nuevo grupo de seguridad

Almacenamiento (volúmenes)
Volúmenes: 1 (25 GiB)

[Cancelar](#) [Lanzar instancia](#) [!\[\]\(1e00971b1cb3637dd13a1b22b0195241_img.jpg\) Código de versión preliminar](#)

Si todo es correcto, se puede dar click al botón de “Lanzar instancia”

Esperar a que se inicialice correctamente, debe aparecer como la imagen a continuación:

La instancia se creará e inicializará “comprobación de estado”, esto tomará unos minutos

Una vez “comprobación de estado” pase de “inicializando” a “2/2 comprobaciones superadas” la instancia estará lista para usarse:

3. Configuración de grupo de seguridad

1. Seleccione la instancia creada abajo aparecerán varias opciones y especificaciones de la instancia, seleccione Seguridad, la instancia tiene un grupo de seguridad asignado

The screenshot shows the AWS EC2 Instances details page for an instance named "MotionParts". The instance is currently running ("En ejecución") and is an "t2.large" type. It is located in the "us-east-1d" availability zone and has the IP address "44.222.203.224". The "Seguridad" tab is selected, showing the "ID del propietario" as "272579828451" and the "Grupos de seguridad" as "sg-082f567e3c9d441 (launch-wizard-2)". Below this, the "Reglas de entrada" section is visible.

Le das click al grupo de seguridad o te diriges al panel ubicado en la izquierda, en la sección “Red y seguridad” le das click a la opción “Security Groups” o “Grupos de seguridad”

The screenshot shows the AWS EC2 Security Groups page. It lists three security groups: "launch-wizard-2" (selected), "default" (VPC security group), and "launch-wizard-1" (created in 2025-10-29T07:27Z). The "Reglas de entrada" tab is selected for the "sg-082f567e3c9d441 - launch-wizard-2" group, showing one rule: "sg-0580f00836c4234ec" (IPv4, TCP port 22, origin anywhere).

Seleccione el grupo de seguridad correspondiente y te vas a la opción de “Reglas de entrada” y editamos las reglas de entrada

2. Al editar las reglas de entrada se deben agregar unas reglas específicas del proyecto para que funcione correctamente

Le das al botón “Agregar regla” con la siguiente configuración

Tipo: TCP personalizado

Intervalo de puertos: puertos del proyecto (4200, 8080, 9000, 80, 22, 443)

Origen: Anywhere IPv4

Descripción: Mencionar a qué se dedica el puerto

Debe verse de la siguiente manera:

Le das al botón “Guardar reglas” y el grupo de seguridad estaría configurado

4. Configuración de IP elástica

Configurar una IP elástica (fija) es importante porque AWS suele cambiar la IP a la cual puedes acceder, para facilitar esto se asigna una IP estática a continuación

1. En la sección “Red y seguridad” te diriges al apartado “Direcciones IP elásticas” o “Elastic IP” y aparecerá la pantalla a continuación

Le das al botón “Asignar dirección IP elástica”

En la pantalla a continuación dar click al botón “Asignar” para crear la nueva dirección IP

2. Verificas que la dirección haya sido creada, si es así seleccionas la dirección correspondiente y te vas al botón “Acciones” y se desplegará un menú, le das a la opción “Dirección IP elástica asociada”

Te redirigirá a la siguiente pantalla, en la sección “Instancia” seleccione la instancia que creamos, en la sección “Dirección IP privada” se selecciona la única opción disponible y le das al botón “Asociado”

La asignación fue exitosa si aparece el mensaje a continuación

5. Conectarse a la instancia

1. Una vez configurada la instancia, para conectarse se selecciona la instancia a la que se desea conectar y le das al botón “Conectar”

The screenshot shows the AWS EC2 Instances page. There are two instances listed:

Name	ID de la instancia	Estado de la i...	Tipo de inst...	Comprobación de...	Estado de la al...	Zona de dispon...	DNS de IPv4 pública	Dirección IP...	IP elástica
Motion	i-0170b39095a42209a	En ejecución	t2.large	2/2 comprobado	Ver alarmas +	us-east-1d	ec2-3-89-17-218.comp...	3.89.17.218	3.89.17.218
MotionParts	i-0f0832ff852e308d3	En ejecución	t2.large	2/2 comprobado	Ver alarmas +	us-east-1d	ec2-44-222-203-224.co...	44.222.203.224	-

The Motion instance is selected. A modal window titled "Instancias (1/2) Información" displays the following details for the Motion instance:

- Detalles:** ID de la instancia: i-0f0832ff852e308d3 (ejemplo), Rol de IAM: sg-02f1567a5cf9d441 (launch-wizard-2).
- Seguridad:** ID del propietario: 272579828451, Hora de lanzamiento: Tue Dec 02 2025 14:58:39 GMT-0500 (hora estándar de Colombia).
- Reglas de entrada:** Filtro reglas.

Se redireccionará a esta pantalla, seleccione la opción “Cliente SSH”

The screenshot shows the "Connect to instance" dialog box. The "Client SSH" tab is selected. The "Conexión de la instancia EC2" section shows the instance ID: i-0f0832ff852e308d3 (ejemplo). Below it, there is a list of steps to connect via SSH:

- Abra un cliente SSH.
- Localice el archivo de clave privada. La clave utilizada para lanzar esta instancia es MotionParts.pem.
- Proporcione el comando, incluyendo la clave, para garantizar que la clave no se pueda ver públicamente.

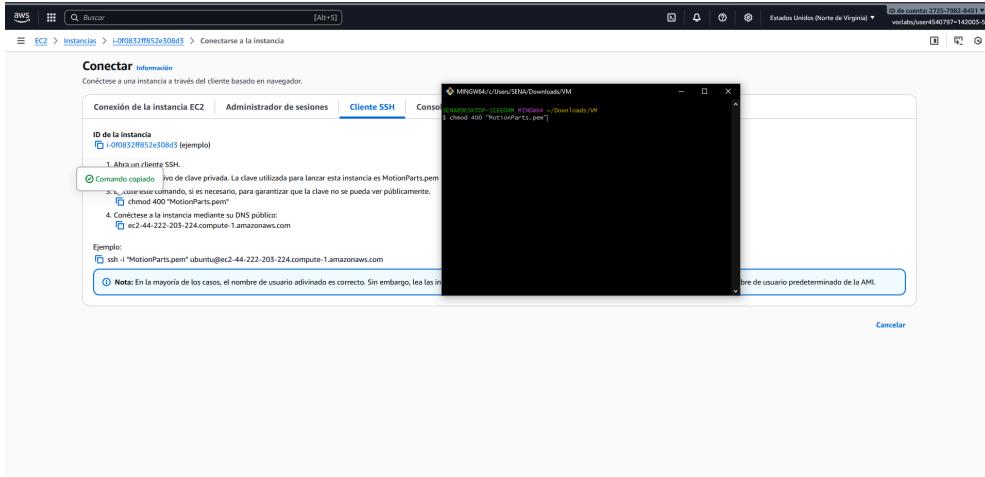
Below the steps, there are two options:

- Conectarse a la instancia mediante su DNS público: ec2-44-222-203-224.compute-1.amazonaws.com
- Conectarse a la instancia mediante su dirección IP: ec2-44-222-203-224

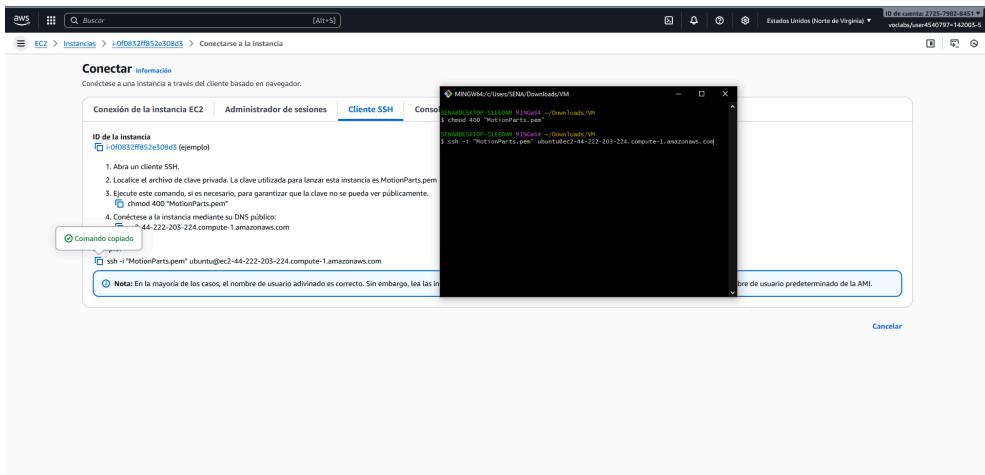
A note at the bottom states: "Nota: En la mayoría de los casos, el nombre de usuario adivinado es correcto. Sin embargo, lea las instrucciones de uso de la AMI para comprobar si el propietario de la AMI ha cambiado el nombre de usuario predeterminado de la AMI." A "Cancelar" button is at the bottom right.

2. Se abre una terminal Git Bash en la ubicación donde se almacenó el archivo .pem. Ejecuta este comando, si es necesario, para garantizar que la clave no se pueda ver públicamente.

chmod 400 "MotionParts.pem"



Conéctese a la instancia mediante su DNS público mediante el siguiente comando brindado por AWS: `ssh -i "MotionParts.pem" ubuntu@ec2-98-87-244-212.compute-1.amazonaws.com`



Nota: Para acceder desde windows, usar una terminal [Git Bash](#)

La conexión fue exitosa si te aparece lo siguiente

```
sanrio@sanrio-PC:~/Descargas/Archivos/AWS$ ssh -i "MotionParts.pem" ubuntu@ec2-98-87-244-212.compute-1.amazonaws.com
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

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

System information as of Wed Dec  3 02:42:50 UTC 2025

System load:  0.19      Processes:          136
Usage of /:   20.2% of 23.17GB  Users logged in:     0
Memory usage: 11%
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

33 updates can be applied immediately.
24 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Tue Dec  2 23:33:52 2025 from 186.154.187.140
ubuntu@ip-172-31-25-185: ~
```

Instalación de Docker en Ubuntu

Paso 1: Actualizar el Sistema

```
sudo apt-get update
```

Paso 2: Instalar Dependencias

```
sudo apt-get install ca-certificates curl
```

Paso 3: Crear Directorio para Llaves

```
sudo install -m 0755 -d /etc/apt/keyrings
```

Paso 4: Agregar la Llave GPG de Docker

```
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o  
/etc/apt/keyrings/docker.asc
```

```
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

Paso 5: Agregar el Repositorio de Docker

1.

```
echo \
```

```
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]  
https://download.docker.com/linux/ubuntu \
```

```
$(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}")  
stable" | \
```

```
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

2. sudo apt-get update

Paso 6: Instalar Docker

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin  
docker-compose-plugin
```

Paso 7: Verificar la Instalación

```
docker -v
```

Debería mostrar la versión de Docker instalada.

Ejemplo:

```
Docker version 29.1.2, build 890dcca
```

Paso 8: Configurar Permisos de Usuario (Opcional)

Si al ejecutar `docker ps` aparece un error de permisos, tiene dos opciones:

Opción A: Usar `sudo` antes de cada comando docker

Opción B: Agregar su usuario al grupo docker (recomendado)

```
sudo usermod -aG docker ubuntu
```

Para aplicar los cambios sin reiniciar:

```
newgrp docker
```

O reinicie la sesión/máquina para que los cambios tengan efecto.

Configuración del Proyecto

Crear el Archivo `docker-compose.yml`

Cree una nueva carpeta llamada “motionparts”, usando “mkdir motionparts && cd motionparts”

Cree un archivo llamado `docker-compose.yml` usando “touch docker-compose.yml && nano docker-compose.yml” dentro de la carpeta “motionparts” del proyecto con el siguiente contenido:

```
services:  
  
  portainer-ce:  
    image: portainer/portainer-ce:latest  
    container_name: portainer-ce
```

```
restart: unless-stopped

command: -H unix:///var/run/docker.sock

ports:
  - "9000:9000"

volumes:
  - /var/run/docker.sock:/var/run/docker.sock
  - portainer_data:/data

db:
  build: ./db

  image: san2rio/motionparts:db-latest

  container_name: motionparts_mongo

  restart: always

  environment:
    MONGO_INITDB_DATABASE: motionparts

  volumes:
    - mongo_data:/data/db

networks:
  - motionparts_net

healthcheck:
  test: ["CMD", "mongosh", "--eval", "db.adminCommand('ping')"]

  interval: 10s

  timeout: 5s
```

```
retries: 5

start_period: 40s


backend:

build: ./Backend

image: san2rio/motionparts:backend-latest

container_name: motionparts_backend

restart: always

depends_on:

db:

condition: service_healthy

environment:

SPRING_DATA_MONGODB_URI: mongodb://db:27017/motionparts

ports:

- "8080:8080"

volumes:

- ./Backend/uploads:/app/uploads

networks:

- motionparts_net

healthcheck:

test: ["CMD", "curl", "-f", "http://localhost:8080/api/dev/ping"]

interval: 10s

timeout: 5s
```

```
    retries: 5

    start_period: 60s


frontend:

  build: ./Frontend

  image: san2rio/motionparts:frontend-latest

  container_name: motionparts_frontend

  restart: always

  depends_on:

    backend:

      condition: service_healthy

  ports:

    - "4200:80"

  networks:

    - motionparts_net


volumes:

  mongo_data:

  portainer_data:

    driver: local


networks:

  motionparts_net:
```

```

GNU nano 7.2                                            docker-compose.yml *

services:
  portainer-ce:
    image: portainer/portainer-ce:latest
    container_name: portainer-ce
    restart: unless-stopped
    command: -H unix:///var/run/docker.sock
    ports:
      - "9000:9000"
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
      - portainer_data:/data

  db:
    build: ./db
    image: san2rio/motionparts:db-latest
    container_name: motionparts_mongo
    restart: always
    environment:
      MONGO_INITDB_DATABASE: motionparts
    volumes:
      - mongo_data:/data/db
    networks:
      - motionparts_net
    healthcheck:
      test: ["CMD", "mongosh", "--eval", "db.adminCommand('ping')"]
      interval: 10s
      timeout: 5s

^G Help          ^O Write Out   ^W Where Is     ^K Cut           ^T Execute       ^C Location      M-U Undo        M-A Set Mark
^X Exit          ^R Read File   ^V Replace      ^U Paste         ^J Justify       ^L Go To Line    M-E Redo        M-G Copy

```

Se da Ctrl + O para guardar y Ctrl + X para salir del editor

Construir y Levantar los Contenedores

Para usar las imágenes desde docker hub solo se necesitaran dos comandos docker

"docker compose pull" este comando trae las imágenes del repositorio san2rio/motionparts

```

ubuntu@ip-172-31-25-185:~/motionparts$ docker compose pull
[+] Pulling 39/40
  ✓ portainer-ce Pulled
  ✓ db Pulled
  ✓ backend Pulled
  ✓ frontend Pulled

```

"docker compose up -d" esto levantara los 4 contenedores (Mongo, Backend, Frontend y Portainer)

```

ubuntu@ip-172-31-25-185:~/motionparts$ docker compose up -d
[+] Running 8/8
  ✓ Network motionparts_default      Created          0.1s
  ✓ Network motionparts_motionparts_net Created          0.0s
  ✓ Volume motionparts_portainer_data Created          0.0s
  ✓ Volume motionparts_mongo_data    Created          0.0s
  ✓ Container motionparts_mongo     Healthy          8.4s
  ✓ Container portainer-ce          Started          2.4s
  ✓ Container motionparts_backend   Healthy          27.6s
  ✓ Container motionparts_frontend  Started          27.7s
ubuntu@ip-172-31-25-185:~/motionparts$
```

Para verificar el estado de los contenedores se ejecuta el comando “docker ps” o “docker ps -a”

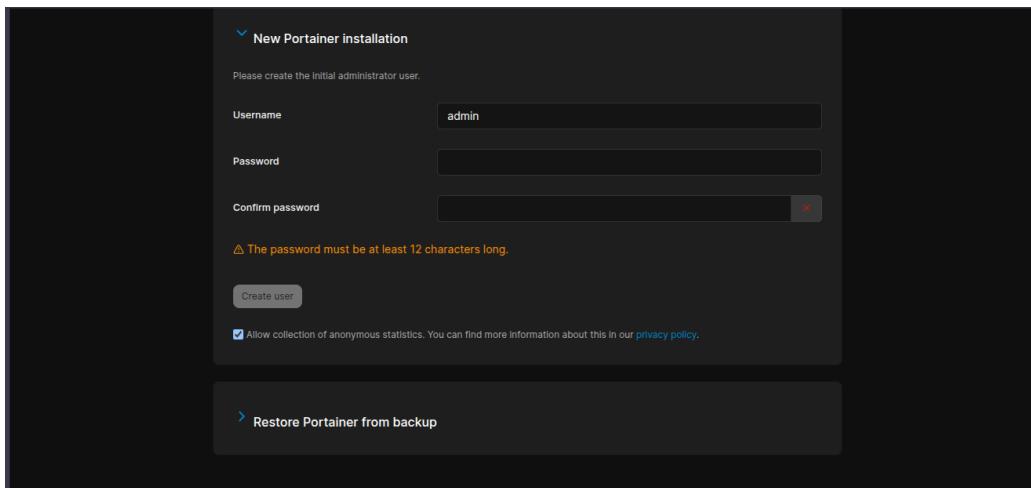
```
ubuntu@ip-172-31-25-185:~$ docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
 NAMES
3f798a0dce0e      san2rio/motionparts:frontend-latest   "/docker-entrypoint..."   3 hours ago       Up 3 hours        0.0.0.0:4200->80/tcp
, [::]:4200->80/tcp
13fe8ac042fe      san2rio/motionparts:backend-latest    "sh -c 'java $JAVA_0..."   3 hours ago       Up 3 hours (healthy)  0.0.0.0:8080->8080/tcp
cp, [::]:8080->8080/tcp
638a0447ff9f      portainer/portainer-ce:latest        "/portainer -H unix:..."   3 hours ago       Up 2 hours        8000/tcp, 9443/tcp,
0.0.0.0:9000->9000/tcp, [::]:9000->9000/tcp
a6d0bbc97459      san2rio/notionparts:db-latest        "docker-entrypoint.s..."   3 hours ago       Up 3 hours (healthy)  27017/tcp
motionparts_mongo

ubuntu@ip-172-31-25-185:~$
```

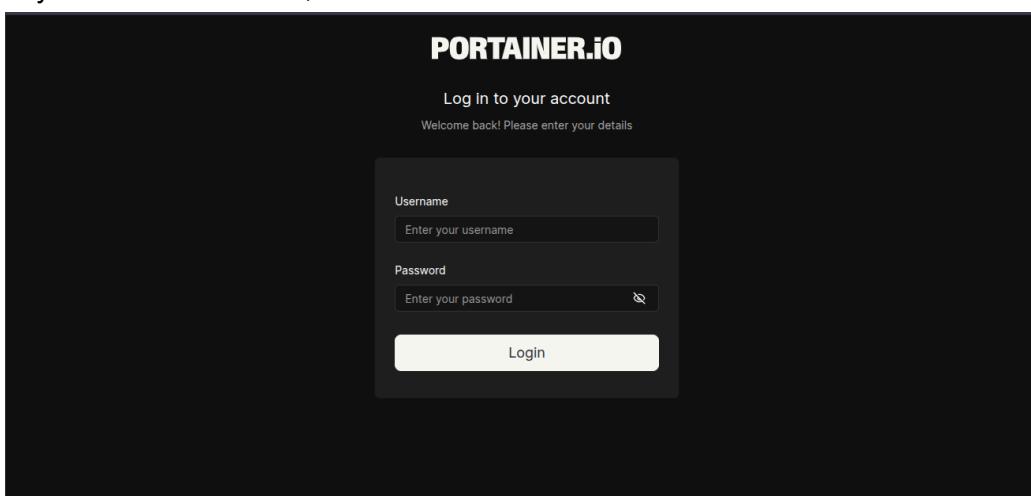
Otra opción es Portainer:

Portainer es una herramienta de gestión visual para contenedores Docker.

1. Abra su navegador y acceda a: [http://\[IP-del-servidor\]:9000](http://[IP-del-servidor]:9000)
2. En el primer acceso, cree una cuenta de administrador



3. Si ya existe una cuenta, solicite las credenciales al administrador del sistema



Desde Portainer puede monitorear el estado de los contenedores, ver logs y realizar tareas administrativas sin necesidad de acceder directamente al servidor

Certificado SSL

Instalación de Caddy

Caddy es un servidor web moderno que gestiona automáticamente los certificados SSL/TLS mediante Let's Encrypt.

Nota: para sacar este certificado es necesario tener un dominio asignado al proyecto, el nombre de estos puede variar según las páginas y herramientas que se usen, hay páginas que tienen un proceso propio de certificación SSL

Paso 1: Instalar Dependencias

```
sudo apt install -y debian-keyring debian-archive-keyring apt-transport-https curl
```

Paso 2: Agregar la Llave GPG de Caddy

```
curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' | sudo gpg --dearmor -o /usr/share/keyrings/caddy-stable-archive-keyring.gpg
```

Paso 3: Agregar el Repositorio de Caddy

```
curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/debian.deb.txt' | sudo tee /etc/apt/sources.list.d/caddy-stable.list
```

Paso 4: Configurar Permisos

```
chmod o+r /usr/share/keyrings/caddy-stable-archive-keyring.gpg
```

```
chmod o+r /etc/apt/sources.list.d/caddy-stable.list
```

Paso 5: Instalar Caddy

```
sudo apt update
```

```
sudo apt install caddy
```

Paso 6: Configurar el Caddyfile

Navegue al directorio de configuración de Caddy:

```
cd /etc/caddy
```

Verifique que existe el archivo **Caddyfile**: con el comando ls

```
ubuntu@ip-172-31-25-185:/etc/caddy$ ls
Caddyfile
ubuntu@ip-172-31-25-185:/etc/caddy$
```

Edite el archivo **Caddyfile**: sudo nano Caddyfile

Reemplace el contenido con la siguiente configuración (sustituya **dominio.org** por su dominio real):

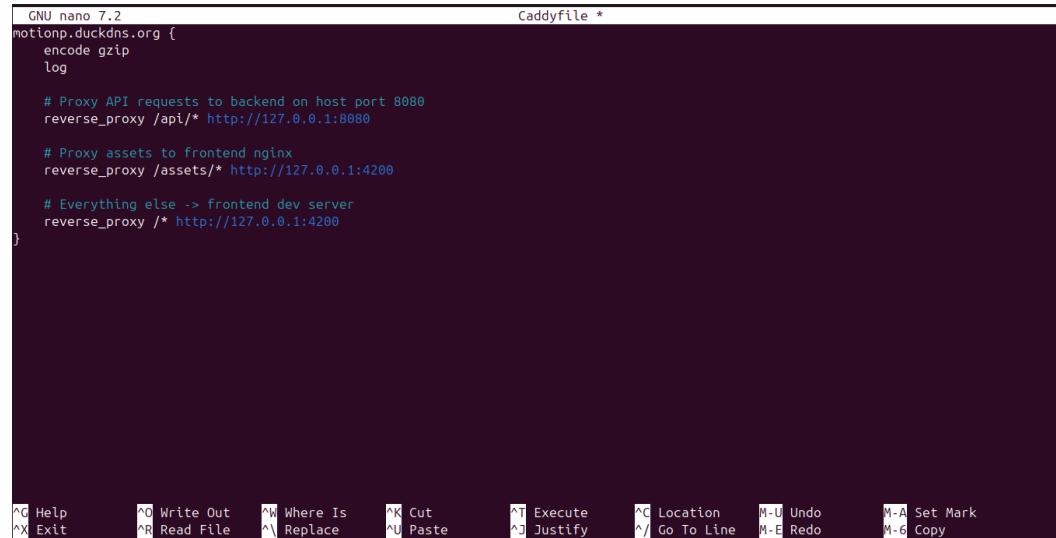
```
dominio.org {
    encode gzip
    log

    # Proxy API requests to backend on host port 8080
    reverse_proxy /api/* http://127.0.0.1:8080

    # Proxy assets to frontend nginx
    reverse_proxy /assets/* http://127.0.0.1:4200

    # Everything else -> frontend dev server
    reverse_proxy /* http://127.0.0.1:4200
}
```

Guarde el archivo (Ctrl + O, Enter, Ctrl + X en nano).



The screenshot shows a terminal window with the nano text editor open. The file is named 'Caddyfile'. The content of the file is as follows:

```
motionp.duckdns.org {
    encode gzip
    log

    # Proxy API requests to backend on host port 8080
    reverse_proxy /api/* http://127.0.0.1:8080

    # Proxy assets to frontend nginx
    reverse_proxy /assets/* http://127.0.0.1:4200

    # Everything else -> frontend dev server
    reverse_proxy /* http://127.0.0.1:4200
}
```

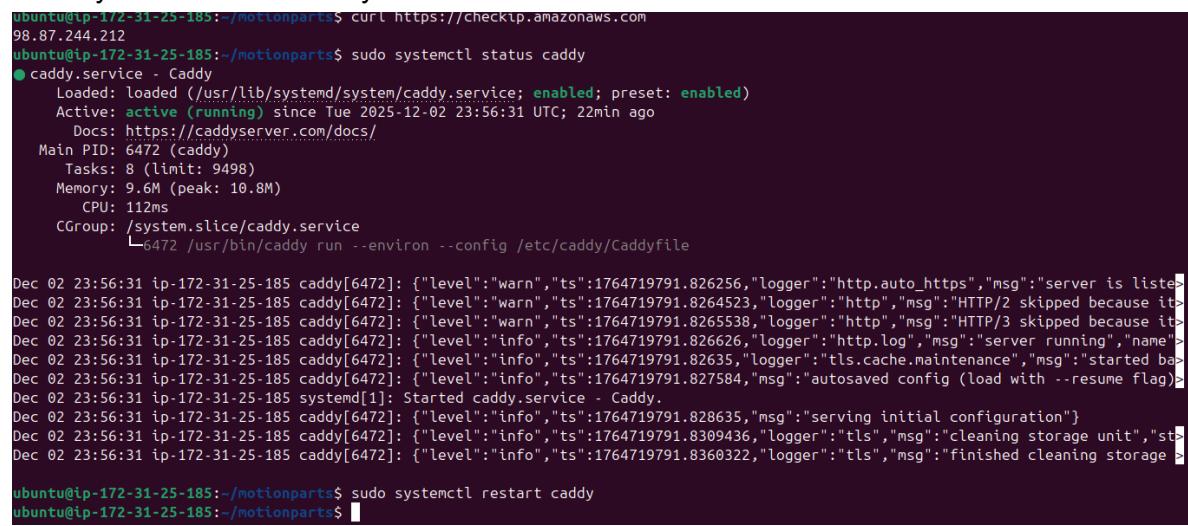
At the bottom of the terminal window, there is a menu bar with various keyboard shortcuts for file operations like Help, Exit, Write Out, Read File, Where Is, Replace, Cut, Paste, Execute, Justify, Location, Go To Line, Undo, Redo, Set Mark, and Copy.

Paso 7: Reiniciar Caddy

```
sudo systemctl restart caddy
```

Paso 8: Verificar el Estado de Caddy

```
sudo systemctl status caddy
```



The screenshot shows a terminal window displaying the output of the 'systemctl status caddy' command. The output indicates that the 'caddy.service' is active (running) since December 2, 2025, at 23:56:31 UTC, with a duration of 22 minutes. It shows the main PID is 6472, and it has 8 tasks with a peak memory usage of 10.8M. The service is running under the cgroup /system.slice/caddy.service.

```
ubuntu@ip-172-31-25-185:~/motionparts$ curl https://checkip.amazonaws.com
98.87.244.212
ubuntu@ip-172-31-25-185:~/motionparts$ sudo systemctl status caddy
● caddy.service - Caddy
   Loaded: loaded (/usr/lib/systemd/system/caddy.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-12-02 23:56:31 UTC; 22min ago
     Docs: https://caddyserver.com/docs/
     Main PID: 6472 (caddy)
        Tasks: 8 (limit: 9498)
       Memory: 9.6M (peak: 10.8M)
          CPU: 112ms
         CGroup: /system.slice/caddy.service
             └─ 6472 /usr/bin/caddy run --environ --config /etc/caddy/Caddyfile

Dec 02 23:56:31 ip-172-31-25-185 caddy[6472]: {"level":"warn","ts":1764719791.826256,"logger":"http.auto_https","msg":"server is liste}
Dec 02 23:56:31 ip-172-31-25-185 caddy[6472]: {"level":"warn","ts":1764719791.8264523,"logger":"http","msg":HTTP/2 skipped because it>
Dec 02 23:56:31 ip-172-31-25-185 caddy[6472]: {"level":"warn","ts":1764719791.8265538,"logger":"http","msg":HTTP/3 skipped because it>
Dec 02 23:56:31 ip-172-31-25-185 caddy[6472]: {"level":"info","ts":1764719791.826626,"logger":"http.log","msg":server running,"name">
Dec 02 23:56:31 ip-172-31-25-185 caddy[6472]: {"level":"info","ts":1764719791.82635,"logger":"tls.cache.maintenance","msg":started ba>
Dec 02 23:56:31 ip-172-31-25-185 caddy[6472]: {"level":"info","ts":1764719791.827584,"msg":autosaved config (load with --resume flag)>
Dec 02 23:56:31 ip-172-31-25-185 systemd[1]: Started caddy.service - Caddy.
Dec 02 23:56:31 ip-172-31-25-185 caddy[6472]: {"level":"info","ts":1764719791.828635,"msg":serving initial configuration}
Dec 02 23:56:31 ip-172-31-25-185 caddy[6472]: {"level":"info","ts":1764719791.8309436,"logger":"tls","msg":cleaning storage unit,"st>
Dec 02 23:56:31 ip-172-31-25-185 caddy[6472]: {"level":"info","ts":1764719791.8360322,"logger":"tls","msg":finished cleaning storage >

ubuntu@ip-172-31-25-185:~/motionparts$ sudo systemctl restart caddy
ubuntu@ip-172-31-25-185:~/motionparts$
```

Importante: Caddy tardará de 3 a 5 minutos en generar y configurar automáticamente el certificado SSL de Let's Encrypt. Durante este tiempo, el sitio puede ser inaccesible temporalmente.

Acceso y Verificación

Una vez completada la instalación, el sistema estará disponible en:

URLs de Acceso

- **Aplicación Principal:** <https://dominio.org>
- **Portainer:** [http://\[IP-del-servidor\]:9000](http://[IP-del-servidor]:9000)

Verificación del Certificado SSL

Después de aproximadamente 5 minutos, verifique que el sitio esté accesible mediante HTTPS con el candado verde en el navegador, indicando que el certificado SSL es válido.

El cambio será de <http://dominio.org> → <https://dominio.org>

Comandos Útiles

Ver logs de los contenedores:

```
docker compose logs -f [nombre_servicio]
```

Reiniciar un servicio específico:

```
docker compose restart [nombre_servicio]
```

Detener todos los servicios:

```
docker compose down
```

Actualizar y reiniciar servicios:

```
docker compose up -d --build
```

Ver logs de Caddy:

```
sudo journalctl -u caddy -f
```