

# Desarrollo Web Integral y AWS



## Prerrequisitos

- [Crear una cuenta de AWS](#)
- [Crear una instancia EC2](#)
- [Instalar Docker](#)
- [Instalar NodeJs](#)
- [Instalar Nginx](#)
- [Instalar Git](#)

<b>Prerrequisitos</b>	<b>2</b>
<b>Introducción a Docker</b>	<b>4</b>
Instalando Amazon Elastic Compute Cloud - EC2	5
Instalar Nginx	10
<b>Para conocer la ruta de directorio actual ingresar comando</b>	<b>11</b>
Configuración HTTPS	12
<b>Creando contenedor Docker-Angular</b>	<b>15</b>
<b>Creando el contenedor Docker para Node</b>	<b>17</b>
<b>Docker MySQL</b>	<b>18</b>

# Introducción a Docker

En Docker tenemos la posibilidad de que las aplicaciones estén aisladas en contenedores seguros que además de contener la aplicación, permite un mejor y más fácil modo de controlarlas.

Docker es una plataforma de software que permite crear, probar e implementar aplicaciones rápidamente. Docker empaqueta software en unidades estandarizadas llamadas contenedores que incluyen todo lo necesario para que el software se ejecute, incluidas bibliotecas, herramientas de sistema, código y tiempo de ejecución. Con Docker, puede implementar y ajustar la escala de aplicaciones rápidamente en cualquier entorno con la certeza de saber que su código se ejecutará.

## Cómo funciona Docker

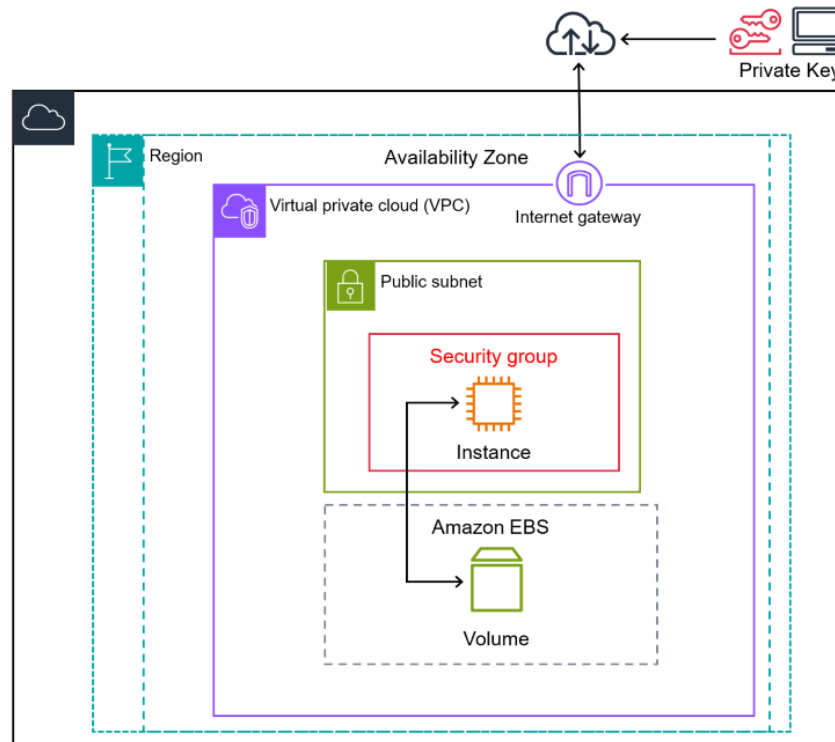
Docker proporciona una manera estándar de ejecutar código. Docker es un sistema operativo para contenedores. De manera similar a cómo una máquina virtual virtualiza (elimina la necesidad de administrar directamente) el hardware del servidor, los contenedores virtualiza el sistema operativo de un servidor. Docker se instala en cada servidor y proporciona comandos sencillos que puede utilizar para crear, iniciar o detener contenedores.

## Por qué usar Docker

Docker permite entregar código con mayor rapidez, estandarizar las operaciones de las aplicaciones, transferir el código con facilidad y ahorrar dinero al mejorar el uso de recursos. Con Docker, se obtiene un solo objeto que se puede ejecutar de manera fiable en cualquier lugar. La sintaxis sencilla y simple de Docker le aporta un control absoluto. La amplia adopción significa que existe un gran ecosistema de herramientas y aplicaciones listas para su uso que puede utilizar con Docker.

# Instalando Amazon Elastic Compute Cloud - EC2

**Amazon Elastic Compute Cloud (Amazon EC2)** proporciona capacidad de computación escalable bajo demanda en la nube de Amazon Web Services (AWS). El uso de Amazon EC2 reduce los costos de hardware para que pueda desarrollar e implementar aplicaciones con mayor rapidez. Puede usar Amazon EC2 para lanzar tantos servidores virtuales como necesite, configurar la seguridad y las redes, y administrar el almacenamiento. Puede agregar capacidad (escalar verticalmente) para gestionar tareas que requieren mucha computación, como los procesos mensuales o anuales, o los picos de tráfico del sitio web. Cuando el uso disminuye, puede volver a reducir la capacidad (reducir verticalmente).



## 1.1. Crear una instancia EC2.

Ver el [tutorial](#) para creación de una instancia EC2

EC2 > Instancias > i-0b01febf11f7f9b28

**Resumen de instancia de i-0b01febf11f7f9b28 (lggc.kdac.backend)** información

Se ha actualizado hace less than a minute

<p>ID de la instancia</p> <p>i-0b01febf11f7f9b28 (lggc.kdac.backend)</p> <p>Dirección IPv6</p> <p>–</p> <p>Tipo de nombre de anfitrión</p> <p>Nombre de IP: ip-172-31-30-107.us-west-1.compute.internal</p> <p>Responder al nombre DNS de recurso privado IPv4 (A)</p> <p>Dirección IP asignada automáticamente</p> <p>54.183.180.16 [IP pública]</p> <p>Rol de IAM</p> <p>–</p> <p>IMDSv2</p> <p>Optional</p>	<p>Dirección IPv4 pública</p> <p>54.183.180.16   <a href="#">dirección abierta</a></p> <p>Estado de la instancia</p> <p>En ejecución</p> <p>Nombre DNS de IP privada (solo IPv4)</p> <p>ip-172-31-30-107.us-west-1.compute.internal</p> <p>Tipo de instancia</p> <p>t2.micro</p> <p>ID de VPC</p> <p>vpc-0aa3e8d1a52f0ad54</p> <p>ID de subred</p> <p>subnet-0611986f14df79cf1</p>	<p>Direcciones IPv4 privadas</p> <p>172.31.30.107</p> <p>DNS de IPv4 pública</p> <p>ec2-54-183-180-16.us-west-1.compute.amazonaws.com   <a href="#">dirección abierta</a></p> <p>Direcciones IP elásticas</p> <p>–</p> <p>Hallazgo de AWS Compute Optimizer</p> <p><a href="#">Suscribirse a AWS Compute Optimizer para recibir recomendaciones.</a>   <a href="#">Más información</a></p> <p>Nombre del grupo de Auto Scaling</p> <p>–</p>
--	--	---

## 2.1. Seleccionar instancia de Ubuntu 20.04 o superior.

Detalles Seguridad Redes Almacenamiento Comprobaciones de estado Monitoreo Etiquetas

▼ Detalles de la instancia información

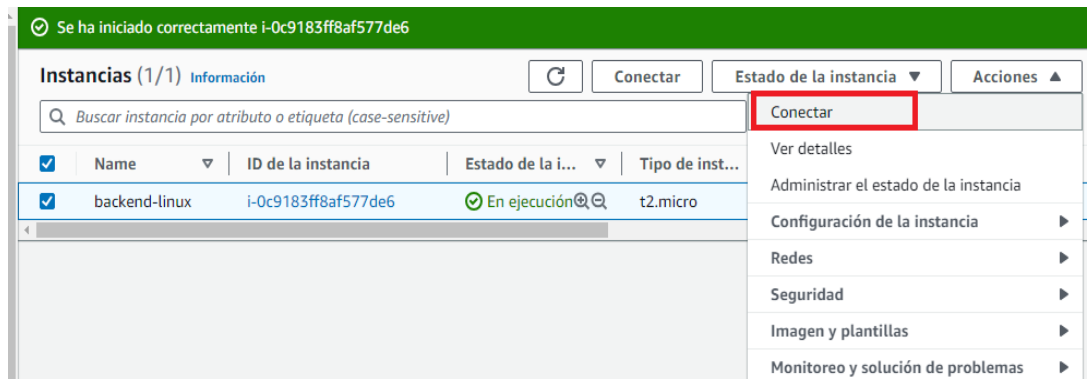
<p>Plataforma</p> <p>Ubuntu (inferido)</p> <p>Detalles de la plataforma</p> <p>Linux/UNIX</p> <p>Detener la protección</p> <p>desactivado</p>	<p>ID de AMI</p> <p>ami-0f8e81a3da6e2510a</p> <p>Nombre de AMI</p> <p>ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20230516</p> <p>Hora de lanzamiento</p> <p>Fri Aug 04 2023 16:46:46 GMT-0600 (hora estándar central) (1 minute)</p>	<p>Monitoreo</p> <p>desactivado</p> <p>Protección de terminación</p> <p>desactivado</p> <p>Ubicación de AMI</p> <p>amazon/ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20230516</p>
---	---	--

## 3.1. Guarda las claves privadas en un lugar donde te acuerdes.

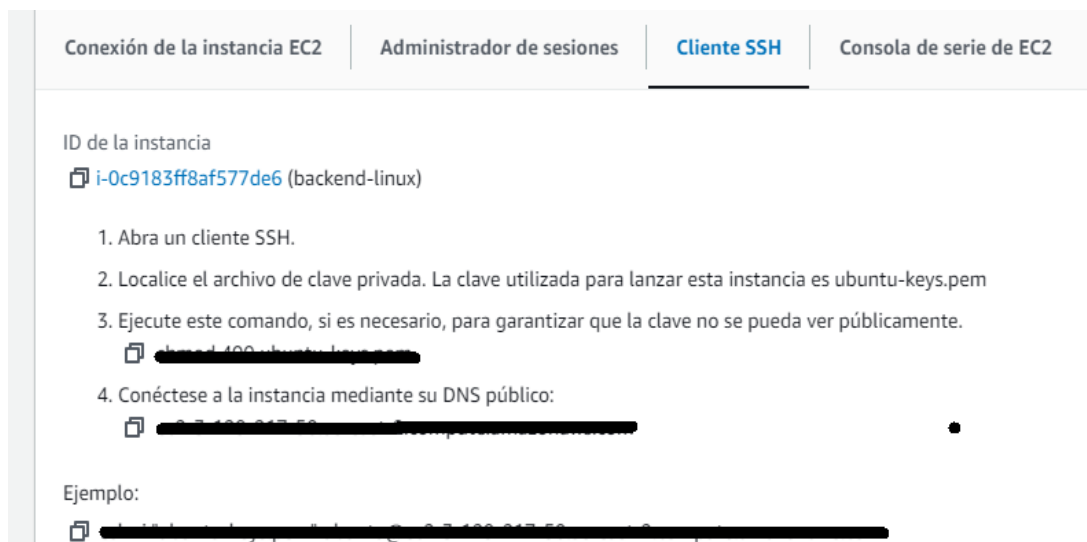
Este equipo > Disco local (D:) > UTNG > C8 > DWI > U-IV > AWS

Nombre	Fecha de modificación	Tipo	Tamaño
backend-keys.pem	04/08/2023 04:44 p. m.	CMS (S/MIME) File	2 KB

4.1. Accede a la instancia de a través de Putty, para mayor referencia observa la siguiente imagen



5.1. Seleccionar pestaña Cliente SSH, Observa los parámetros de conexión



## 6.1. Abrir consola y ubicar archivo de claves y ejecutar comando

```
C:\Users\usuario\Downloads>ssh -i "ubuntu-keys.pem" ubuntu@ec2-3-129-217-50.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-3-129-217-50.us-east-2.compute.amazonaws.com (3.129.217.50)' can't be established.
ECDSA key fingerprint is SHA256:K3907Cdac3IzSpiyap304W1pCQfr53oqHu6dx7Wwsws.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

### 7.1. Listo

## Práctica de la Lección

Una vez ingresado a la consola de la instancia ejecutar los siguientes comandos:

- Listar el contenido del directorio actual

```
ubuntu@ip-172-31-30-107:~$ ls
lg_kd_ejemplo  web-integral
```

- Ejecutar comando que indica la ruta de directorio actual.

```
ubuntu@ip-172-31-30-107:~$ pwd
/home/ubuntu
```

- Crear un directorio llamado workdir y cambiarse al directorio.

```
ubuntu@ip-172-31-30-107:~$ mkdir workdir && cd workdir
ubuntu@ip-172-31-30-107:~/workdir$ |
```

- Crear un archivo llamado **welcome.txt** que tenga como contenido “Bienvenidos a Ubuntu”; despliega el contenido del archivo a través de línea de comandos.

```
ubuntu@ip-172-31-30-107:~/workdir$ echo "Bienvenidos a Ubuntu" > welcome.txt
ubuntu@ip-172-31-30-107:~/workdir$ cat welcome.txt
Bienvenidos a Ubuntu
```

- Ejecutar comando que instala la última versión de los paquetes

```
ubuntu@ip-172-31-30-107:~/workdir$ sudo apt-get update && sudo apt-get upgrade
Hit:1 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Hit:4 https://download.docker.com/linux/ubuntu focal InRelease
```

## 1. Creando una aplicación Angular

2.1. Ingresar a la consola EC2.

2.2. Crear un directorio de trabajo.

2.3. Instalar [NodeJs](#).

```
ubuntu@ip-172-31-30-107:~$ node -v
v18.17.0
ubuntu@ip-172-31-30-107:~$ npm -v
9.6.7
```

2.4. Instalar Angular CLI

```
sudo pm install -g @angular/cli@latest
```



```
ubuntu@ip-172-31-30-107:~$ ng version
? Would you like to enable autocompletion? This will set up your terminal so pressing TAB while typing
Angular CLI commands will show possible options and autocomplete arguments. (Enabling autocompletion
will modify configuration files in your home directory.) Yes
Appended 'source <(ng completion script)>' to '/home/ubuntu/.bashrc'. Restart your terminal or run the f
ollowing to autocomplete 'ng' commands:

  source <(ng completion script)
? Would you like to share pseudonymous usage data about this project with the Angular Team
at Google under Google's Privacy Policy at https://policies.google.com/privacy. For more
details and how to change this setting, see https://angular.io/analytics. No
Global setting: disabled
Local setting: No local workspace configuration file.
Effective status: disabled

Angular CLI

Angular CLI: 16.1.8
Node: 18.17.0
Package Manager: npm 9.8.1
OS: linux x64

Angular:
...
Package      Version
-----
@angular-devkit/architect    0.1601.8 (cli-only)
@angular-devkit/core         16.1.8 (cli-only)
@angular-devkit/schematics   16.1.8 (cli-only)
@schematics/angular          16.1.8 (cli-only)
```

## 2.5. Crear una nueva aplicación Angular

`ng new lg_kd_ejemplo --strict --style=scss --routing --skip-tests`

```
ubuntu@ip-172-31-30-107:~$ ng new lg_kd_ejemplo --strict --style=scss --routing --skip-tests
CREATE lg_kd_ejemplo/README.md (1065 bytes)
CREATE lg_kd_ejemplo/.editorconfig (274 bytes)
CREATE lg_kd_ejemplo/.gitignore (548 bytes)
CREATE lg_kd_ejemplo/angular.json (3498 bytes)
CREATE lg_kd_ejemplo/package.json (1044 bytes)
CREATE lg_kd_ejemplo/tsconfig.json (901 bytes)
CREATE lg_kd_ejemplo/tsconfig.app.json (263 bytes)
CREATE lg_kd_ejemplo/tsconfig.spec.json (273 bytes)
CREATE lg_kd_ejemplo/.vscode/extensions.json (130 bytes)
CREATE lg_kd_ejemplo/.vscode/launch.json (470 bytes)
CREATE lg_kd_ejemplo/.vscode/tasks.json (938 bytes)
CREATE lg_kd_ejemplo/src/main.ts (214 bytes)
CREATE lg_kd_ejemplo/src/favicon.ico (948 bytes)
CREATE lg_kd_ejemplo/src/index.html (297 bytes)
```

## 2.6. Cambiarse al directorio recién creado de la aplicación

`cd lg_kd_ejemplo`

```
ubuntu@ip-172-31-30-107:~$ cd lg_kd_ejemplo/
ubuntu@ip-172-31-30-107:~/lg_kd_ejemplo$ |
```

## 2.7. Configurar la instancia EC2, seleccionar pestaña Seguridad y agregar las nuevas reglas

Detalles

Seguridad

Redes

Almacenamiento

Comprobaciones de estado

Monitoreo


Etiquetas

▼ Detalles de seguridad

Rol de IAM

–


ID del propietario

 627304545074

Hora de lanzamiento

Fri Aug 04 2023 16:46:46 GMT-0600 (hora estándar central)

Grupos de seguridad

 sg-0f5b58cd1ba006943 (launch-wizard-1)

▼ Reglas de entrada





Q

Filtrar reglas

<

1

>

Nombre	ID de la regla del grupo d...	Intervalo de pu...	Protocolo	Origen	Grupos de seguridad	Descripción
–	sg-0178405365984274b	4200	TCP	0.0.0/0/0	launch-wizard-1 	–
–	sg-0764ae8a148a3e693	80	TCP	0.0.0/0/0	launch-wizard-1 	–
–	sg-0c3a03f9d284a94b	443	TCP	0.0.0/0/0	launch-wizard-1 	–
–	sg-04074a906584df4d9	22	TCP	0.0.0/0/0	launch-wizard-1 	–

2.8. Iniciar el servidor para conocer ejecución de la herramienta

`ng serve --host 0.0.0.0`

```
ubuntu@ip-172-31-30-107:~/lg_kd_ejemplo$ ng serve --host 0.0.0.0
Warning: This is a simple server for use in testing or debugging Angular applications
locally. It hasn't been reviewed for security issues.

Binding this server to an open connection can result in compromising your application or
computer. Using a different host than the one passed to the "--host" flag might result in
websocket connection issues. You might need to use "--disable-host-check" if that's the
case.

✓ Browser application bundle generation complete.

Initial Chunk Files | Names | Raw Size
vendor.js           | vendor | 2.28 MB
polyfills.js        | polyfills | 333.16 kB
styles.css, styles.js | styles | 230.92 kB
main.js             | main | 48.11 kB
runtime.js          | runtime | 6.52 kB
                    | Initial Total | 2.89 MB

Build at: 2023-08-04T23:46:39.136Z - Hash: ad1b32aa98236b8e - Time: 24365ms
** Angular Live Development Server is listening on 0.0.0.0:4200, open your browser on http://localhost:4200/ **

✓ Compiled successfully.
```

2.9. Copiar la IP pública de la instancia y pegarla en el navegador Web, acceder a través del protocolo HTTP y agregar el puerto 4200 de angular

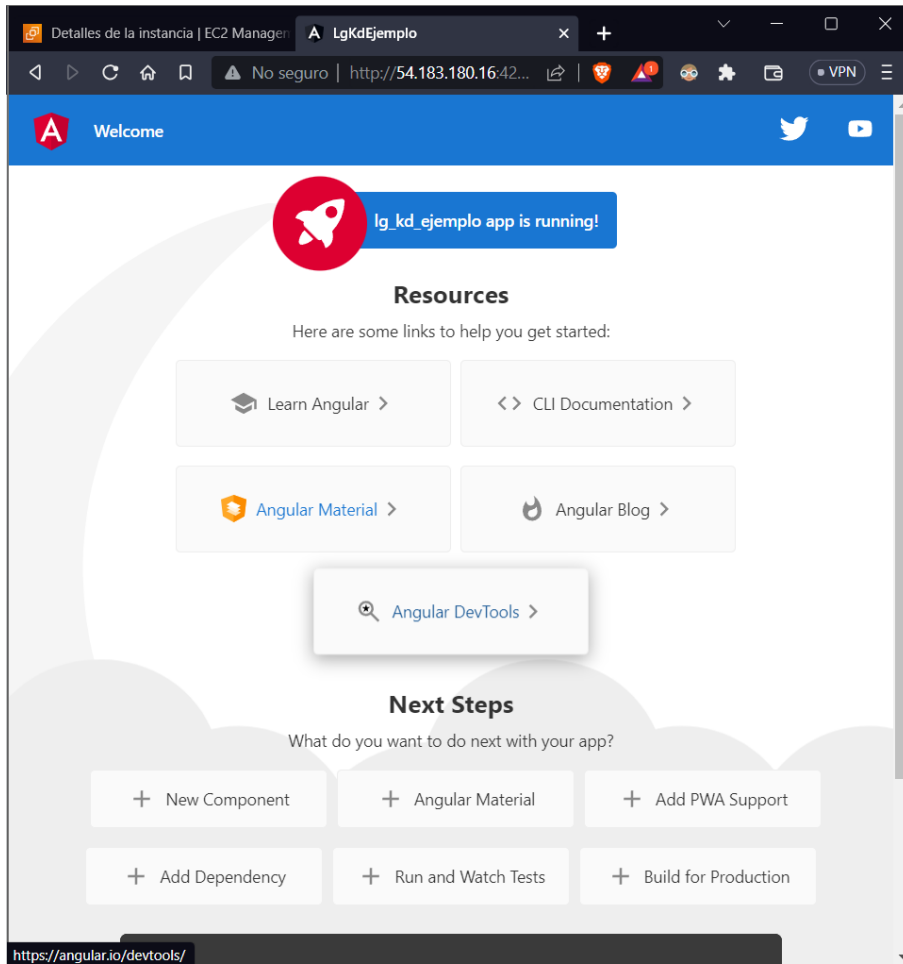
### Resumen de instancia de i-0b01fefb11f7f9b28 (lggc.kdac.backend) [Información](#)

Se ha actualizado hace less than a minute

 [Conectar](#) [Estado de la instancia ▼](#) [Acciones ▼](#)

ID de la instancia  i-0b01fefb11f7f9b28 (lggc.kdac.backend)	Dirección IPv4 pública  54.183.180.16   <a href="#">dirección abierta</a> 
Direcciones IPv4 privadas  172.31.30.107	Dirección IPv6 -
Estado de la instancia  <b>En ejecución</b>	DNS de IPv4 pública  ec2-54-183-180-16.us-west-1.compute.amazonaws.com   <a href="#">dirección abierta</a> 

2.10. Abrir un navegador web e ingresar la dirección IP junto con su puerto



# Instalar Nginx

## Referencia

Instalar las fuentes de configuración

**sudo apt update**

```
ubuntu@ip-172-31-30-107:~/workdir$ sudo apt update
Hit:1 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
```

Instalar el servidor Web Nginx con el comando

**sudo apt install nginx**

```
ubuntu@ip-172-31-30-107:~/workdir$ sudo apt install nginx
Reading package lists... Done
Building dependency tree... Done
```

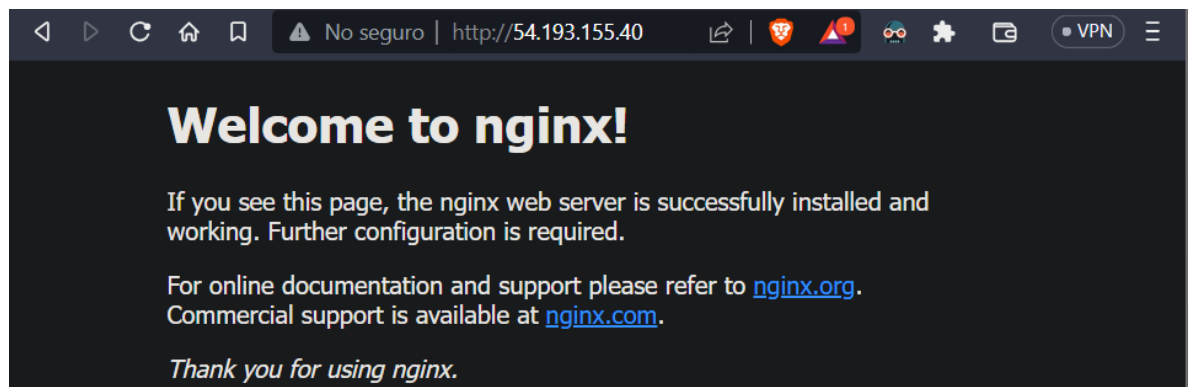
Verificar el estado del servidor web

**systemctl status nginx**

```
ubuntu@ip-172-31-30-107:~/workdir$ systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-08-08 23:02:46 UTC; 2min 49s ago
     Docs: man:nginx(8)
  Main PID: 12148 (nginx)
    Tasks: 2 (limit: 1141)
   Memory: 4.4M
      CPU: 24ms
   CGroup: /system.slice/nginx.service
           └─12148 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─12151 "nginx: worker process"

Aug 08 23:02:46 ip-172-31-30-107 systemd[1]: Starting A high performance web server and a reverse proxy server: nginx.
Aug 08 23:02:46 ip-172-31-30-107 systemd[1]: Started A high performance web server and a reverse proxy server: nginx.
lines 1-14/14 (END)
```

Verificar qué se levantó de manera correcta el servidor web, ingresa la IP Pública de EC2



Ahora vamos a construir el proyecto test, ingresar el comando

**ng build**

```

ubuntu@ip-172-31-30-107:~/lg_kd_ejemplo$ ng build
✓ Browser application bundle generation complete.
✓ Copying assets complete.
✓ Index html generation complete.

Initial Chunk Files | Names | Raw Size | Estimated Transfer Size
main.c08da3a59ab2eb4e.js | main | 204.56 kB | 55.92 kB
polyfills.54a0363253be8951.js | polyfills | 33.02 kB | 10.66 kB
runtime.7700d59d10d49011.js | runtime | 904 bytes | 512 bytes
styles.ef46db3751d8e999.css | styles | 0 bytes | -

Initial Total | 238.46 kB | 67.09 kB

Build at: 2023-08-08T23:17:49.403Z - Hash: a98cf6de4b73ed18 - Time: 9064ms

```

Ingresar el comando para el listado de carpetas y archivos

**ls -l**

```

ubuntu@ip-172-31-30-107:~/lg_kd_ejemplo$ ls -l
total 476
-rw-r--r-- 1 root root 743 Aug 4 23:59 Dockerfile
-rw-rw-r-- 1 ubuntu ubuntu 1065 Aug 4 23:28 README.md
-rw-rw-r-- 1 ubuntu ubuntu 3498 Aug 4 23:28 angular.json
drwxrwxr-x 3 ubuntu ubuntu 4096 Aug 8 23:17 dist
drwxrwxr-x 560 ubuntu ubuntu 20480 Aug 4 23:28 node_modules
-rw-rw-r-- 1 ubuntu ubuntu 426868 Aug 4 23:28 package-lock.json
-rw-rw-r-- 1 ubuntu ubuntu 1044 Aug 4 23:28 package.json
drwxrwxr-x 4 ubuntu ubuntu 4096 Aug 4 23:28 src
-rw-rw-r-- 1 ubuntu ubuntu 263 Aug 4 23:28 tsconfig.app.json
-rw-rw-r-- 1 ubuntu ubuntu 901 Aug 4 23:28 tsconfig.json
-rw-rw-r-- 1 ubuntu ubuntu 273 Aug 4 23:28 tsconfig.spec.json

```

Para conocer la ruta de directorio actual ingresar comando

**pwd**

```

ubuntu@ip-172-31-30-107:~/lg_kd_ejemplo$ pwd
/home/ubuntu/lg_kd_ejemplo

```

Copiar la ruta y agregar la carpeta dist/test, por ejemplo:

**/home/ubuntu/lg\_kd\_ejemplo/dist/lg\_kd\_ejemplo**

```

ubuntu@ip-172-31-30-107:~/lg_kd_ejemplo/dist/lg_kd_ejemplo$ pwd
/home/ubuntu/lg_kd_ejemplo/dist/lg_kd_ejemplo

```

Modificar el archivo de configuración de Nginx

**sudo nano /etc/nginx/sites-available/default**

```

ubuntu@ip-172-31-30-107:~/lg_kd_ejemplo/dist/lg_kd_ejemplo$ sudo nano /etc/nginx/sites-available/default

```

Modificar línea: root **/var/www/html**; Por ruta: **/home/ubuntu/workdir/test/dist/test**

```
GNU nano 6.2 /etc/nginx/sites-available/default *
#
# Self signed certs generated by the ssl-cert package
# Don't use them in a production server!
#
# include snippets/snakeoil.conf;

root /home/ubuntu/lg_kd_ejemplo/dist/lg_kd_ejemplo;

# Add index.php to the list if you are using PHP
index index.html index.htm index.nginx-debian.html;

server_name _;

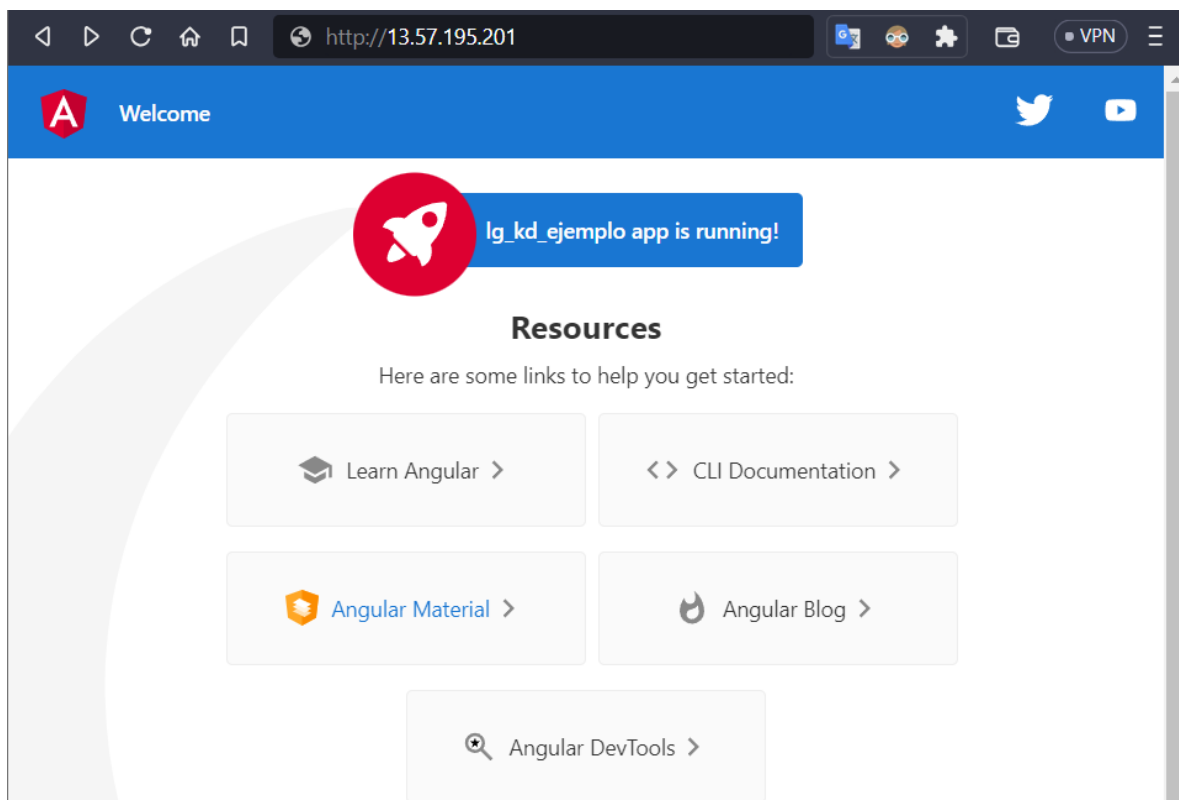
location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ =404;
}
```

Reiniciar servidor web con comando

**sudo systemctl restart nginx**

```
ubuntu@ip-172-31-30-107:~$ sudo systemctl restart nginx
ubuntu@ip-172-31-30-107:~$
```

Ingresar nuevamente la IP de la dirección pública de EC2 en su navegador



```
ubuntu@ip-172-31-30-107:~/ssl$ openssl req -new -key clave-privada.key -out cert.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
```

```

Country Name (2 letter code) [AU]:MX
State or Province Name (full name) [Some-State]:GTO
Locality Name (eg, city) []:DH
Organization Name (eg, company) [Internet Widgits Pty Ltd]:UTNG
Organizational Unit Name (eg, section) []:TICS
Common Name (e.g. server FQDN or YOUR name) []:DWI
Email Address []:luisggc321@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:QaZwSx3
An optional company name []:UTNG
ubuntu@ip-172-31-30-107:~/ssl$

```

Generar nuevamente certificado firmado

```

openssl x509 -req -days 365 -in /home/ubuntu/ssl/cert.csr -signkey
/home/ubuntu/ssl/clave-privada.key -out /home/ubuntu/ssl/cert-auto.crt

```

```

cert.csr clave-privada.key
ubuntu@ip-172-31-30-107:~/ssl$ openssl x509 -req -days 365 -in /home/ubuntu/ssl/
cert.csr -signkey /home/ubuntu/ssl/clave-privada.key -out /home/ubuntu/ssl/cert-
auto.crt
Certificate request self-signature ok
subject=C = MX, ST = GTO, L = DH, O = UTNG, OU = TICS, CN = DWI, emailAddress =
luisggc321@gmail.com

```

Unir ambos certificados para mayor seguridad

```

cat cert-auto.crt clave-privada.key > cert-completo.crt

```

```

ubuntu@ip-172-31-30-107:~/ssl$ ls -l
total 16
-rw-rw-r-- 1 ubuntu ubuntu 1265 Aug 16 10:13 cert-auto.crt
-rw-rw-r-- 1 ubuntu ubuntu 2969 Aug 16 10:14 cert-completo.crt
-rw-rw-r-- 1 ubuntu ubuntu 1090 Aug 16 10:11 cert.csr
-rw----- 1 ubuntu ubuntu 1704 Aug 11 22:34 clave-privada.key
ubuntu@ip-172-31-30-107:~/ssl$

```



Modificar archivo en servidor web Nginx

```
sudo nano /etc/nginx/sites-available/default
```

```
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # SSL configuration
    listen 443 ssl default_server;
    ssl_certificate /home/ubuntu/ssl/cert-completo.crt;
    ssl_certificate_key /home/ubuntu/ssl/clave-privada.key;
    # listen [::]:443 ssl default_server;
```

CTRL + S Guardar

CTRL + X Salir

Reiniciar nuevamente el servidor

```
sudo systemctl restart nginx
```

```
ubuntu@ip-172-31-30-107:~/ssl$ sudo systemctl restart nginx
ubuntu@ip-172-31-30-107:~/ssl$ systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-08-16 10:44:46 UTC; 3s ago
```

Abrir navegador web con la direccion: <https://54.176.162.127/>

**Resumen de instancia de i-0b01fefb11f7f9b28**

(lggc.kdac.backend) [Información](#)

Se ha actualizado hace less than a minute

Conectar

Estado de la instancia ▼

Acciones ▼

ID de la instancia  
i-0b01fefb11f7f9b28 (lggc.kdac.backend)

Dirección IPv4 pública  
54.176.162.127 | [dirección abierta](#)

No seguro | <https://54.176.162.127>

La conexión no es privada

Es posible que algunos atacantes intenten robar tu información de 54.176.162.127 (p. ej., contraseñas, mensajes o tarjetas de crédito). [Más información](#)

NET::ERR\_CERT\_AUTHORITY\_INVALID

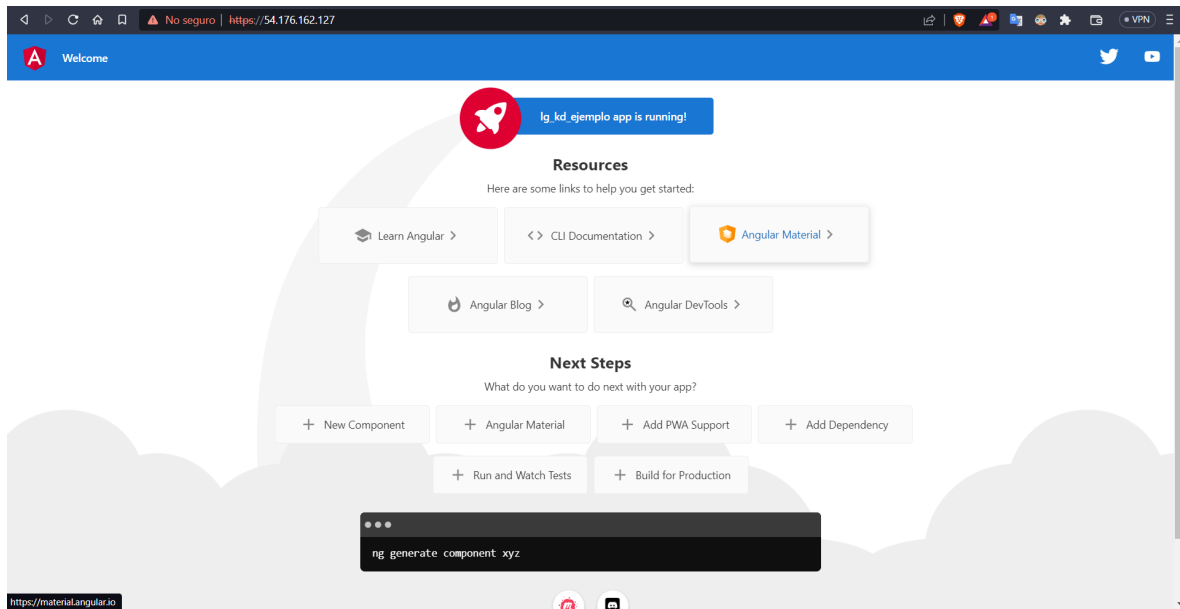
Ocultar detalles avanzados

Volver a un sitio seguro

Este servidor no pudo probar que su dominio es 54.176.162.127; el sistema operativo de la computadora no confía en el certificado de seguridad. Es posible que esto se deba a una configuración incorrecta o a que un atacante interceptó la conexión.

[Continuar a 54.176.162.127 \(no seguro\)](#)

Seleccionar configuración avanzada | Continuar a 54.176.162.127 (no seguro)



# Instalar MySQL

### Referencia

Crear una nueva instancia donde se deposite la base de datos.

Resumen de instancia de i-0d39a9a6f6c46b936 (bd\_luis-karla) Información

Se ha actualizado hace less than a minute

Estado de la instancia ▼

Acciones ▼

ID de la instancia

i-0d39a9a6f6c46b936 (bd\_luis-karla)

Dirección IPv6

–

Tipo de nombre de anfitrión

Nombre de IP: ip-172-31-6-244.us-west-1.compute.internal

Responder al nombre DNS de recurso privado IPv4 (A)

Dirección IP asignada automáticamente

54.67.110.131 [IP pública]

Rol de IAM

–

IMDSv2

Optional

Dirección IPv4 pública

54.67.110.131 | [dirección abierta](#)

Estado de la instancia

Pendiente

Nombre DNS de IP privada (solo IPv4)

ip-172-31-6-244.us-west-1.compute.internal

Tipo de instancia

t2.micro

ID de VPC

vpc-0aa3e8d1a52f0ad54

ID de subred

subnet-09494a1545cb10cb38

Direcciones IPv4 privadas

172.31.6.244

DNS de IPv4 pública

ec2-54-67-110-131.us-west-1.compute.amazonaws.com | [dirección abierta](#)

Direcciones IP elásticas

–

Hallazgo de AWS Compute Optimizer

[Suscribirse a AWS Compute Optimizer para recibir recomendaciones.](#) | [Más información](#)

Nombre del grupo de Auto Scaling

–

Detalles

Seguridad

Redes

Almacenamiento

Comprobaciones de estado

Monitoreo

Etiquetas

▼ Detalles de la instancia Información

Plataforma

Ubuntu (inferido)

Detalles de la plataforma

Linux/UNIX

ID de AMI

ami-0f8e81a3da6e2510a

Nombre de AMI

ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20230516

Monitoreo

desactivado

Protección de terminación

desactivado

## Abrir el puerto 3306 de EC2

Reglas de entrada (2)									
<input type="text" value="Filtrar reglas de grupo de seguridad"/>									
<input type="checkbox"/>	Name	ID de la regla del g...	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción	
<input type="checkbox"/>	-	sg-r-0762f07f02ed29767	IPv4	SSH	TCP	22	0.0.0.0/0	-	
<input type="checkbox"/>	-	sg-r-0e87b669d77dfdfa2	IPv4	MYSQL/Aurora	TCP	3306	0.0.0.0/0	-	

## Instalar la base de datos MySQL

```
ubuntu@ip-172-31-6-244:~$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libbcgi-fast-perl libbcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7
  libfcgi-bin libfcgi-perl libfcgi0ldbl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2
  libprotobuf-lite23 libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils
  mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libbusiness-isbn-perl libwww-perl mailx tinyca
The following NEW packages will be installed:
  libbcgi-fast-perl libbcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7
  libfcgi-bin libfcgi-perl libfcgi0ldbl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2
  libprotobuf-lite23 libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils
  mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server mysql-server-8.0
  mysql-server-core-8.0
0 upgraded, 28 newly installed, 0 to remove and 109 not upgraded.
```

```
ubuntu@ip-172-31-6-244:~$ sudo systemctl start mysql.service
```

Crear la base de datos web\_integral;

```

ubuntu@ip-172-31-6-244:~$ mysql -u lgarcia -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 8.0.34-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2023, 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> CREATE DATABASE web_integral;
Query OK, 1 row affected (0.01 sec)

mysql> USE web_integral;
Database changed
mysql>

```

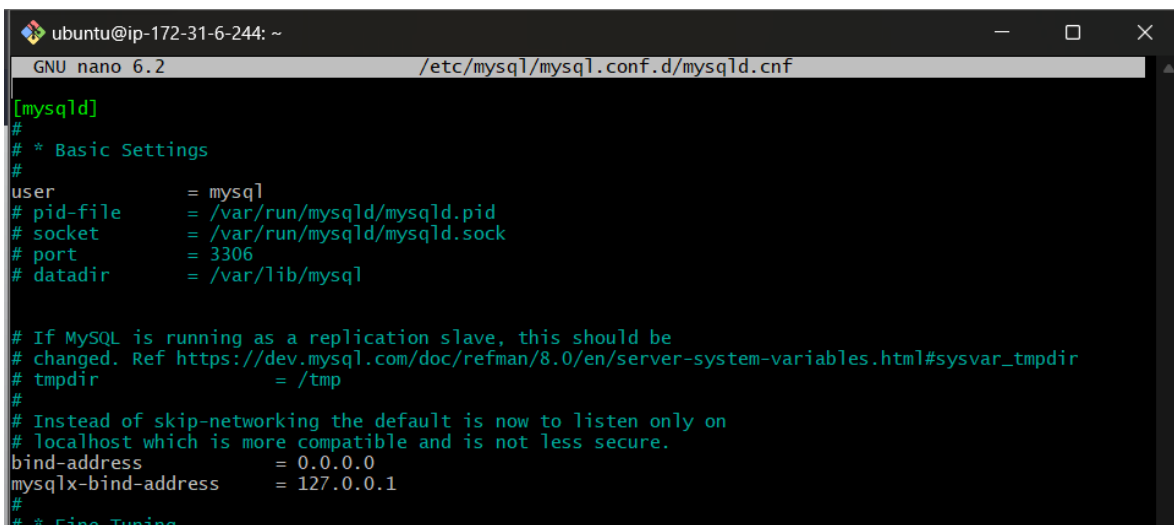
Crear tabla users:

```

mysql> CREATE TABLE tbl_users (
  -> Username VARCHAR(30) PRIMARY KEY,
  -> Password VARCHAR(250) NOT NULL,
  -> Role VARCHAR(30) NOT NULL
  -> );
Query OK, 0 rows affected (0.03 sec)

```

[Configurar base de datos para acceso remoto](#)



```

ubuntu@ip-172-31-6-244: ~
GNU nano 6.2 /etc/mysql/mysql.conf.d/mysqld.cnf

[mysqld]
#
# * Basic Settings
#
user                = mysql
# pid-file           = /var/run/mysqld/mysqld.pid
# socket             = /var/run/mysqld/mysqld.sock
# port               = 3306
# datadir            = /var/lib/mysql

# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir             = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address         = 0.0.0.0
mysqlx-bind-address  = 127.0.0.1
#
# * Fine Tuning

```

```

mysql> CREATE USER 'lgkd'@'%' IDENTIFIED BY 'MyP@sswOrd';
Query OK, 0 rows affected (0.03 sec)

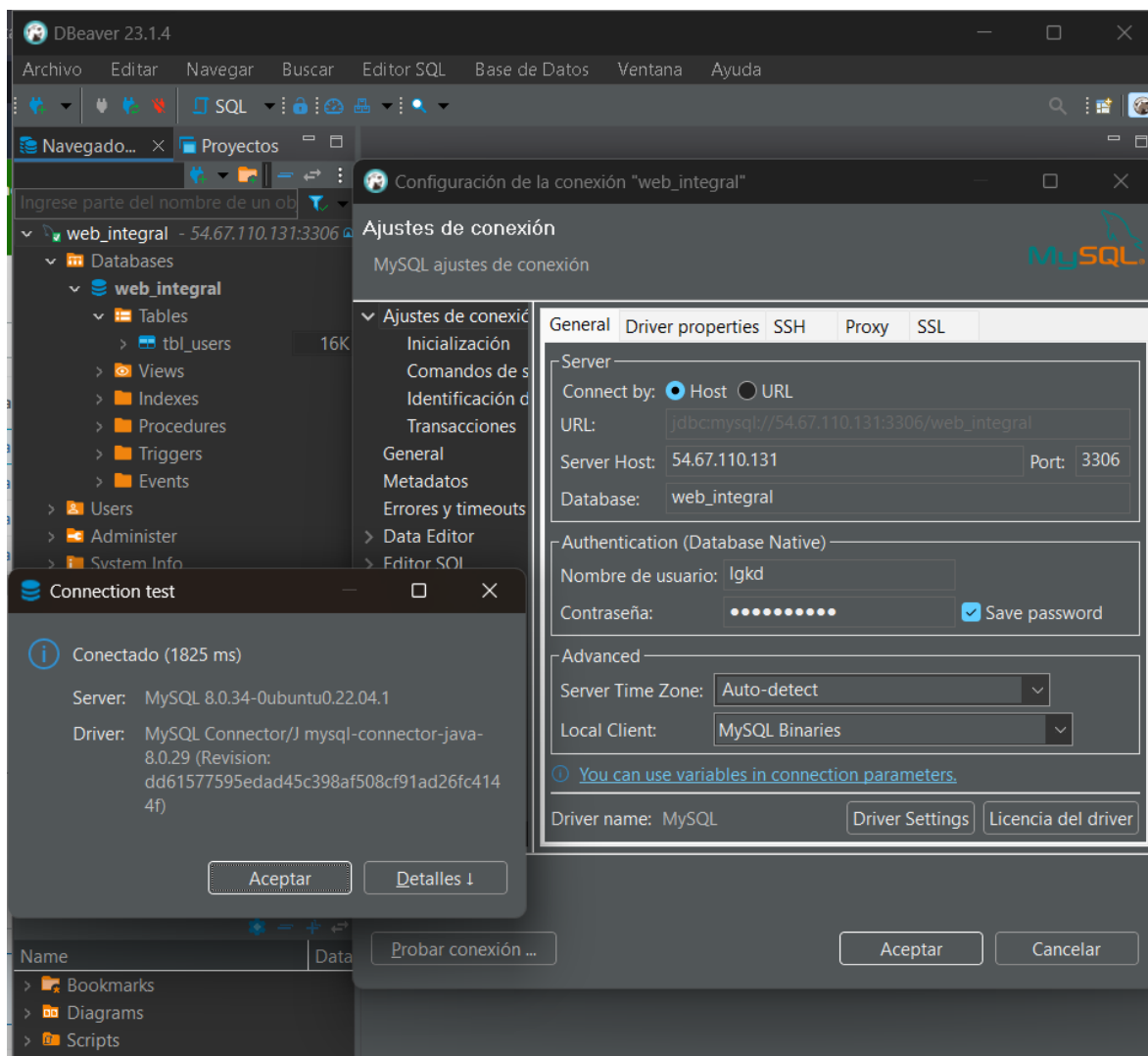
mysql> GRANT ALL PRIVILEGES ON * TO 'lgkd'@'%';
ERROR 1046 (3D000): No database selected
mysql> GRANT ALL PRIVILEGES ON web_integral.* TO 'lgkd'@'%';
Query OK, 0 rows affected (0.01 sec)

mysql> exit

```

Instalar [DBeaver](#) para acceder a la BD

pasar los parámetros de conexión



# Creando contenedor Docker-Angular

1. Una vez estando en carpeta de aplicación recién creada en angular, crear un archivo **DockerFile** en la raíz del proyecto. [referencia](#)

**sudo nano Dockerfile**

pegar el siguiente código para crear la imagen

Ingresa las siguientes instrucciones

```
# Etapa 1: Compila y construye codebase de Angular
# Utilizar la última imagen oficial de Node
FROM node:latest as build

# Establecer el directorio de trabajo
WORKDIR /usr/local/app

# Agregar el código fuente al directorio app
COPY ./ /usr/local/app/

# Instalar las dependencias
RUN npm install

# Generar y correr la aplicación
RUN npm run build

# Etapa 2: Servidor Web with nginx

# Uso oficial nginx image as the base image
# Utilizar la imagen oficial nginx como imagen base
FROM nginx:latest

# Copy the build output to replace the default nginx contents.
# Copiar la salida build a reemplazar los contenidos default NGINX.
COPY --from=build /usr/local/app/dist/test
/usr/share/nginx/html

# Exponer puerto 80
EXPOSE 80
```

**Guardar**      Ctrl + S

**Salir**        Ctrl + X

Ejecutando el contenedor Docker

Para una mayor referencia VÉASE [AQUÍ](#)

Construir la imagen

**sudo docker build -t test:1.0.1 .**

```
ubuntu@ip-172-31-6-75: ~/angular/ejemplo
[+] Building 106.7s (10/12)
=> [stage-1 1/2] FROM docker.io/library/nginx:latest@sha256:67f9a4f10d147a6e04629340e6493c9703300ca23a2f7f3aa561
=> resolve docker.io/library/nginx:latest@sha256:67f9a4f10d147a6e04629340e6493c9703300ca23a2f7f3aa561
=> sha256:67f9a4f10d147a6e04629340e6493c9703300ca23a2f7f3aa561 1.86kB / 1.86kB
=> sha256:73e957793f1266530db0a9c1fda3f87c1e59943f4c13eb340bb8521c6041d7 1.78kB / 1.78kB
=> sha256:89da1fb6dc964dd35c3f41b7b93ffc35eaf20bc61f2e1335fea710a18424287 8.15kB / 8.15kB
=> sha256:648e0aadf75ac2ef63c5390adc6dc14fde37a5ad88c2870ea604df0a9c0eb4e5 29.12MB / 29.12MB
=> sha256:262696647b70a57f5f7dbf97a91091e7b51c1d2537dff72a2bdf908a06549197 41.46MB / 41.46MB
=> extracting sha256:648e0aadf75ac2ef63c5390adc6dc14fde37a5ad88c2870ea604df0a9c0eb4e5
=> sha256:55ac49bd649c325395133ae4f3640a07e28d9a25c4a56eb8ac3df91e1c639f51 957B / 957B
=> sha256:8015f365966bfa259003c319a44df5bb9290d279ca775b4f249f291e323b38f1 1.21kB / 1.21kB
=> sha256:e66d0270d23f3038e0e8c94ee9244950fbfdb582476f61736b3c28a6ed159921 625B / 625B
=> sha256:cbf42f5a00d268edb1684b8eb9039543669fc5f5d0aa801a01d3460048b7d487 366B / 366B
=> sha256:4cadff8bc2aa83b23dd9e02a590174a84691f954eff4346888724c073ddc2524 1.40kB / 1.40kB
=> extracting sha256:262696647b70a57f5f7dbf97a91091e7b51c1d2537dff72a2bdf908a06549197
=> extracting sha256:e66d0270d23f3038e0e8c94ee9244950fbfdb582476f61736b3c28a6ed159921
=> extracting sha256:55ac49bd649c325395133ae4f3640a07e28d9a25c4a56eb8ac3df91e1c639f51
=> extracting sha256:cbf42f5a00d268edb1684b8eb9039543669fc5f5d0aa801a01d3460048b7d487
=> extracting sha256:8015f365966bfa259003c319a44df5bb9290d279ca775b4f249f291e323b38f1
=> extracting sha256:4cadff8bc2aa83b23dd9e02a590174a84691f954eff4346888724c073ddc2524
=> [internal] load build context
=> transferring context: 412.81MB
=> [build 2/5] WORKDIR /usr/local/app
=> [build 3/5] COPY ./ /usr/local/app/
=> [build 4/5] RUN npm install
=> [build 5/5] RUN npm run build
=> # > ejemplo@0.0.0 build
=> # > ng build
=> # - Generating browser application bundles (phase: setup)...
```

Verificar que exista la imagen mediante el comando, el cual lista a todas las imágenes.

**sudo docker image ls**

Ejecutar la imagen y exponerla en el puerto 8080

**sudo docker run -d -p 8080:80 test:1.0.1**

Mostrar un listado de los contenedores en ejecución

**sudo docker ps**

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
12834dafbbde	test:1.0.1	"/docker-entrypoint..."	16 minutes ago	Up 16 minutes	0.0.0.0:8080->80/tcp, :::8080->80/tcp	intelligent_merkle

Verificar salida ingresando la URL <http://localhost:8080/>

