

Hugo Sánchez Barroso

Despliegue de Guzpasen en AWS

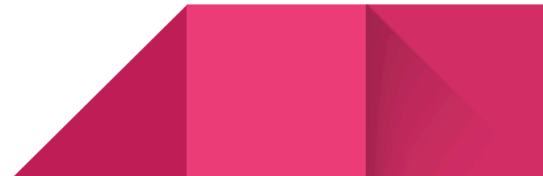
5 de junio del 2025



IES Torres de los Guzmanes - 2 DAM

ÍNDICE

1. Introducción.....	3
2. Paso 1. Acceso a AWS Academy.....	4
3. Paso 2. Acceso al Learner Lab.....	5
4. Paso 3. Generar Claves EC2.....	6
5. Paso 4: Desplegar la Plantilla de CloudFormation.....	7
6. Paso 5. Conectar MySql Workbench a la base de datos.....	9
7. Paso 6. Mover empaquetados a la carpeta de Samba.....	11
8. Paso 7. Instalar PuTTY y Conectarse por SSH.....	14
9. Paso 8. Desplegar los Microservicios con Spring Boot.....	16



1. Introducción.

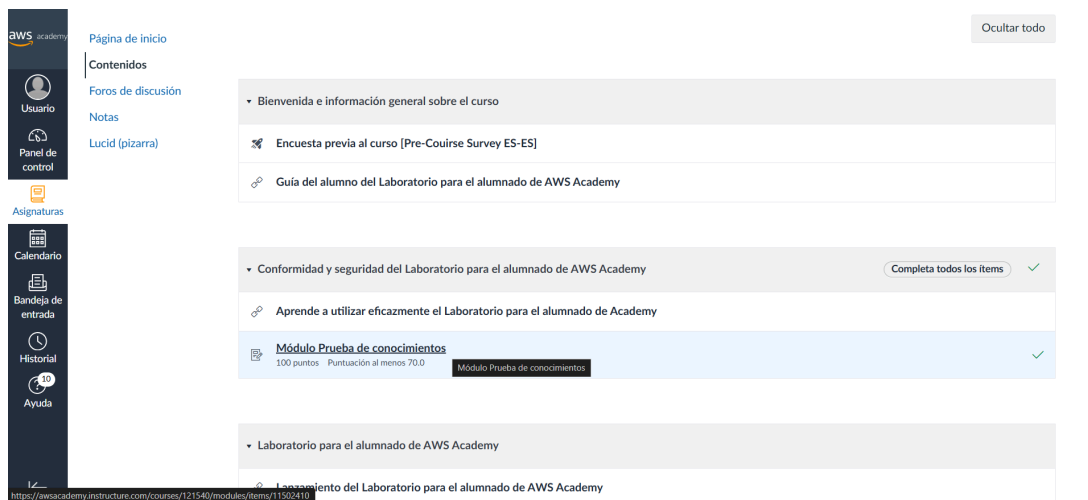
En este documento se detalla el proceso completo de despliegue de mi aplicación en Amazon Web Services (AWS). A lo largo de las siguientes secciones, se explicará paso a paso cada una de las acciones realizadas. El objetivo es proporcionar una guía clara y estructurada que permita comprender cómo se ha llevado a cabo el despliegue y facilitar su futura replicación o mejora.



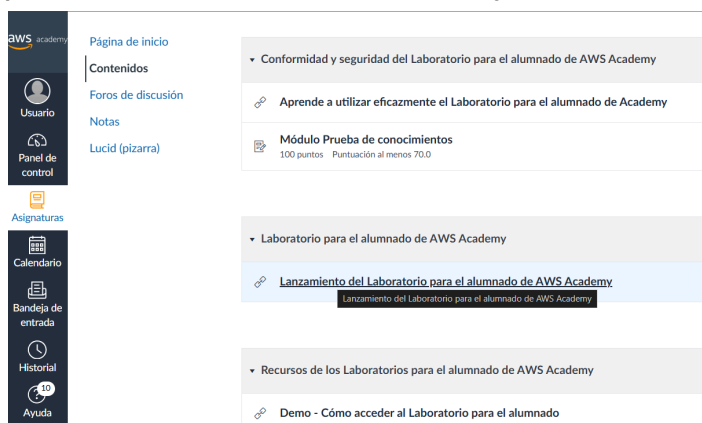
2. Paso 1. Acceso a AWS Academy.

En primer lugar nos crearemos una cuenta. Una vez nos registremos, accedemos al apartado de “Asignaturas”.

Para poder tener acceso a la clase que contiene el laboratorio de AWS, realizaremos una breve prueba de conocimientos. Esta consta únicamente de 7 preguntas de las cuales debemos acertar un 70% para poder seguir con el manual.

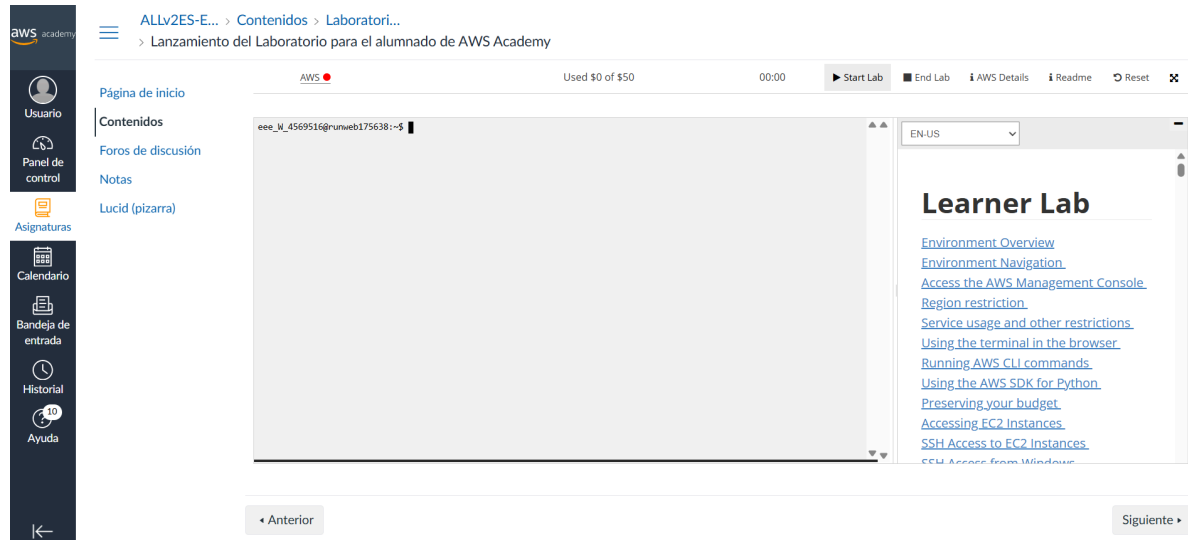


Una vez completada la prueba, accedemos el apartado de “Lanzamiento del Laboratorio para el alumnado de AWS Academy”:



3. Paso 2. Acceso al Learner Lab.

Al entrar en este apartado, veremos la siguiente pantalla:

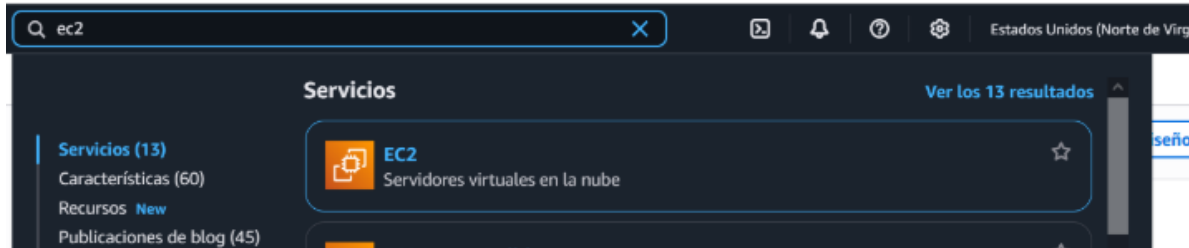


Tendremos que clicar en el botón “Start Lab” para poder iniciar nuestro Laboratorio. Tendremos que esperar a que el círculo **Rojo** de al lado de AWS cambie a **verde**.

4. Paso 3. Generar Claves EC2.

Una vez se inicie nuestro laboratorio, debemos clicar en el icono de **AWS con el círculo verde**.

Buscamos “EC2” en el buscador que nos aparece en la parte superior de la pantalla:



A continuación accedemos al apartado de “Pares de clave”. En este menú, crearemos un par de claves con la siguiente configuración:

Crear par de claves [Información](#)

Par de claves
Un par de claves, compuesto por una clave privada y una clave pública, es un conjunto de credenciales de seguridad que se utilizan para demostrar su identidad cuando se conecta a una instancia.

Nombre

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

Tipo de par de claves [Información](#)
☒ RSA ☐ ED25519

Formato de archivo de clave privada
☐ .pem
Para usar con OpenSSH
☒ .ppk
Para usar con PuTTY

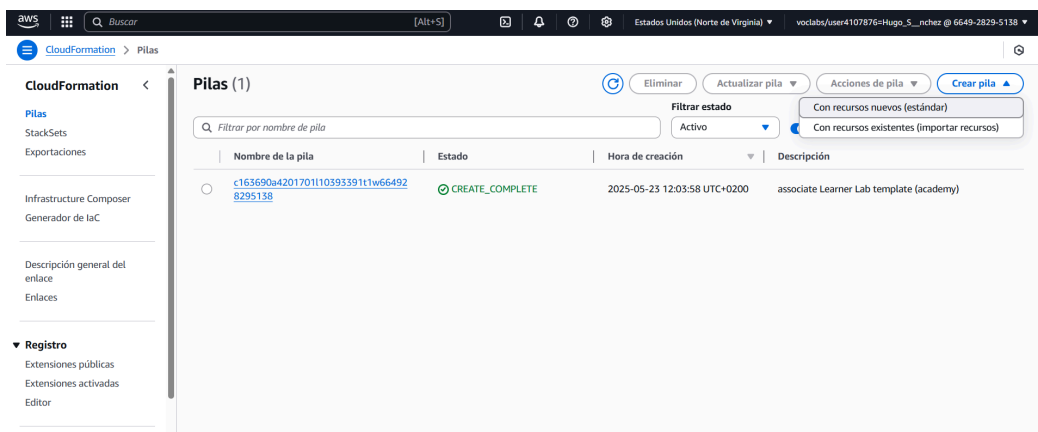
Etiquetas: opcional
No hay etiquetas asociadas a este recurso.
[Agregar nueva etiqueta](#)
Puede agregar hasta 50 etiquetas más.

5. Paso 4: Desplegar la Plantilla de CloudFormation.

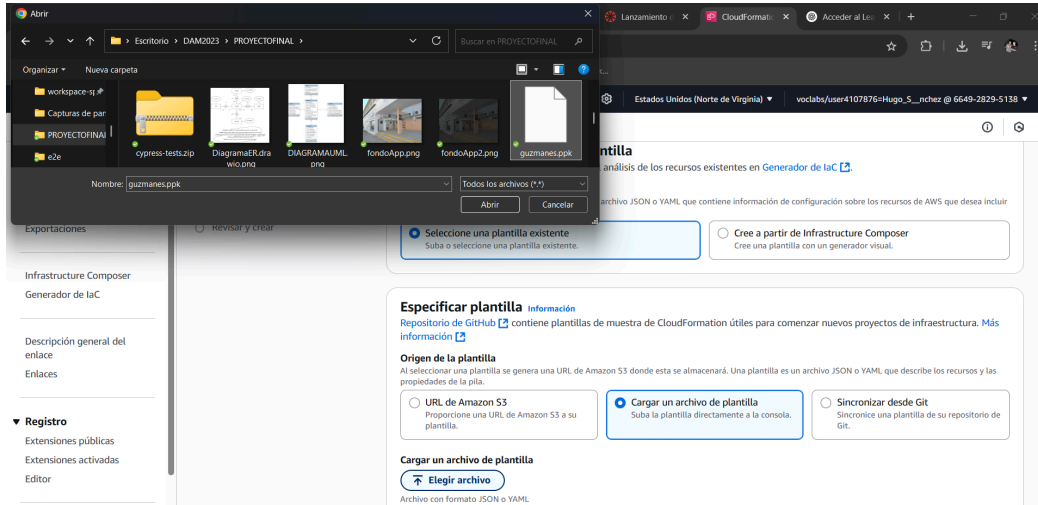
CloudFormation permite desplegar infraestructura como código. Una plantilla de CloudFormation en AWS es un archivo (normalmente en formato YAML o JSON) que describe la infraestructura que quieres crear en la nube, de forma automatizada y repetible.

A continuación, crearemos nuestra pila. Buscamos en nuestro buscador “CloudFormation” y accedemos al apartado de “Pilas”.

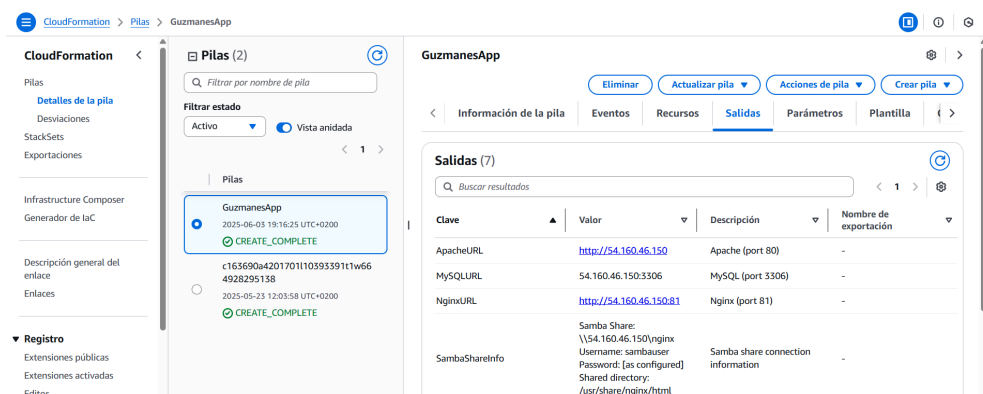
Una vez aquí, seleccionamos Crear Pila > Con recursos nuevos (estándar):



A la hora de especificar una plantilla, escogemos la opción de cargar un archivo de plantilla y buscamos nuestra [plantilla](#) descargada. Después le damos a “Siguiente” hasta llegar a “Crear Pila”.

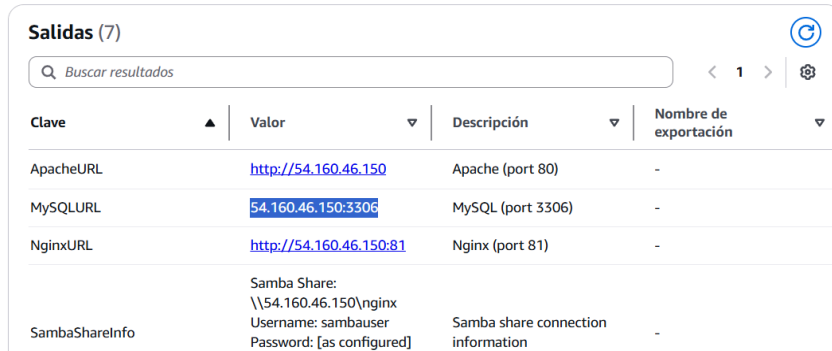


Una vez se cree la Pila, podremos comprobar en “Salidas” las rutas en las que se ha desplegado nuestro apache, nuestro tomcat, nuestra bbdd...



6. Paso 5. Conectar MySql Workbench a la base de datos.

Ahora buscamos la IP de MySQL, la cual usaremos para conectar nuestra base de datos:

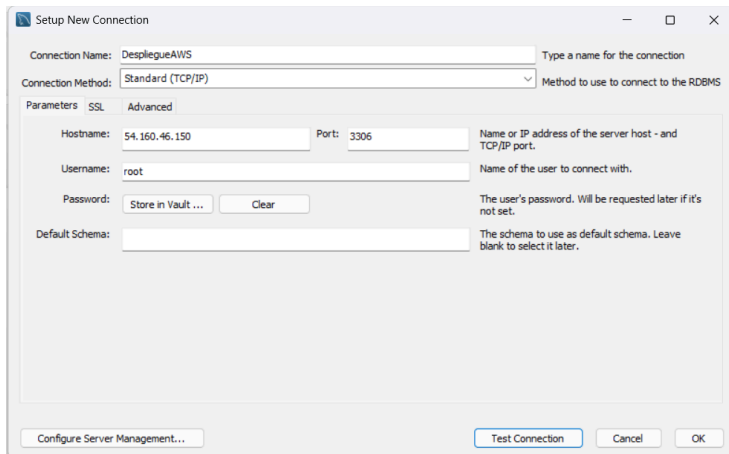


Clave	Valor	Descripción	Nombre de exportación
ApacheURL	http://54.160.46.150	Apache (port 80)	-
MySQLURL	54.160.46.150:3306	MySQL (port 3306)	-
NginxURL	http://54.160.46.150:81	Nginx (port 81)	-
SambaShareInfo	Samba Share: \\54.160.46.150\nginx Username: sambauser Password: [as configured]	Samba share connection information	-

Nos dirigimos a nuestro MySQL y creamos nuestra base de datos con la siguiente configuración:

- Tipo: MariaDB
- Host: IP/DNS de la base de datos
- Puerto: 3306
- Usuario: (root)
- Contraseña: (root1234)

Nos quedaría algo así:



Setup New Connection

Connection Name: DespliegueAWS Type a name for the connection

Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

Parameters **SSL** Advanced

Hostname: 54.160.46.150 Name or IP address of the server host - and TCP/IP port. Port: 3306

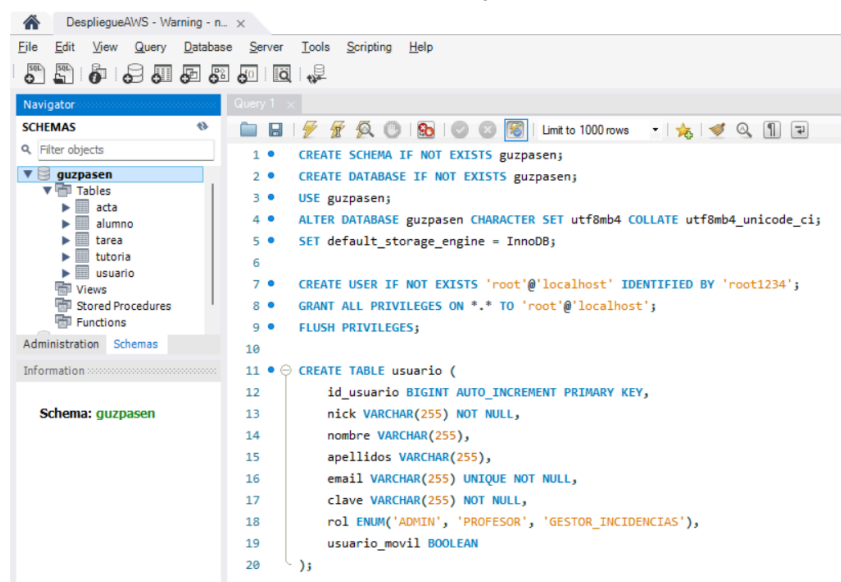
Username: root Name of the user to connect with.

Password: Store in Vault ... Clear The user's password. Will be requested later if it's not set.

Default Schema: The schema to use as default schema. Leave blank to select it later.

Configure Server Management... **Test Connection** Cancel OK

Por último, lanzamos nuestro script de creación de la base de datos.



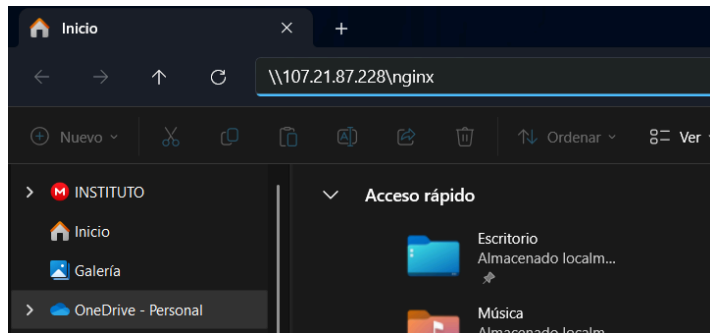
7. Paso 6. Mover empaquetados a la carpeta de Samba.

Samba es un software libre que permite compartir archivos y carpetas entre sistemas Linux/Unix y Windows a través de una red. Con Samba puedes configurar una carpeta en un servidor Linux para que sea accesible desde otros equipos (Windows, macOS o Linux), como si fuera una carpeta compartida de red.

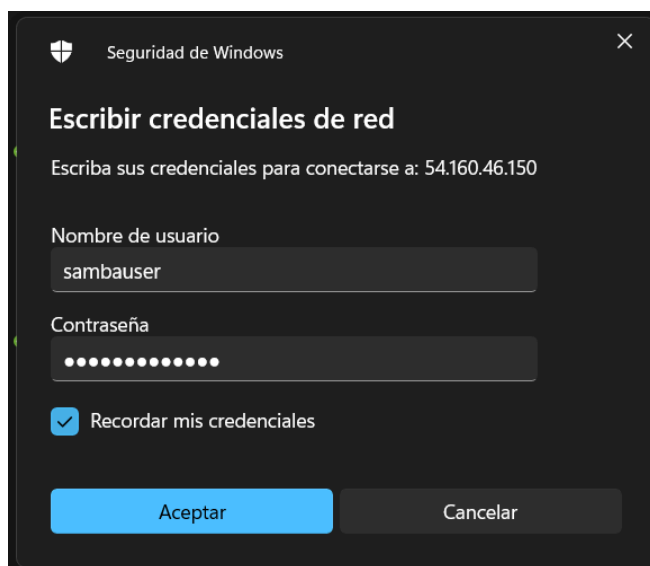
Al igual que antes, en el apartado de “Salida” de Cloudformation buscamos la ruta de Samba.

	Samba Share: \3.95.159.250\nginx	
SambaShareInfo	Username: sambauser Password: [as configured] Shared directory: /usr/ share/nginx/html	Samba share connection information -

Una vez tengamos nuestra base de datos lista, nos dirigimos a nuestro explorador de archivos, y pegamos la ruta de Samba:



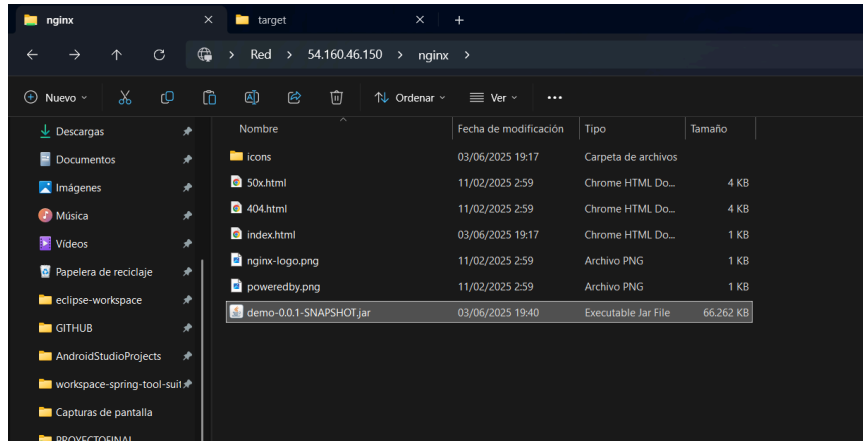
Si hemos realizado correctamente nos debe pedir un nombre de usuario y una contraseña:



Introducimos las siguientes credenciales:

- Nombre de suario: sambauser
- Contraseña: SambaPass1234

El último paso será pegar el .jar de nuestro proyecto en esta carpeta:

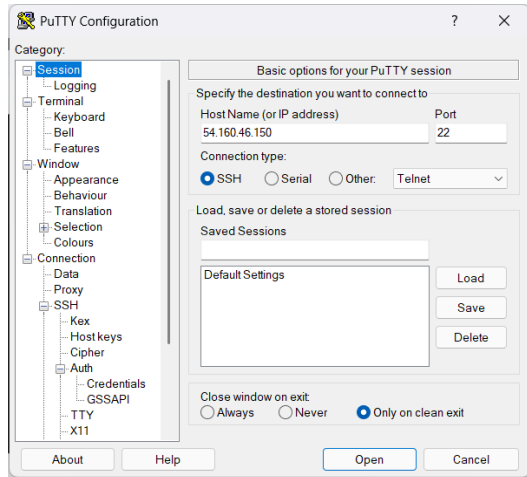


8. Paso 7. Instalar PuTTY y Conectarse por SSH.

PuTTY es un cliente SSH para Windows.

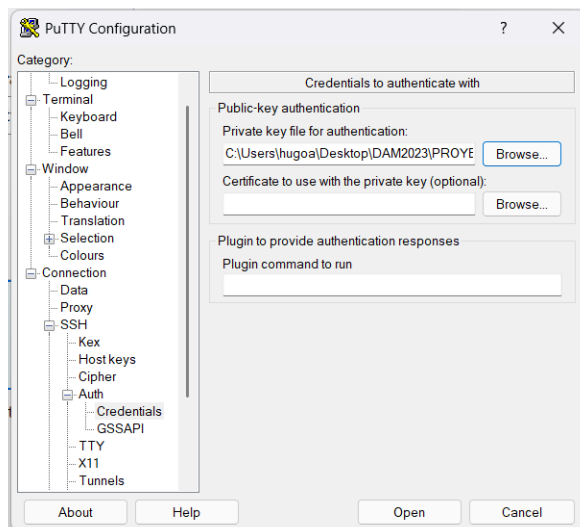
Para descargar PuTTY puedes hacerlo desde <https://www.putty.org>

A continuación, abrimos PuTTY y lo configuramos:



En “Host Name” introducimos la dirección IP de nuestro EC2.

Y en el apartado de Connection > SSH > Auth > Credentials adjuntamos nuestro archivo .ppk (Generado en el paso 3).



Ahora hacemos clic en Open, y si hemos realizado todos los pasos correctamente nos deberá aparecer una consola de Linux:

[illegible]

Nos logueamos con el usuario: ec2-user.

9. Paso 8. Desplegar los Microservicios con Spring Boot.

Por último, ejecutamos los siguientes comandos:

- `cd /usr/share/nginx/html`
- `java -jar archivo.jar`

```
ec2-user@ip-172-31-26-103:/usr/share/nginx/html
login as: ec2-user
Authenticating with public key "guzmanes"

#####
  ____          _  ____
 / ___|        / \|  / ___|
 \___ \        / _ \| |___ \
  ___) |       / ___ \| ___) |
 |_____|      /_/ |_| |_____|

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

Last login: Tue Jun  3 17:48:56 2025 from 86.127.226.80
[ec2-user@ip-172-31-26-103 ~]$ ls
[ec2-user@ip-172-31-26-103 ~]$ dir
[ec2-user@ip-172-31-26-103 ~]$ cd /usr/share/nginx/html
[ec2-user@ip-172-31-26-103 html]$ java -jar demo-0.0.1-SNAPSHOT.jar
```

Si se ha realizado todo correctamente, se nos ejecutará la aplicación y nos aparecerá lo siguiente:

```
ec2-user@ip-172-31-19-51:/usr/share/nginx/html
Isolation level: undefined/unknown
Minimum pool size: undefined/unknown
Maximum pool size: undefined/unknown
2025-06-05T10:30:00.037Z INFO 4457 --- [HSBGUZPASEN] [main] o.h.e.t.
j.p.i.JtaPlatformInitiator : HHH000489: No JTA platform available (set 'hi
bernate.transaction.jta.platform' to enable JTA platform integration)
2025-06-05T10:30:00.231Z INFO 4457 --- [HSBGUZPASEN] [main] j.LocalC
ontainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for pers
istence unit 'default'
2025-06-05T10:30:01.455Z WARN 4457 --- [HSBGUZPASEN] [main] JpaBaseC
onfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default
. Therefore, database queries may be performed during view rendering. Explicitly
configure spring.jpa.open-in-view to disable this warning
2025-06-05T10:30:02.275Z WARN 4457 --- [HSBGUZPASEN] [main] ion$Defa
ultTemplateResolverConfiguration : Cannot find template location: classpath:/tem
plates/ (please add some templates, check your Thymeleaf configuration, or set s
pring.thymeleaf.check-template-location=false)
2025-06-05T10:30:02.677Z INFO 4457 --- [HSBGUZPASEN] [main] o.s.b.w.
embedded.tomcat.TomcatWebServer : Tomcat started on port 8081 (http with conte
xt path '/')
2025-06-05T10:30:02.720Z INFO 4457 --- [HSBGUZPASEN] [main] guzpasen
.MyWeb : Started MyWeb in 12.11 seconds (process runni
ng for 13.405)
```