

Ejecutar una instancia

- Para ejecutar una instancia nueva, accederá a la dirección <https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#LaunchInstances:> , dentro de la página, se encontrará un configurador para la nueva instancia.
- Configurara el nombre de la instancia.

EC2 > Instances > Launch an instance

Launch an instance [Info](#)

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

Name and tags [Info](#)

Name


[Add additional tags](#)

- Seleccionará el Sistema Operativo bajo el cual se ejecutará el servidor en AWS y sus características (En este caso al ser una aplicación básica, se elegirá el sistema operativo Ubuntu Server 24.04 LTS en su configuración más básica):


▼ 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


Quick Start




Amazon Linux




macOS




Ubuntu




Windows



Red Hat



SUSE Linux



[Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

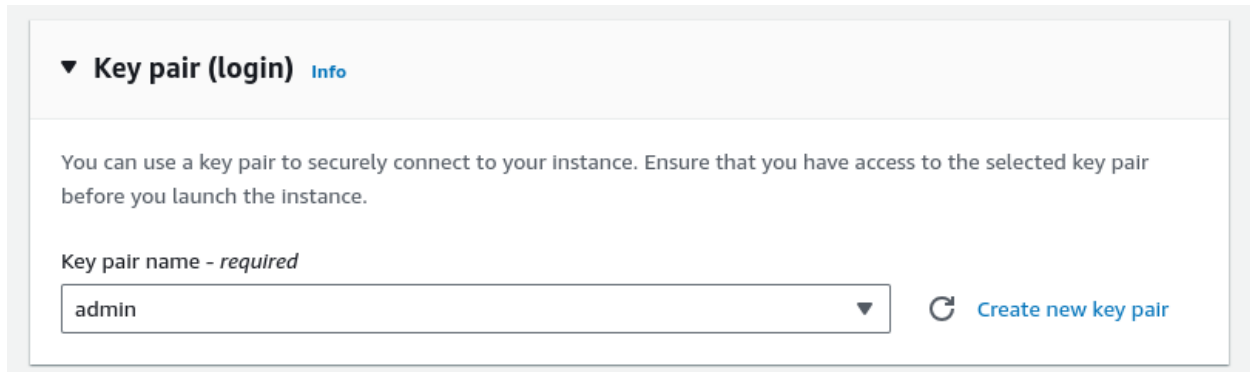
Ubuntu Server 24.04 LTS (HVM), SSD Volume Type

Free tier eligible ▼

ami-09040d770ffe2224f (64-bit (x86)) / ami-0acb327475c6fd498 (64-bit (Arm))

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

- El “Key pair” es un conjunto de llaves encriptadas que requiere el servidor para la comunicación a través del protocolo SSH (Secure SHell), por lo que se tendrá que elegir un conjunto de llaves ya generadas, en caso de no tener ninguna, se creará un nuevo conjunto dando clic en la opción “Create new key pair”.



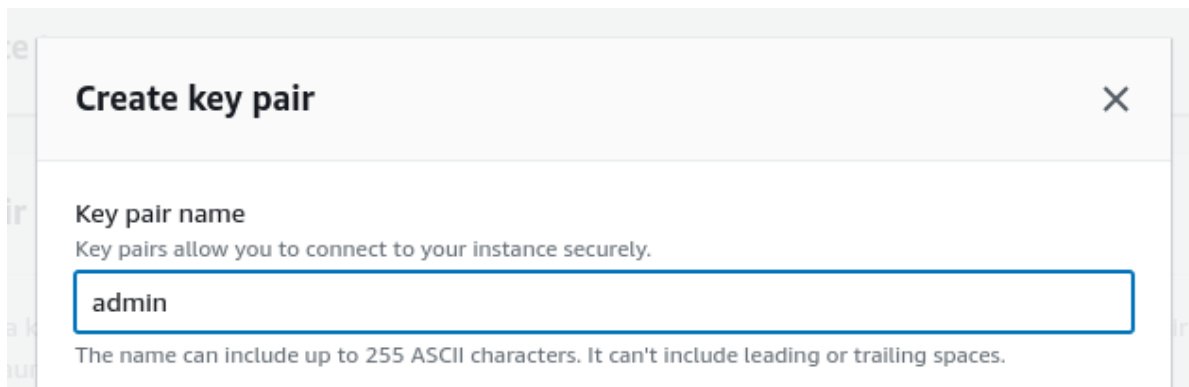
▼ **Key pair (login)** [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

[Create new key pair](#)

- Esta opción desplegará un configurador, donde se tendrá que poner un nombre al conjunto de llaves nueva, así como el tipo de encriptación (RSA) y el formato del archivo (Que en este caso se utilizara .pem para la comunicación por Open SSH).



Create key pair ×

Key pair name
Key pairs allow you to connect to your instance securely.

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

- una vez configuradas, se dará clic en “Create key pair”; automáticamente se descargará un archivo .pem con el nombre configurado (es importante no perder o eliminar este archivo).

The screenshot shows the 'Create key pair' dialog box. Under 'Key pair type', the 'RSA' option is selected, with a description 'RSA encrypted private and public key pair'. The 'ED25519' option is also visible, described as 'ED25519 encrypted private and public key pair'. Under 'Private key file format', the '.pem' option is selected, noted as 'For use with OpenSSH'. The '.ppk' option is noted as 'For use with PuTTY'. A yellow warning box states: 'When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)'. At the bottom right are 'Cancel' and 'Create key pair' buttons.

Configuración de la red

- Para la configuración de la red, se tendrá que crear un nuevo grupo de seguridad (por defecto).

The screenshot shows the 'Network settings' panel. It includes an 'Edit' button in the top right. The 'Network' section shows the VPC ID 'vpc-05f900d82c95eda2a'. The 'Subnet' section is set to 'No preference (Default subnet in any availability zone)'. The 'Auto-assign public IP' option is 'Enable', with a note that 'Additional charges apply when outside of free tier allowance'. The 'Firewall (security groups)' section has an explanatory text: '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.' At the bottom, there are two buttons: 'Create security group' (selected) and 'Select existing security group'.

- Para permitir la comunicación hacia el servidor por el puerto 22 (Para el protocolo SSH) En las opciones de seguridad se seleccionará la opción 'Allow SSH traffic from', a continuación, la opción (My IP) para que el servidor únicamente reconozca

y permita las peticiones provenientes de la dirección IP del administrador de la instancia.

We'll create a new security group called 'launch-wizard-1' with the following rules:

- ☒ Allow SSH traffic from
Helps you connect to your instance

My IP
189.188.74.233/32

- Una vez configuradas estas opciones, se dará clic en el botón "Launch instance", terminado el proceso de configuración; aparecerá una notificación 'Success'.

[EC2](#) > [Instances](#) > Launch an instance

✓ Success


Successfully initiated launch of instance [\(i-0e5bcabc6d4011146\)](#)

► Launch log

Configuración de la instancia


- Para la configuración de la instancia, accederá al panel de control en la siguiente dirección [https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#Instances:v=3;\\$case=tags:true%5C,client:false;\\$regex=tags:false%5C,client:false](https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#Instances:v=3;$case=tags:true%5C,client:false;$regex=tags:false%5C,client:false), dentro de esta, se desplegará una lista, donde mostrará las instancias disponibles.





Instances (1) [Info](#)

 [Connect](#) [Instance state ▼](#) [Actions ▼](#)

[Launch instances ▼](#)

[All states ▼](#)

< 1 > 

| <input type="checkbox"/> | Name  ▼ | Instance ID | Instance state ▼ | Instance type |
|--------------------------|--|-------------------------------------|---|---------------|
| <input type="checkbox"/> | sps_practica | i-0c7ab724bf7a76cb6 |  Running   | t2.micro |

- Seleccione su instancia para ver los detalles.

i-0c7ab724bf7a76cb6 (sps_practica)



< **Details** | Status and alarms | Monitoring | Security | Networking

▼ Instance summary [Info](#)

Instance ID

i-0c7ab724bf7a76cb6 (sps_practica)

Private IPv4 addresses

172.31.44.208

Instance state

Running

Hostname type

IP name: ip-172-31-44-208.us-east-2.compute.internal

Public IPv4 address

3.131.158.117 | [open address](#)

IPv6 address

–

Public IPv4 DNS

ec2-3-131-158-117.us-east-2.compute.amazonaws.com
| [open address](#)

Private IP DNS name (IPv4 only)

ip-172-31-44-208.us-east-2.compute.inte

Answer private resource DNS name
IPv4 (A)

- La opción Public IPv4 DNS, es la dirección a la cual se accederá hacia el recurso montado dentro del servidor, para configurar el acceso se dará clic en la pestaña “Security” y después en la opción “Security groups”.

i-0c7ab724bf7a76cb6 (sps_practica)



< Details Status and alarms Monitoring **Security** Networking s >

▼ Security details

IAM Role

–

Launch time

Fri Jun 07 2024 12:19:06 GMT-0500 (Central Daylight Time)

Owner ID

891376970339

Security groups

sg-0de982d407e0179d1 (launch-wizard-1)

▼ Inbound rules

- Dentro del configurador, se verán todas las reglas de entrada y salida del servidor, para configurar una nueva se dará clic en la opción “Edit inbound rules”.

| Inbound rules (3) | | | | Manage tags | Edit inbound rules |
|-------------------------------------|---|----------|-------|-------------|--------------------|
| <input type="text" value="Search"/> | | | | | |
| | | | < 1 > | | |
| Type | ▼ | Protocol | ▼ | Port range | |
| Custom TCP | | TCP | | 3307 | |
| Custom TCP | | TCP | | 3000 | |
| SSH | | TCP | | 22 | |

- Dentro de este configurador, se agregarán dos nuevas reglas, las dos de tipo “Custom TCP”.
- La primera que ingrese por el puerto 3000 (Para acceder al frontend) desde cualquier puerto, cualquier dirección (0.0.0.0/0).

Inbound rule 1

Delete

Security group rule ID

sgr-0d9af5a7817571a08

Type [Info](#)

Custom TCP ▼

Protocol [Info](#)

TCP

Port range [Info](#)

3307

Source type [Info](#)

Custom ▼

Source [Info](#)

Q

0.0.0.0/0 X

Description - optional [Info](#)

- La segunda que ingrese por el puerto 3307 (Para acceder al backend) desde cualquier puerto, cualquier dirección (0.0.0.0/0); Una vez configuradas las nuevas reglas, se dará clic en el botón 'Save rules'.

Inbound rule 2

Delete

Security group rule ID

sgr-0851473a1a3fdee17

Type [Info](#)

Custom TCP ▼

Protocol [Info](#)

TCP

Port range [Info](#)

3000

Source type [Info](#)

Custom ▼

Source [Info](#)

Q

0.0.0.0/0 X

Description - optional [Info](#)

Configuración del servidor

Acceder al servidor

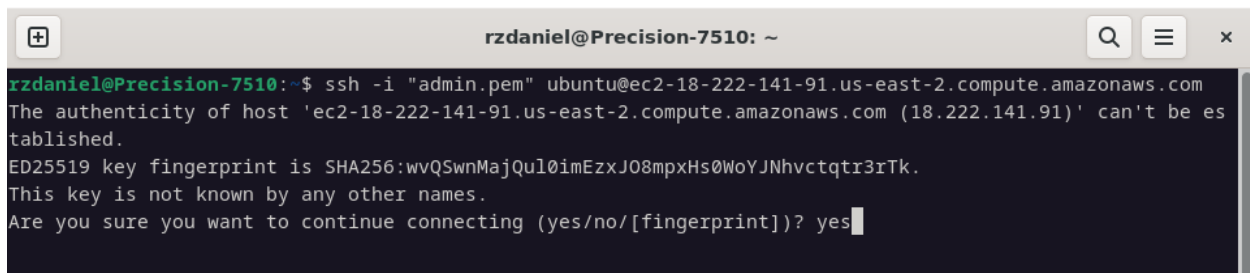
Para acceder al servidor se requiere de la llave .pem generada en el configurador de la nueva instancia, es importante no perder esta llave para continuar garantizando el acceso hacia el servidor. Para el caso de este ejemplo, se realizó la configuración del servidor con OpenSSH desde un sistema operativo Linux (Fedora).

- Primero se tiene que cambiar los permisos del archivo a solo lectura, para esto se ejecutara el comando “sudo chmod 400 admin.pem” donde admin es el nombre del archivo .pem generado por el configurador de la instancia.

```
rzdaniel@Precision-7510:~$ sudo chmod 400 admin.pem
[sudo] password for rzdaniel:
```

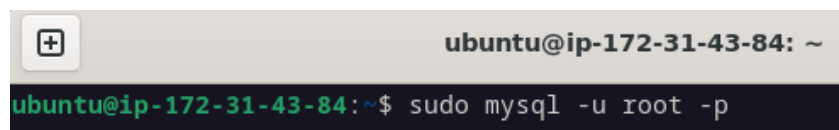
- Una vez realizado este proceso, se accederá hacia el servidor por medio del protocolo SSH ejecutando el siguiente comando desde la terminal:

ssh -i "admin.pem" ubuntu@ec2-18-222-141-91.us-east-2.compute.amazonaws.com



```
rzdaniel@Precision-7510: ~
rzdaniel@Precision-7510:~$ ssh -i "admin.pem" ubuntu@ec2-18-222-141-91.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-18-222-141-91.us-east-2.compute.amazonaws.com (18.222.141.91)' can't be es
tablished.
ED25519 key fingerprint is SHA256:vvQSwNjMajQul0imEzxJ08mpxHs0WoYJNhvctqtr3rTk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

- Donde -i corresponde a “identity_file” que es la llave pem antes mencionada, ubuntu es el usuario con el que nos conectaremos, y ec2-... corresponde al DNS o la ip de nuestro servidor.
- Una vez establecida la comunicación cambiara nuestro usuario y nombre del equipo a las del servidor:



```
ubuntu@ip-172-31-43-84: ~
ubuntu@ip-172-31-43-84:~$ sudo mysql -u root -p
```

- Dentro del servidor, el primer paso es actualizar el servidor, para eso, se ejecutará el siguiente comando:

```
sudo apt update && sudo apt upgrade
```

Instalación y configuración de MySQL server

- Para la instalación de MySQL server, se requiere ejecutar el siguiente comando:

```
sudo apt install mysql-server
```


- Una vez terminado el proceso, se deberá de comprobar que el servicio esté disponible, para eso se ejecutará el comando:

```
sudo systemctl status mysql.service
```

- Si el servicio se encuentra ejecutado correctamente, se mostrará la siguiente información en la terminal:

```
ubuntu@ip-172-31-44-208:~$ sudo systemctl status mysql.service
• mysql.service - MySQL Community Server
  Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: >
  Active: active (running) since Fri 2024-06-07 17:19:44 UTC; 3h 53min ago
  Main PID: 484 (mysqld)
  Status: "Server is operational"
  Tasks: 39 (limit: 1121)
  Memory: 366.6M
  CPU: 53.369s
  CGroup: /system.slice/mysql.service
          └─484 /usr/sbin/mysqld

Jun 07 17:19:39 ip-172-31-44-208 systemd[1]: Starting MySQL Community Server...
Jun 07 17:19:44 ip-172-31-44-208 systemd[1]: Started MySQL Community Server.
lines 1-13/13 (END)
```

- Una vez validada la ejecución, accederemos al servidor mysql con el comando:

```
sudo mysql -u root -p
```

- Es importante ejecutar el comando como super usuario ya que para acceder al usuario root únicamente se puede hacer desde el super usuario, el argumento `-u` corresponde al usuario de acceso y el argumento `-p` corresponde la contraseña de este, para acceder al usuario root no se requiere contraseña.

```
ubuntu@ip-172-31-43-84: ~  
ubuntu@ip-172-31-43-84:~$ sudo mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 11  
Server version: 8.0.36-2ubuntu3 (Ubuntu)  
  
Copyright (c) 2000, 2024, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql>
```

- Dentro del servidor, se creará primero la base de datos;

```
mysql> CREATE DATABASE sps_practica;  
Query OK, 1 row affected (0.01 sec)
```

- Se seleccionará la base de datos

```
mysql> USE sps_practica;  
Database changed
```

- Dentro de esta base de datos, se crearán las tablas correspondientes al script del proyecto “schema.sql”, una vez creadas estas tablas, se podrá ejecutar el comando “show tables” para mostrar las tablas que han sido creadas.

```
mysql> show tables;  
+-----+  
| Tables_in_sps_practica |  
+-----+  
| habilidad               |  
| participante            |  
| participante_habilidad  |  
+-----+  
3 rows in set (0.00 sec)  
  
mysql> █
```

- Se tendrá que crear un nuevo usuario que permita el acceso del backend hacia la base de datos, esto es importante pues el uso del usuario root podría comprometer la integridad del servidor, ya que el usuario root tiene acceso a todos los permisos dentro de las bases de datos alojadas dentro del servidor, para prevenir esto, primero se tendrá que crear un nuevo usuario con el siguiente comando:

```
CREATE USER 'backend'@'localhost' IDENTIFIED BY 'd0Em#-  
Sf5UDS8cCSsLV0g!6kAHh*3O#c';
```

- Donde “backend” corresponde al nombre de usuario “localhost” corresponde a la dirección desde la cual el usuario podrá acceder e “IDENTIFIED BY” es la contraseña de este usuario.
- Una vez creado el usuario, habrá que darle permisos para poder seleccionar e insertar datos dentro de la base de datos “sps_practica” para esto, se ejecutará el siguiente comando:

```
GRANT SELECT, INSERT ON sps_practica.* TO 'backend'@'localhost';
```

- Donde “GRANT SELECT, INSERT” corresponden a los permisos del usuario para poder ejecutar las acciones de lectura y escritura, “sps_practica” es el nombre de la base de datos donde se podrán ejecutar estas acciones, “*” corresponde a las tablas en las cuales se podrá ejecutar estas acciones, en este caso, se colocara el * para indicar que todas las tablas dentro de la base de datos “sps_practicas” podrán ser manipuladas por este usuario y “backend@localhost” corresponde al usuario y la dirección a los cuales se les darán los permisos.
- Una vez configurados estos permisos, se tienen que recargar los mismos, para esto se ejecuta el siguiente comando.

```
FLUSH PRIVILEGES;
```

- Una vez ejecutado este comando, se concluye la configuración de la base de datos.

Configuración del proyecto en el servidor

Para configurar el proyecto dentro del servidor, se requiere instalar nodejs y dentro de este la dependencia llamada “serve”, serve es un servidor para desplegar este tipo de proyectos.

- Para instalar nodejs dentro del servidor se ejecutará el siguiente comando

```
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -  
  
sudo apt-get install -y nodejs
```

- Una vez ejecutado se instalará serve con el comando

```
npm install -g serve
```

Una vez instalador nodejs y serve para desplegar el proyecto, se creará un archivo .env, con las variables de entorno de producción, esto para la ejecución del backend.

- Para crear este archivo, se ejecuta el siguiente comando

```
sudo nano /etc/.env
```

- Una vez creado, dentro del editor nano, se creará el siguiente listado de variables.

```
DB_HOST=localhost  
DB_PORT=3306  
DB_USER=backend  
DB_PASS= d0Em#-Sf5UDS8cCSsLV0g!6kAHh*3O#c  
DB_DABA=sps_practica  
BACK_PORT=3307
```

```
ubuntu@ip-172-31-43-84: ~
GNU nano 7.2 /etc/.env *
DB_HOST=localhost
DB_PORT=3306
DB_USER=backend
DB_PASS=d0Em#-Sf5UDS8cCSsLV0g!6kAHh*30#c
DB_DABA=sps_practica
BACK_PORT=3307
```

Ejecución del proyecto

- Para la ejecución del proyecto, primero tendremos que clonarlo en el servidor, para esto se utilizará el comando git clone.

```
ubuntu@ip-172-31-43-84: ~$ git clone https://github.com/loop-danielr/sps-practica.git
Cloning into 'sps-practica'...
Username for 'https://github.com': loop-danielr
Password for 'https://loop-danielr@github.com':
remote: Enumerating objects: 143, done.
remote: Counting objects: 100% (143/143), done.
remote: Compressing objects: 100% (74/74), done.
remote: Total 143 (delta 59), reused 133 (delta 52), pack-reused 0
Receiving objects: 100% (143/143), 324.61 KiB | 6.12 MiB/s, done.
Resolving deltas: 100% (59/59), done.
ubuntu@ip-172-31-43-84: ~$
```

- Una vez clonado el repositorio, entraremos a la carpeta contenedora y cambiaremos la branch del proyecto, de "main" a "development".

```
ubuntu@ip-172-31-43-84: ~/sps-practica
ubuntu@ip-172-31-43-84:~$ cd sps-practica/
ubuntu@ip-172-31-43-84:~/sps-practica$ git branch
* main
ubuntu@ip-172-31-43-84:~/sps-practica$ git checkout development
branch 'development' set up to track 'origin/development'.
Switched to a new branch 'development'
ubuntu@ip-172-31-43-84:~/sps-practica$ git branch
* development
  main
ubuntu@ip-172-31-43-84:~/sps-practica$
```

- Accederemos a la carpeta del backend y ejecutaremos el comando `npm install`, para instalar las dependencias requeridas por el proyecto Express, este paso no es necesario en el frontend ya que, este cuenta con su compilación para desplegar el proyecto en producción.

Una vez instaladas las dependencias se tendrán que crear los servicios para poder realizar la ejecución de los proyectos.

- Para la creación de los servicios, primero se creará la configuración del servicio “sps-backend” para ello, se ejecutará el siguiente comando:

```
sudo nano /etc/systemd/system/sps-backend.service
```

- Dentro de este archivo, se pegará la siguiente configuración:

```
[Unit]
Description=Express server to sps practica
After=network.target multi-user.target

[Service]
User=ubuntu
WorkingDirectory=/home/ubuntu/sps-practica/backend
ExecStart=/usr/bin/node index.js
Restart=always
Environment=NODE_ENV=production
EnvironmentFile=/etc/.env
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=sps-backend

[Install]
WantedBy=multi-user.target
```

- Donde `WorkDirectory` es la ruta donde se encuentra el backend, `ExecStar` es el comando para ejecutar nuestro backend y `EnvironmentFile` es el archivo `.env` contenedor de las variables de entorno antes creadas

```
ubuntu@ip-172-31-43-84: ~/sps-practica/backend
GNU nano 7.2 /etc/systemd/system/sps-backend.service *
[Unit]
Description=Express backend to sps practica
After=network.target multi-user.target

[Service]
User=ubuntu
WorkingDirectory=/home/ubuntu/sps-practica/backend
ExecStart=/usr/bin/node index.js
Restart=always
Environment=NODE_ENV=production
EnvironmentFile=/etc/.env
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=sps-practica-backend

[Install]
WantedBy=multi-user.target
```

- Para la creación del servicio correspondiente a "sps-frontend" se ejecutará el siguiente comando

```
sudo nano /etc/systemd/system/sps-frontend.service
```

- Dentro de este se pegará la siguiente configuración:

```
[Unit]
Description=React to sps practica
After=network.target multi-user.target

[Service]
User=ubuntu
WorkingDirectory=/home/ubuntu/sps-practica/frontend
ExecStart=/usr/bin/serve -n -s build
Restart=always
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=sps-frontend
```

[Install]

WantedBy=multi-user.target

- Una vez creados los servicios, se requiere reiniciar los procesos del sistema (System daemons), para ello se ejecutará el siguiente comando:

```
sudo systemctl daemon-reload
```

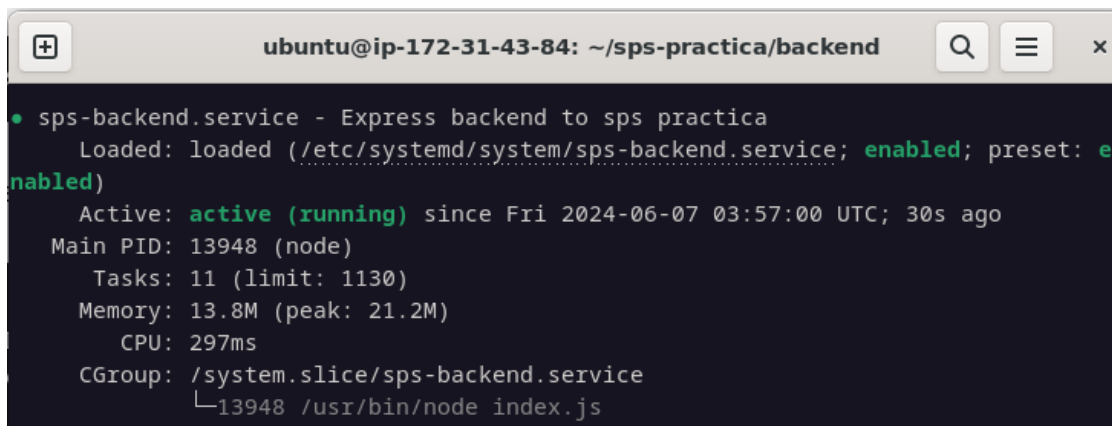
- Para habilitar los nuevos servicios, se ejecuta el siguiente comando:

```
sudo systemctl enable sps-backend.service && sudo systemctl enable sps-frontend.service
```

- Ejecutados estos comandos, los servicios se deberán encontrar disponibles, para comprobarlo, se ejecutará el comando:

```
sudo systemctl status sps-backend
```

```
sudo systemctl status sps-frontend
```



A terminal window titled 'ubuntu@ip-172-31-43-84: ~/sps-practica/backend'. The terminal output shows the status of the 'sps-backend.service'. It is loaded and enabled, and is currently active and running. The output includes details such as the main PID (13948), tasks (11), memory usage (13.8M), CPU usage (297ms), and the CGroup path.

```
• sps-backend.service - Express backend to sps practica
  Loaded: loaded (/etc/systemd/system/sps-backend.service; enabled; preset: e
nabled)
  Active: active (running) since Fri 2024-06-07 03:57:00 UTC; 30s ago
    Main PID: 13948 (node)
      Tasks: 11 (limit: 1130)
     Memory: 13.8M (peak: 21.2M)
        CPU: 297ms
      CGroup: /system.slice/sps-backend.service
              └─13948 /usr/bin/node index.js
```