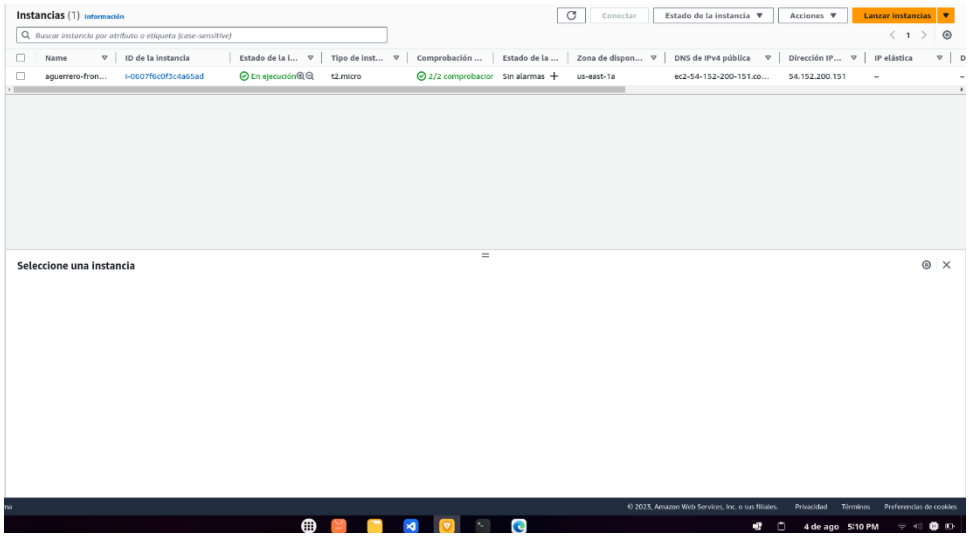
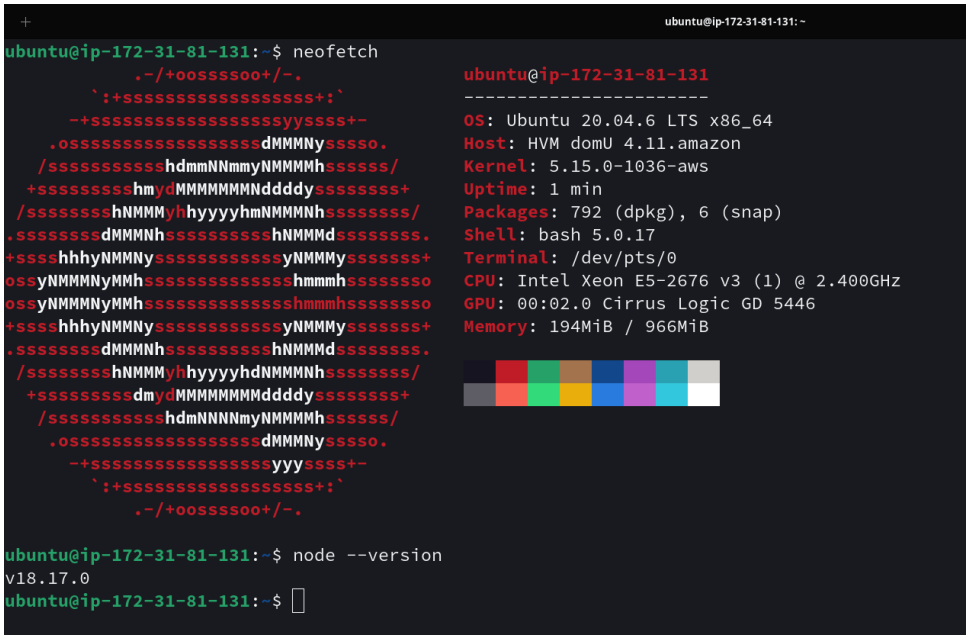
Integrantes:

Alejandro Rafael Guerrero Lozano

David Enrique López Juárez.

Creando instancia.





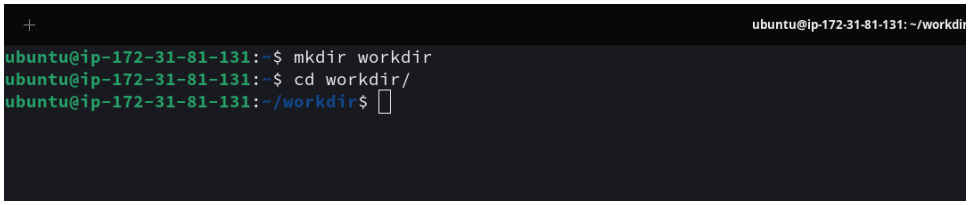
Práctica de la Lección

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

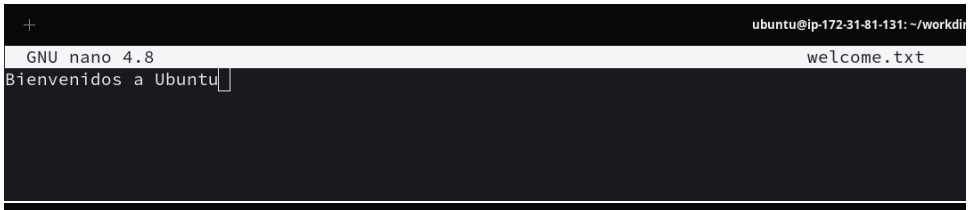
- Listar el contenido del directorio actual

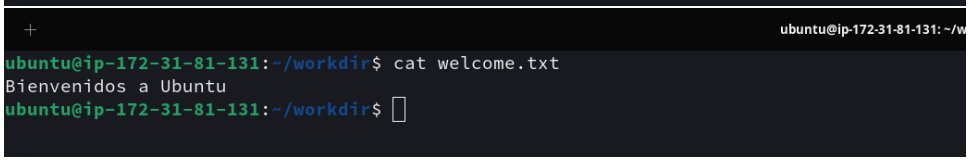
- Ejecutar comando qué indica la ruta de directorio actual.

- Crear un directorio llamado workdir y cambiarse al directorio.

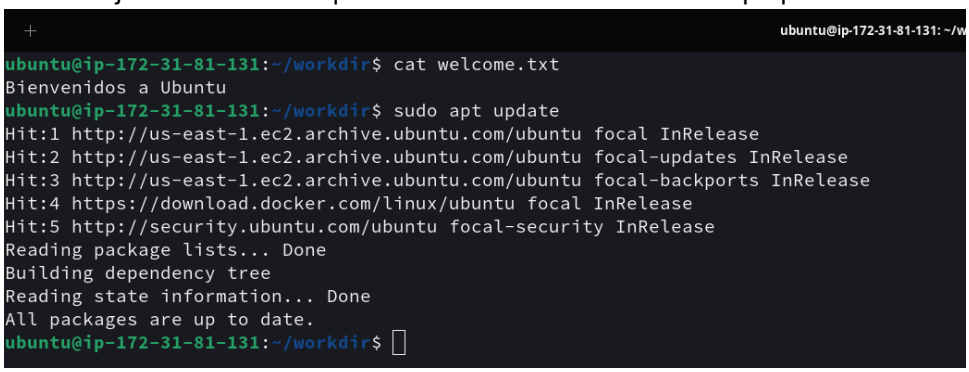


- Crear un archivo llamado **welcome.txt** qué tenga como contenido “Bienvenidos a Ubuntu”; despliega el contenido del archivo a través de línea de comandos.





- Ejecutar comando qué instala la última versión de los paquetes

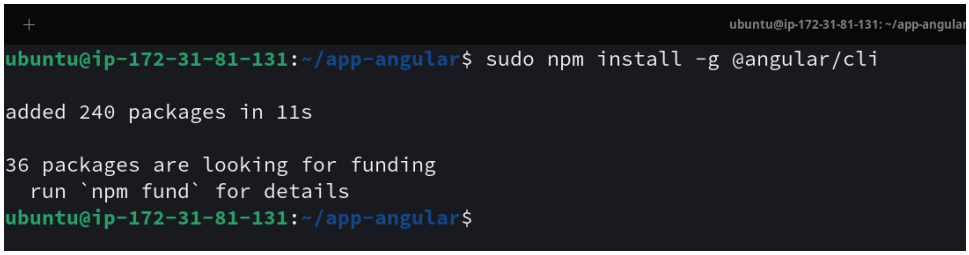


Creando aplicación de angular.

1. Crear un directorio de trabajo.

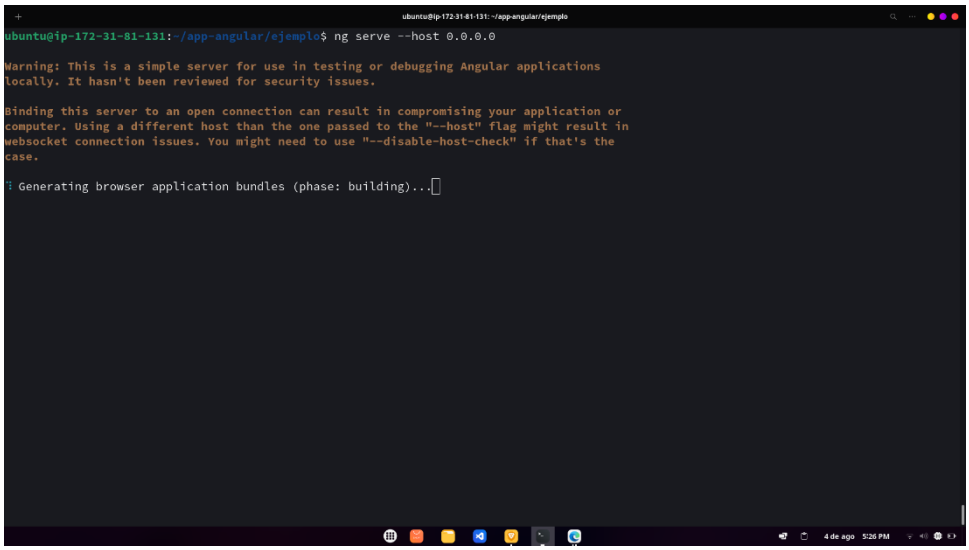
2. Instalar Angular CLI

npm install -g @angular/CLI@latest

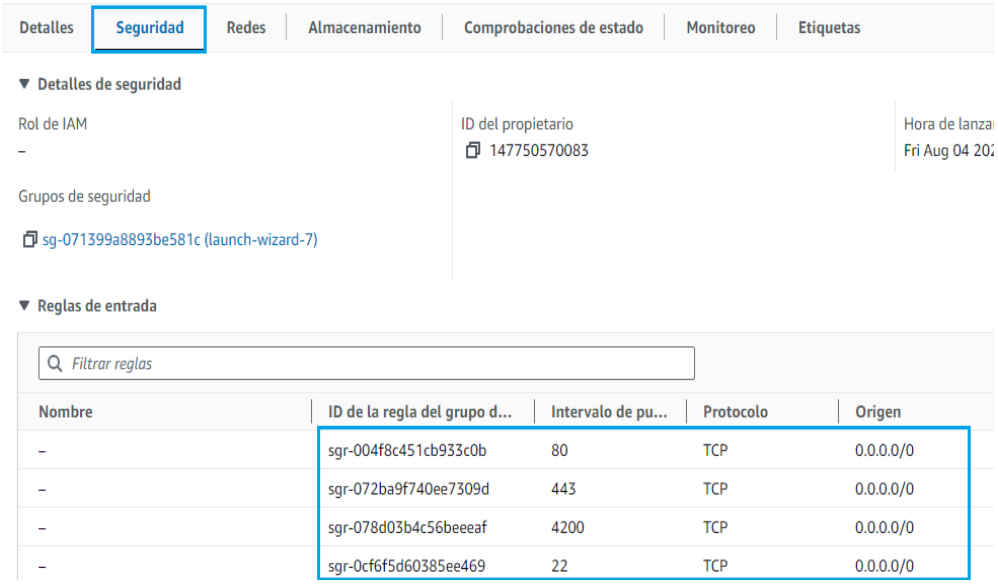


3. Crear una nueva aplicación Angular

ng new ejemplo --strict --style=scss --routing --skip-tests



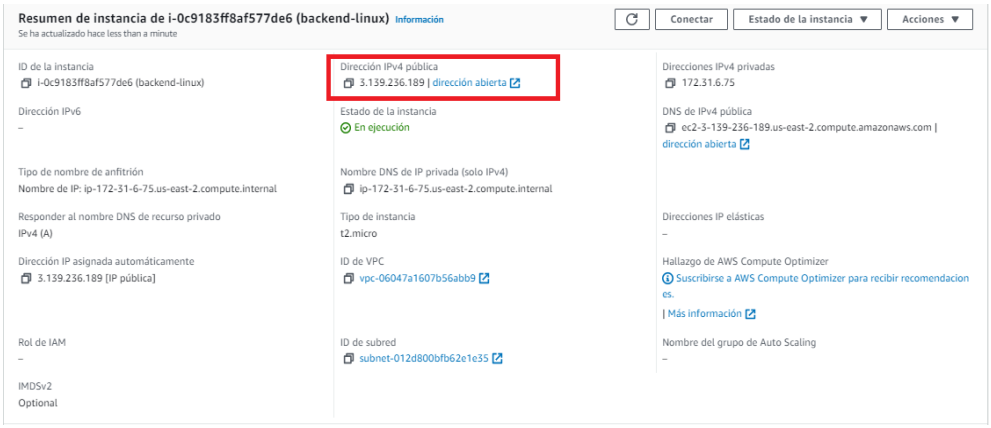
4. Configurar la instancia EC2, seleccionar pestaña Seguridad y agregar las nuevas reglas



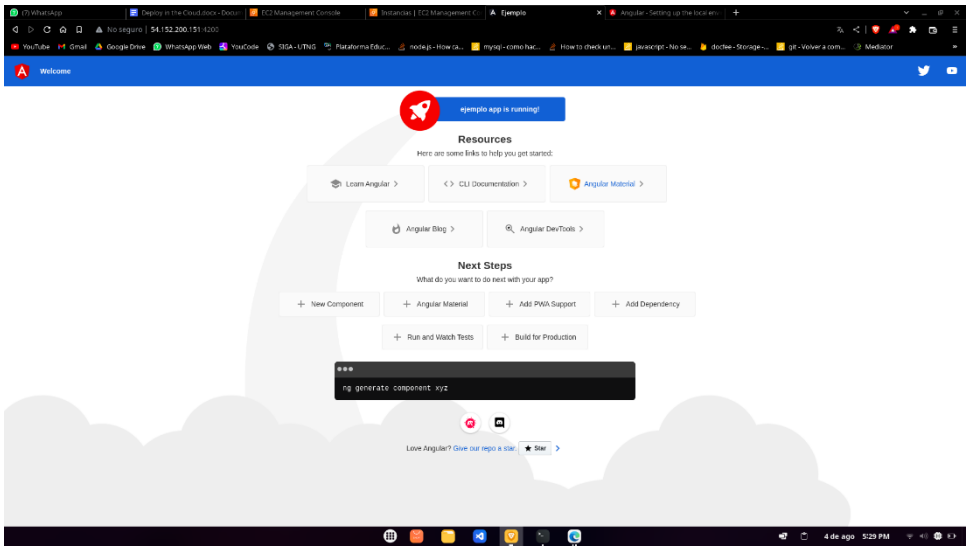
5. Iniciar el servidor para conocer ejecución de la herramienta

```
ng serve --host 0.0.0.0
```

6. 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



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


Welcome
ejemplo app is running!
Resources
Here are some links to help you get started:
Learn Angular
CLI Documentation
Angular Material
Angular Blog
Angular DevTools
Next Steps
What do you want to do next with your app?
New Component
Angular Material
Add PWA Support
Add Dependency
Run and Watch Tests
Build for Production
ng generate component xyz
Love Angular? Give our repo a star.
Star
4 de ago 5:29 PM

Instalar Nginx

Referencia

Instalar las fuentes de configuración

sudo apt update

```
ubuntu@ip-172-31-81-131:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu focal InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
ubuntu@ip-172-31-81-131:~$ sudo apt install openssl
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssl is already the newest version (1.1.1f-1ubuntu2.19).
openssl set to manually installed.
The following packages were automatically installed and are no longer required:
  gyp javascript-common libauthen-sasl-perl libc-ares2 libdata-dump-perl libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2 libdrm-radeon1 libencode-locale-perl
  libfile-basedir-perl libfile-desktopentry-perl libfile-listing-perl libfile-mimeinfo-perl libfont-afm-perl libfontenc1 libgl1 libgl1-mesa-dri
  libglapi-mesa libglvnd0 libglx-mesa0 libglx0 libhtml-form-perl libhtml-format-perl libhtml-parser-perl libhtml-tagset-perl libhtml-tree-perl
  libhttp-cookies-perl libhttp-daemon-perl libhttp-date-perl libhttp-message-perl libhttp-negotiate-perl libice6 libio-html-perl libio-socket-ssl-perl
  libio-stringy-perl libipc-system-simple-perl libjs-inherits libjs-is-typedarray libjs-psl libjs-typedarray-to-buffer libllvm12 liblwp-mediatypes-perl
  liblwp-protocol-https-perl libmailtools-perl libnet-dbus-perl libnet-http-perl libnet-smtp-ssl-perl libnet-ssleay-perl libnode-dev libnode64 libpciaccess0
  libpython2-stdlib libpython2.7-minimal libpython2.7-stdlib libsensors-config libsensors5 libsm6 libssl-dev libtie-ixhash-perl libtimedate-perl
  libtry-tiny-perl liburi-perl libuv1-dev libvulkan1 libwayland-client0 libwww-perl libwww-robotrules-perl libx11-protocol-perl libx11-xcb1 libxaw7
  libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-randr0 libxcb-shape0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libxcomposite1 libxcursor1
  libxfixes3 libxft2 libxi6 libxinerama1 libxkbfile1 libxml-parser-perl libxml-twig-perl libxml-xpathengine-perl libxmu6 libxpm4 libxrandr2 libxrender1
  libxshmfence1 libxt6 libxtst6 libxv1 libxxf86dga1 libxxf86vm1 mesa-vulkan-drivers nodejs-doc perl-openssl-defaults python-pkg-resources python2
  python2-minimal python2.7 python2.7-minimal x11-common x11-utils x11-xserver-utils xdg-utils
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-81-131:~$
```

Instalar el servidor Web Nginx con el comando

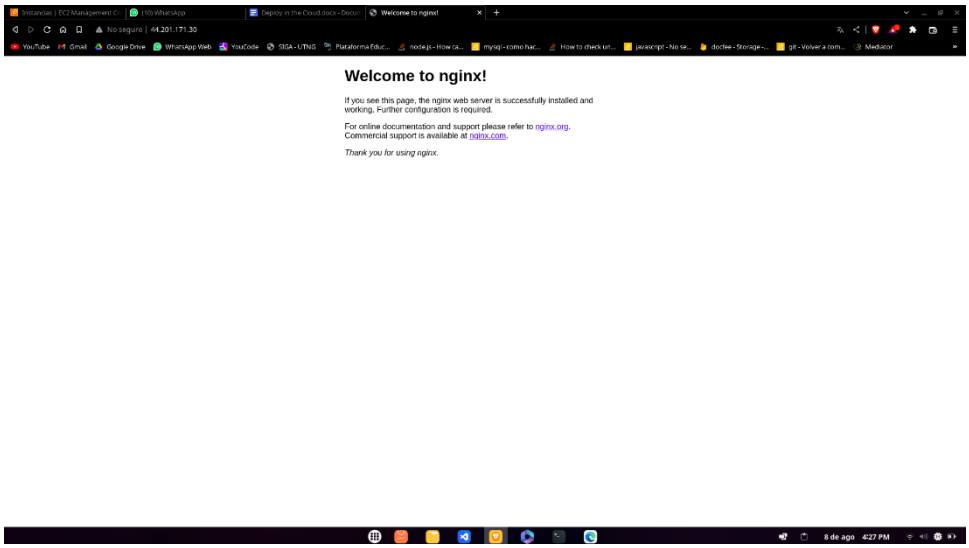
sudo apt install nginx

```
E: The update command takes no arguments
ubuntu@ip-172-31-81-131:~$ sudo apt install nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gyp javascript-common libauthen-sasl-perl libc-ares2 libdata-dump-perl libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2 libdrm-radeon1 libencode-locale-perl
  libfile-basedir-perl libfile-desktopentry-perl libfile-listing-perl libfile-mimeinfo-perl libfont-afm-perl libfontenc1 libgl1 libgl1-mesa-dri
  libglapi-mesa libglvnd0 libglx-mesa0 libglx0 libhtml-form-perl libhtml-format-perl libhtml-parser-perl libhtml-tagset-perl libhtml-tree-perl
  libhttp-cookies-perl libhttp-daemon-perl libhttp-date-perl libhttp-message-perl libhttp-negotiate-perl libice6 libio-html-perl libio-socket-ssl-perl
  libio-stringy-perl libipc-system-simple-perl libjs-inherits libjs-is-typedarray libjs-psl libjs-typedarray-to-buffer libllvm12 liblwp-mediatypes-perl
  liblwp-protocol-https-perl libmailtools-perl libnet-dbus-perl libnet-http-perl libnet-smtp-ssl-perl libnet-ssleay-perl libnode-dev libnode64 libpciaccess0
  libpython2-stdlib libpython2.7-minimal libpython2.7-stdlib libsensors-config libsensors5 libsm6 libssl-dev libtie-ixhash-perl libtimedate-perl
  libtry-tiny-perl liburi-perl libuv1-dev libvulkan1 libwayland-client0 libwww-perl libwww-robotrules-perl libx11-protocol-perl libx11-xcb1 libxaw7
  libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-randr0 libxcb-shape0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libxcomposite1 libxcursor1
  libxfixes3 libxft2 libxi6 libxinerama1 libxkbfile1 libxml-parser-perl libxml-twig-perl libxml-xpathengine-perl libxmu6 libxrandr2 libxrender1
  libxshmfence1 libxt6 libxtst6 libxv1 libxxf86dga1 libxxf86vm1 mesa-vulkan-drivers nodejs-doc perl-openssl-defaults python-pkg-resources python2
  python2-minimal python2.7 python2.7-minimal x11-common x11-utils x11-xserver-utils xdg-utils
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libgd3 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream nginx-common nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  libgd3 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream nginx nginx-common nginx-core
0 upgraded, 8 newly installed, 0 to remove and 0 not upgraded.
Need to get 723 kB of archives.
After this operation, 2604 kB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libgd3 amd64 2.2.5-5.2ubuntu2.1 [118 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 nginx-common all 1.18.0-0ubuntu1.4 [37.7 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libnginx-mod-http-image-filter amd64 1.18.0-0ubuntu1.4 [14.8 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libnginx-mod-http-xslt-filter amd64 1.18.0-0ubuntu1.4 [13.0 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libnginx-mod-mail amd64 1.18.0-0ubuntu1.4 [42.9 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libnginx-mod-stream amd64 1.18.0-0ubuntu1.4 [67.4 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 nginx-core amd64 1.18.0-0ubuntu1.4 [425 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 nginx all 1.18.0-0ubuntu1.4 [3620 B]
Fetched 723 kB in 0s (15.8 MB/s)
Preconfiguring packages ...
Selecting previously unselected package libgd3:amd64.
```

Verificar el estado del servidor web

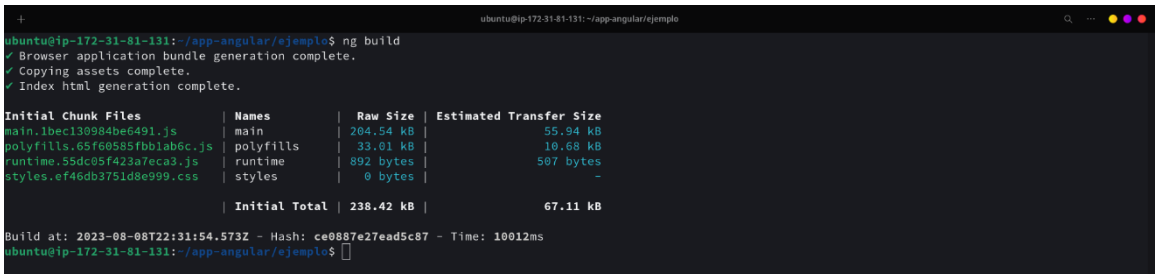
systemctl status nginx

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



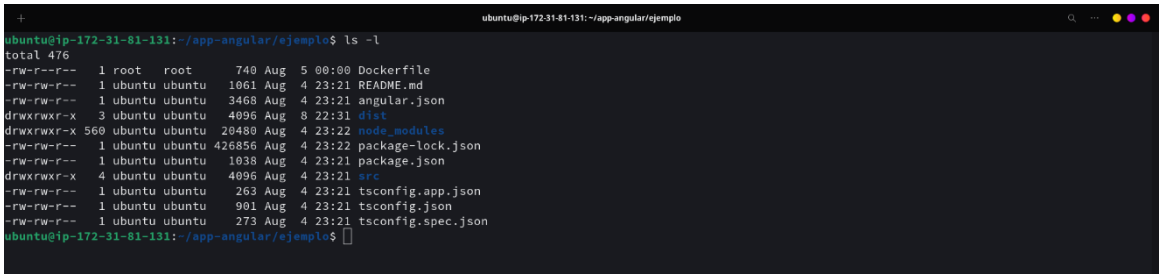
Ahora vamos a construir el proyecto test, ingresar el comando

ng build



Ingresar el comando para el listado de carpetas y archivos

ls –l



Para conocer la ruta de directorio actual ingresar comando

pwd

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

/home/ubuntu/workdir/test/dist/test

Modificar el archivo de configuración de Nginx

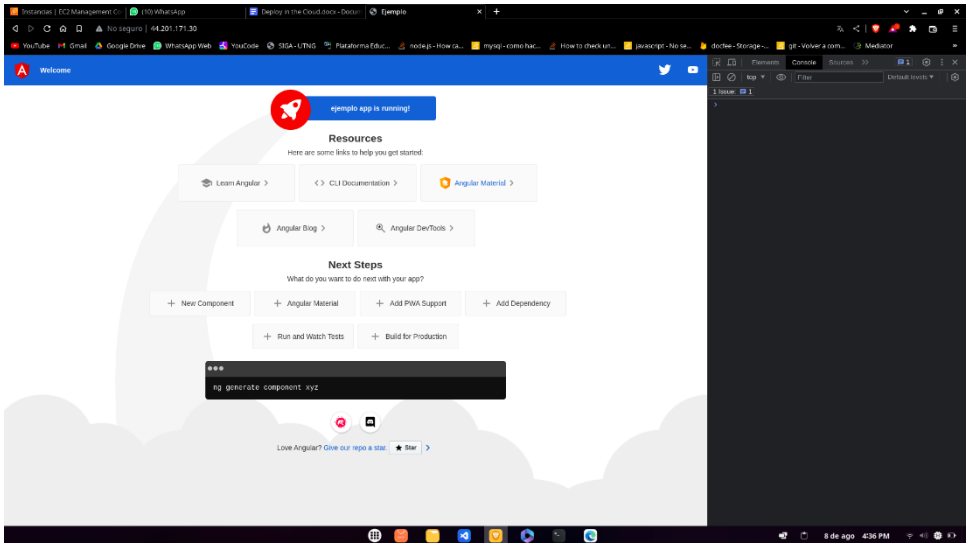
sudo nano /etc/nginx/sites-available/default

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

Reiniciar servidor seb con comando

sudo systemctl restart nginx

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



Configuración HTTPS

Crear una carpeta llamada ssl

mkdir ssl

Instalar openssl

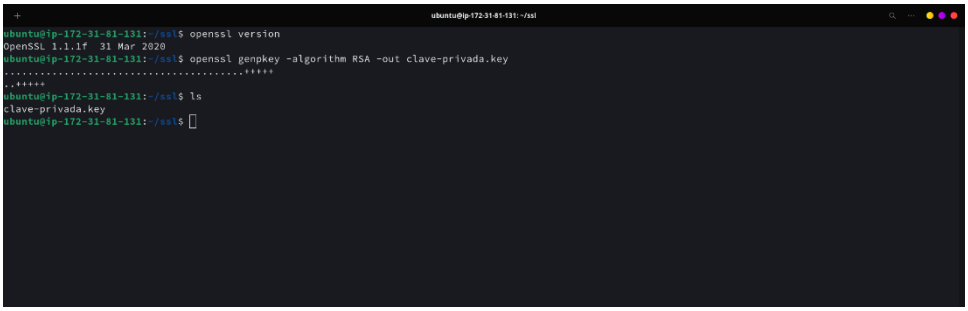
sudo apt install openssl

Conocer versión de openssl

openssl version

Generar una clave privada llamada clave-privada.key

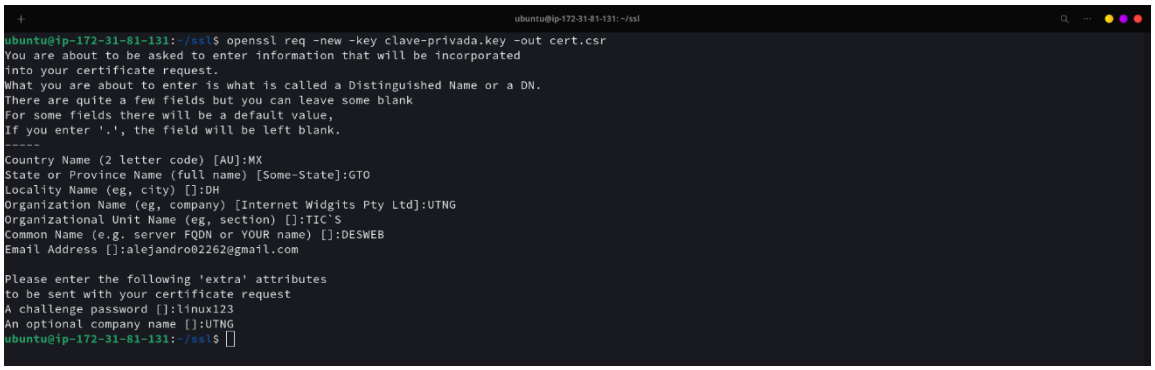
openssl genpkey -algorithm RSA -out clave-privada.key



Escribir comando para verificar qué se creo

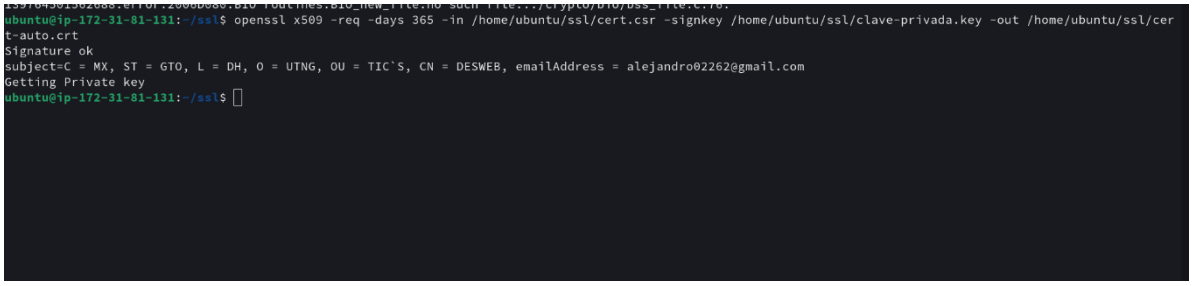
Generar un certificado en base a la clave privada recién creada, responde a las interrogantes como en imagen de ejemplo

openssl req -new -key clave-privada.key -out cert.csr



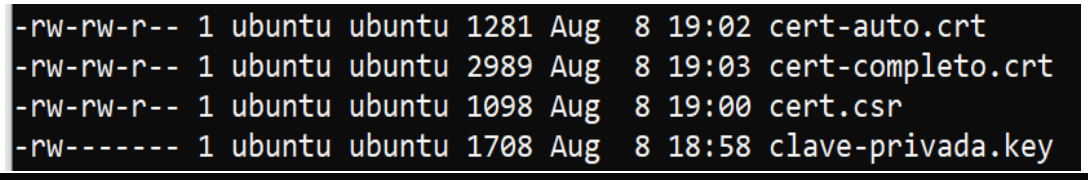
Generar nuevamente certificado firmado

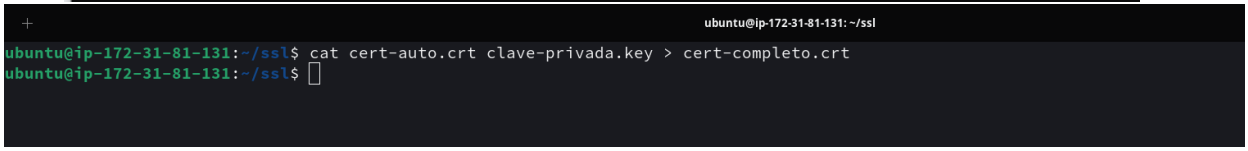
openssl x509 -req -days 365 -in /home/ubuntu/workdir/ssl/cert.csr -signkey /home/ubuntu/workdir/ssl/clave-privada.key -out /home/ubuntu/workdir/ssl/cert-auto.crt



Unir ambos certificados para mayor seguridad

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





Modificar archivo en servidor web Nginx

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

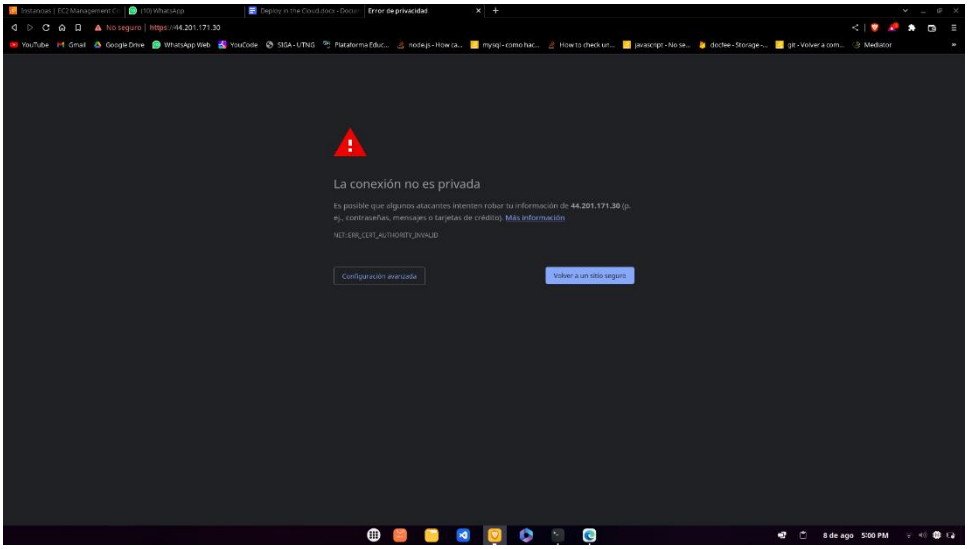
CTRL + S Guardar

CTRL + X Salir

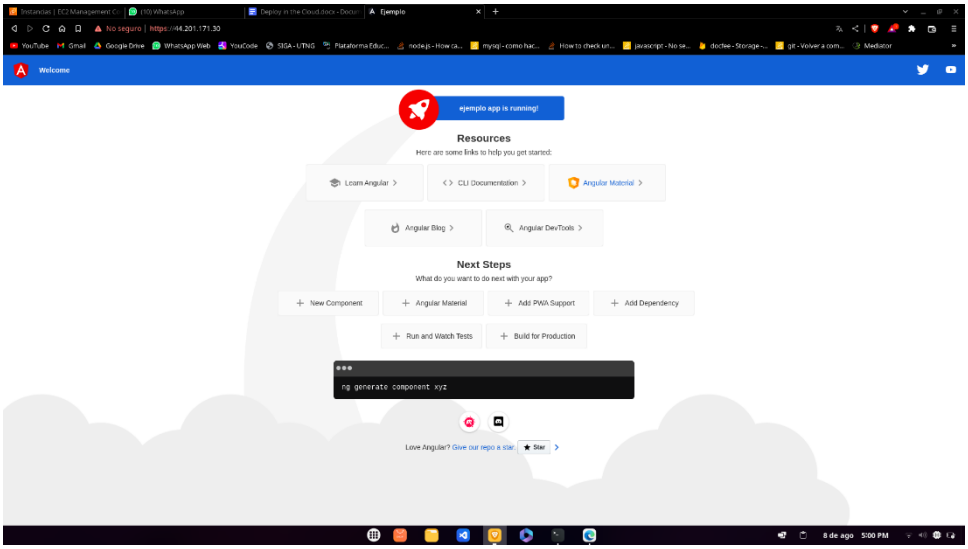
Reiniciar nuevamente el servidor

sudo systemctl restart nginx

Abrir navegador web con la direccion: https://44.201.171.30/

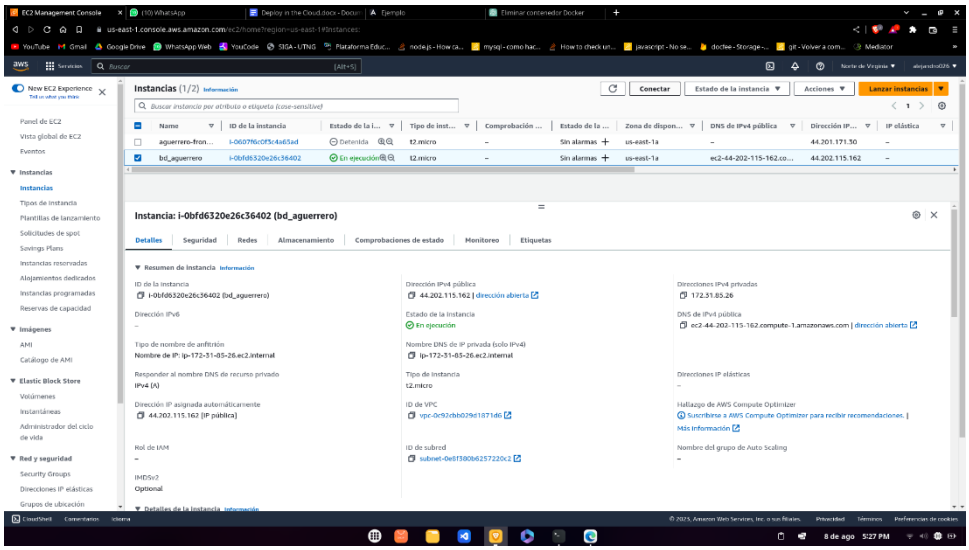


Seleccionar configuración avanzada | Continuar a 18.233.156.132 (no seguro)

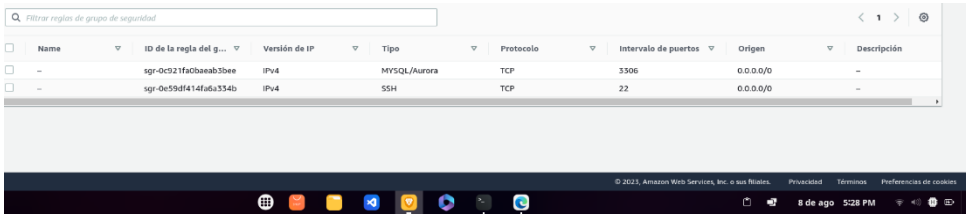


Instalar MySQL

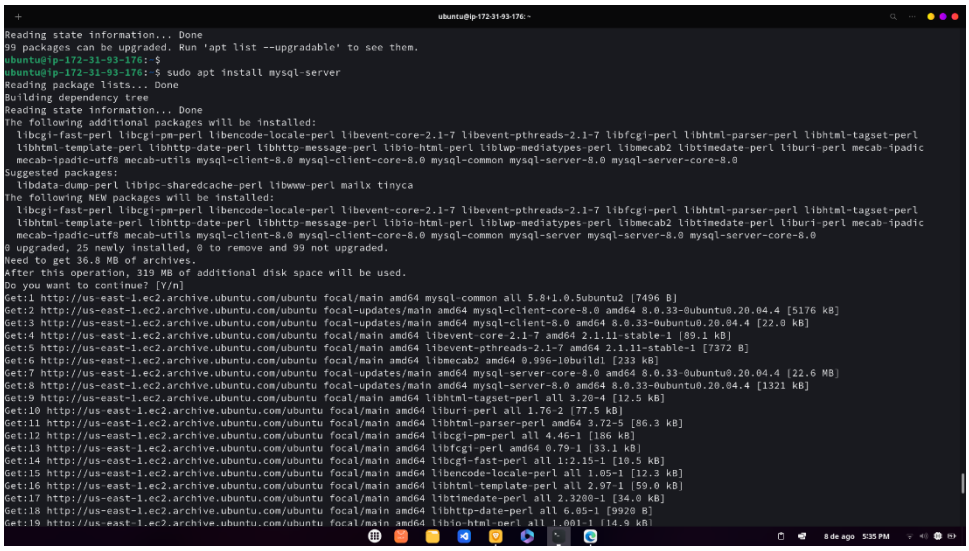
Crear una nueva instancia donde se deposite la base de datos.



Abrir el puerto 3306 de EC2



Instalar la base de datos MySQL



```
ubuntu@ip-172-31-93-176:~$ sudo systemctl start mysql.service
ubuntu@ip-172-31-93-176:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
     Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2023-08-08 23:35:57 UTC; 26s ago
   Main PID: 2552 (mysqld)
     Status: "Server is operational"
      Tasks: 38 (limit: 1141)
     Memory: 355.5M
     CGroup: /system.slice/mysql.service
             └─2552 /usr/sbin/mysqld

Aug 08 23:35:56 ip-172-31-93-176 systemd[1]: Starting MySQL Community Server...
Aug 08 23:35:57 ip-172-31-93-176 systemd[1]: Started MySQL Community Server.
ubuntu@ip-172-31-93-176:~$
```

```
ubuntu@ip-172-31-93-176:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: y

There are three levels of password validation policy:

LOW    Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary                  file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0

Skipping password set for root as authentication with auth_socket is used by default.
If you would like to use password authentication instead, this can be done with the "ALTER_USER" command.
See https://dev.mysql.com/doc/refman/8.0/en/alter-user.html#alter-user-password-management for more information.

By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : Y
Success.


Normally, root should only be allowed to connect from
```

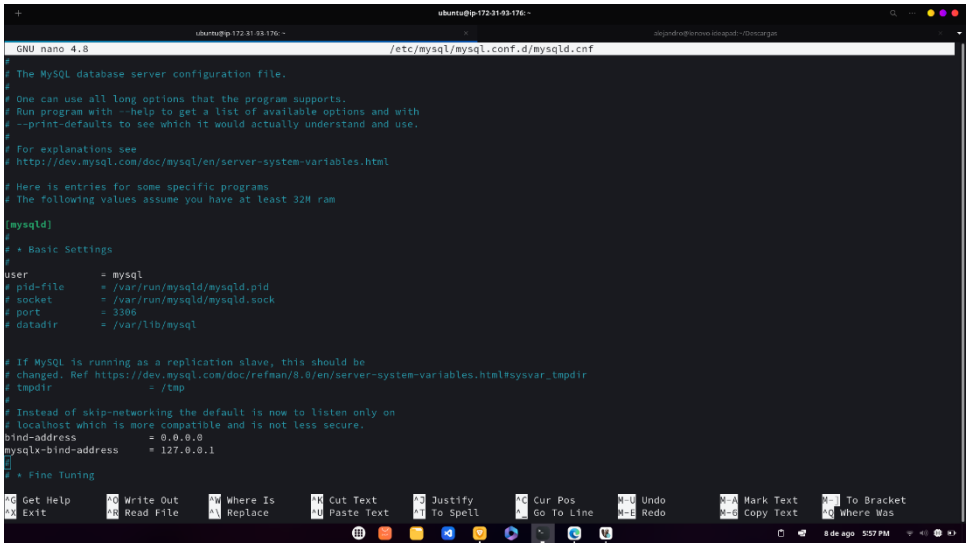
Crear la base de datos web_integral;

```
ubuntu@ip-172-31-93-176:~$ mysql -u aguerrero -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.33-0ubuntu0.20.04.4 (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 (0.05 sec)

mysql> use web_integral;
Database changed
mysql>
```

Creando tabla de usuario.

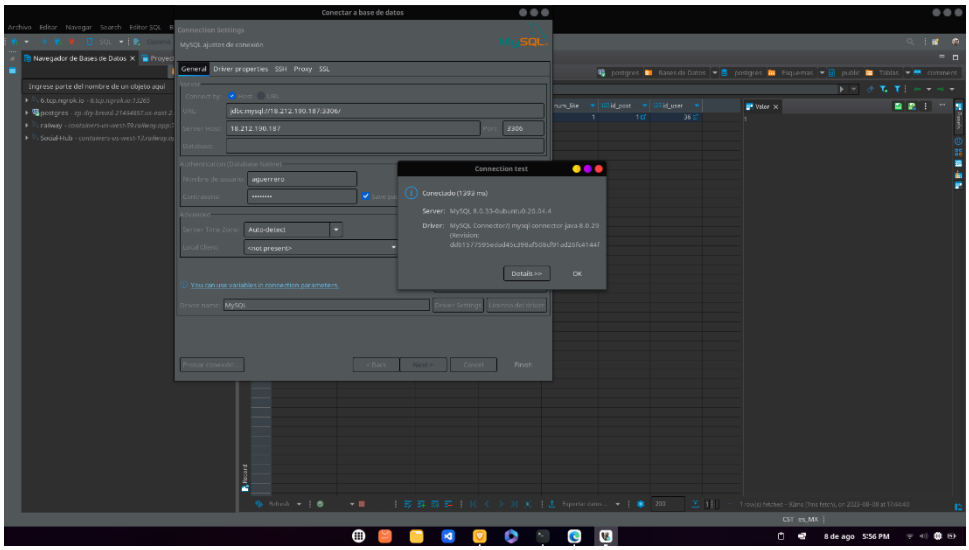
```
mysql> CREATE TABLE tbl_usuario (
    ->     Username VARCHAR(30) PRIMARY KEY,
    ->     Password VARCHAR(250) NOT NULL,
    ->     Role VARCHAR(20) NOT NULL
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql>
```

[Configurar base de datos para acceso remoto]



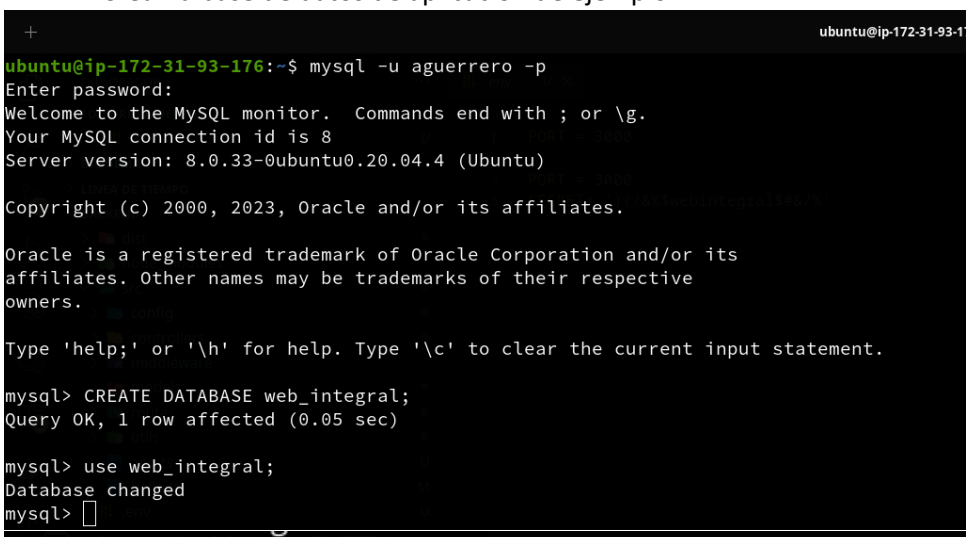
Instalar [DBeaver]para acceder a la BD

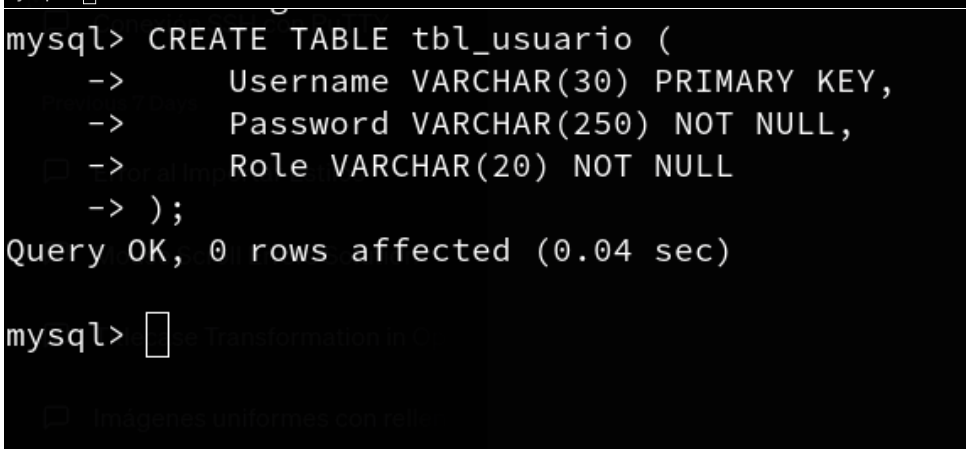
pasar los parámetros de conexión



Práctica de la lección

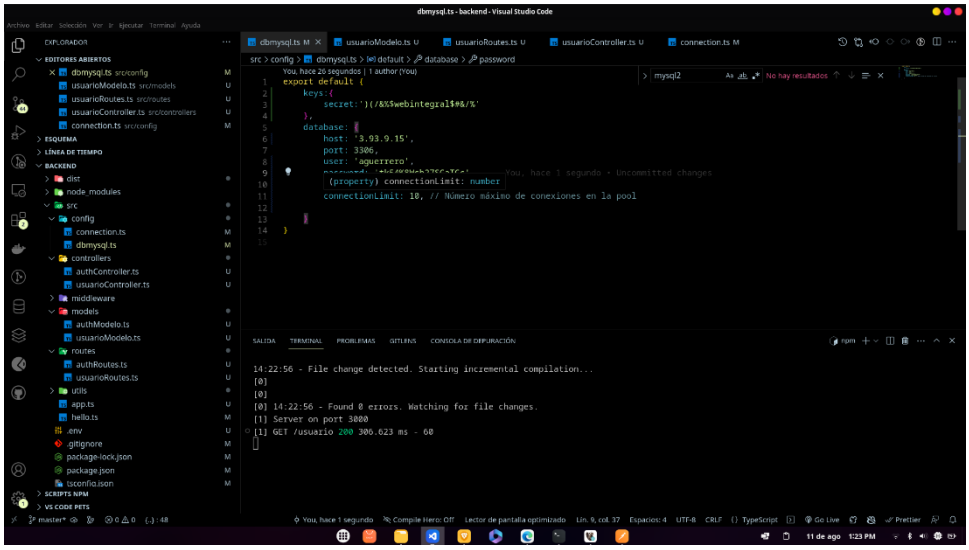
- Crear la base de datos de aplicación de ejemplo.



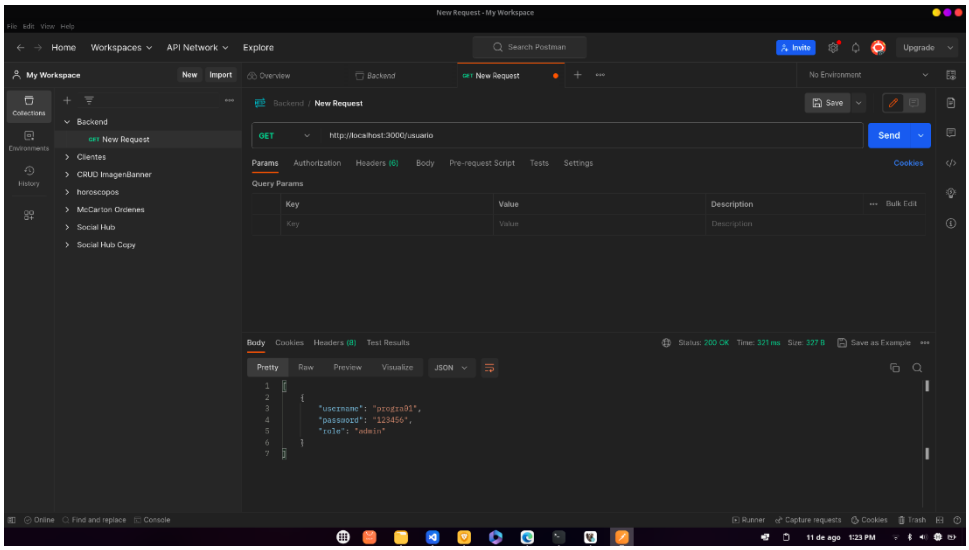


- Montar el backend

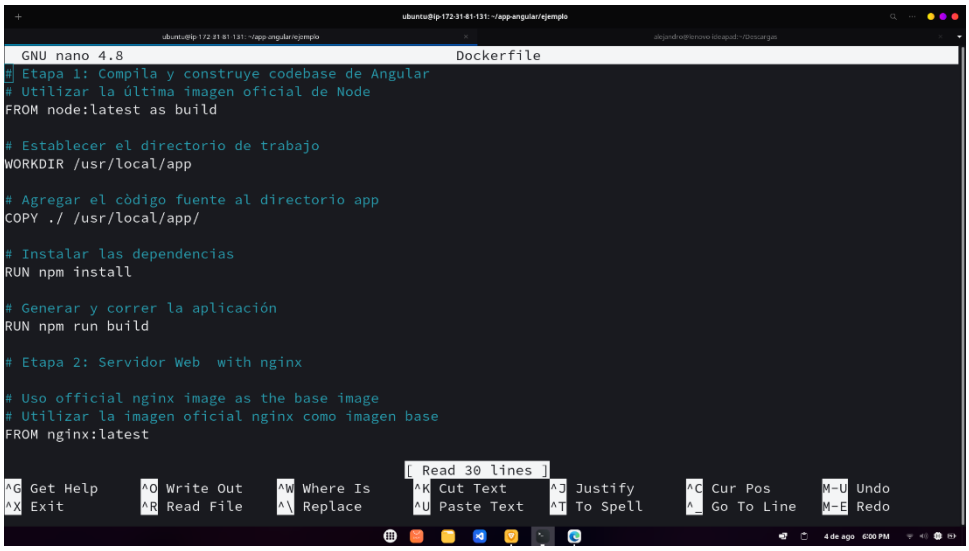
Configurando backend

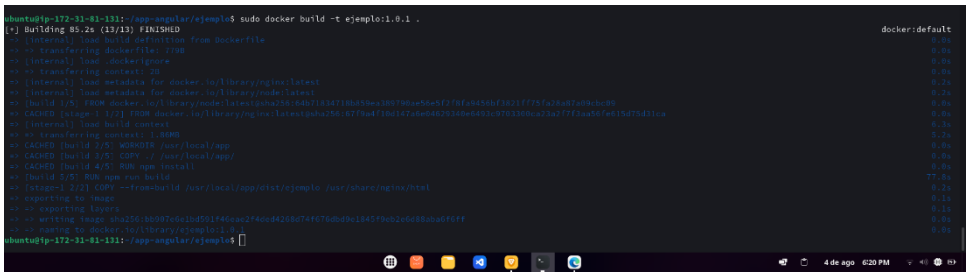


- Verificar funcionalidad con Postman

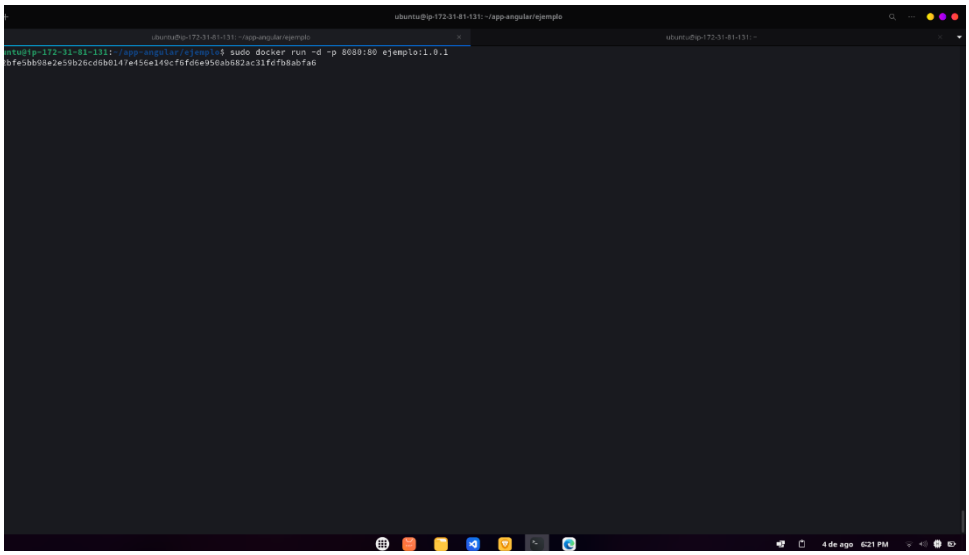

New Request - My Workspace
Home
Workspaces
API Network
Explore
Search Postman
Invite
Upgrade
My Workspace
New
Import
Overview
Backend
New Request
No Environment
Collections
Environments
History
Backend
New Request
Clientes
CRUD ImagenBanner
horoscopos
McCarton Ordenes
Social Hub
Social Hub Copy
Backend / New Request
Save
GET
http://localhost:3000/usuario
Send
Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
Body
Cookies
Headers (8)
Test Results
Status: 200 OK
Time: 321 ms
Size: 327 B
Save as Example
Pretty
Raw
Preview
Visualize
JSON
{
"username": "progra91",
"password": "123456",
"role": "admin"
}
Online
Find and replace
Console
Runner
Capture requests
Cookies
Trash
11 de ago 1:23 PM

Creando contenedor Docker-Angular

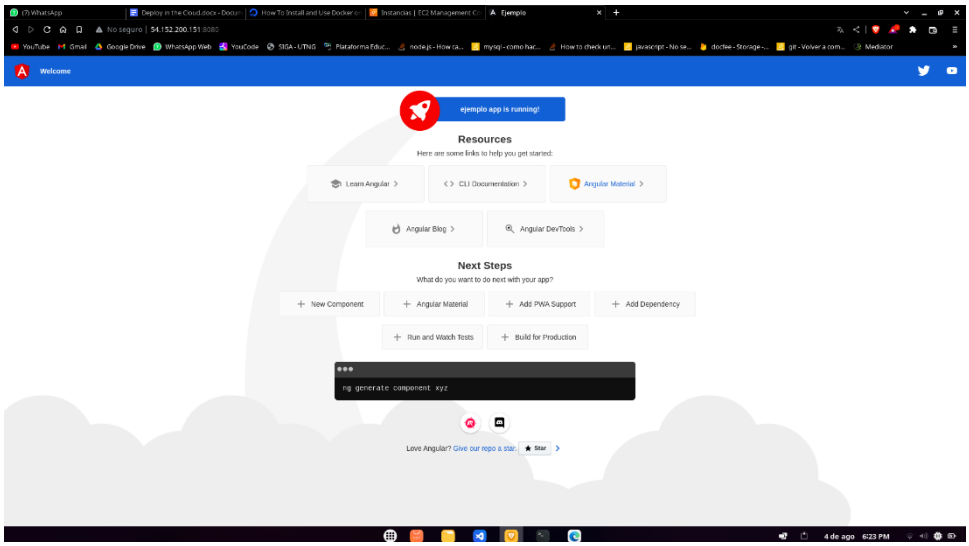




Ejecutando imagen.

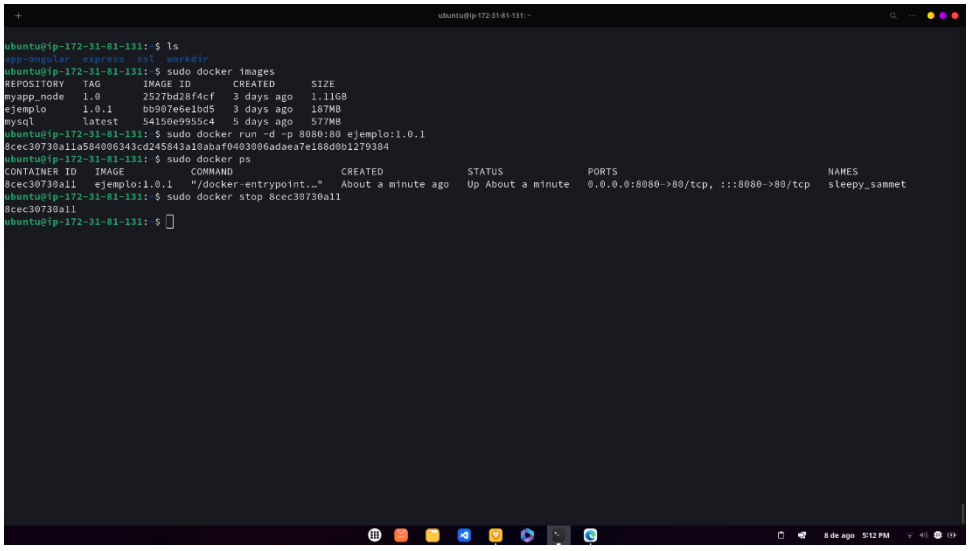


Probando en la nube.

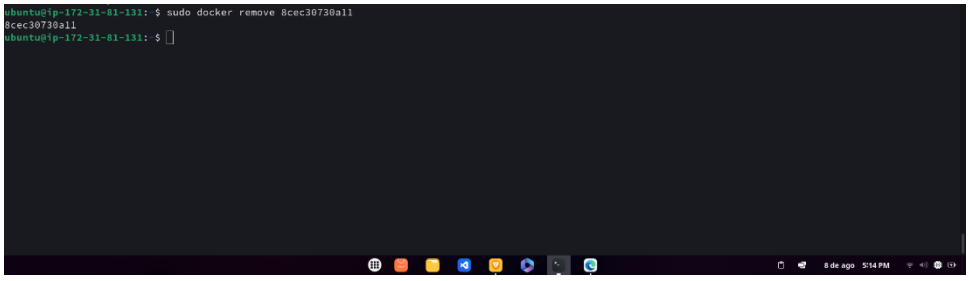


Práctica de la sesión

- Para contenedor recién ejecutado.



- Eliminar el contenedor recien ejecutado



- Eliminar la imagen

```
ubuntu@ip-172-31-81-131: ~
ubuntu@ip-172-31-81-131:~$ sudo docker images
REPOSITORY   TAG       IMAGE ID       CREATED      SIZE
myapp_node   1.0       2527bd28f4cf   3 days ago   1.11GB
ejemplo      1.0.1     bb907e6e1bd5   3 days ago   187MB
mysql        latest    54150e9955c4   5 days ago   577MB
ubuntu@ip-172-31-81-131:~$ sudo docker rmi bb907e6e1bd5
Untagged: ejemplo:1.0.1
Deleted: sha256:bb907e6e1bd591f46eae2f4ded4268d74f676dbd9e1845f9eb2e6d88aba6f6ff
ubuntu@ip-172-31-81-131:~$
```

- Listar todas las imágenes

```
ubuntu@ip-172-31-81-131:~$ sudo docker images
REPOSITORY   TAG       IMAGE ID       CREATED      SIZE
myapp_node   1.0       2527bd28f4cf   3 days ago   1.11GB
mysql        latest    54150e9955c4   5 days ago   577MB
ubuntu@ip-172-31-81-131:~$
```

Docker NodeJs

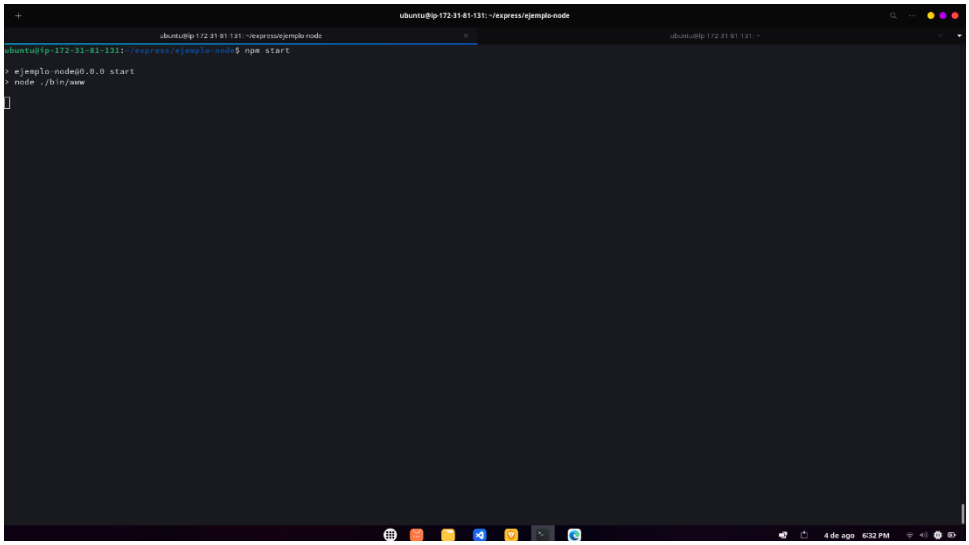
Ubicar un directorio de trabajo para alojar la aplicación.

Instalar express-generator, para mayor referencia véase AQUI

```
npx express-generator
```

Crear el proyecto

express ejemplo-node



Ingresar la URL http://localhost:3000

Parar el servidor Node

Ingresar Ctrl+C

Creando el contenedor Docker para Node

Crear un archivo **Dockerfile** en el directorio raíz del proyecto e ingresar las siguientes instrucciones

```
FROM node:latest as build
WORKDIR /usr/local/app
COPY package*.json /usr/local/app/
RUN npm install
COPY ./ /usr/local/app/
CMD [ "npm", "start" ]
```

```
ubuntu@ip-172-31-81-131: ~/express/ejemplo-node
  GNU nano 4.8
FROM node:latest as build

WORKDIR /usr/local/app

COPY package*.json /usr/local/app/

RUN npm install

COPY ./ /usr/local/app/

CMD [ "npm", "start" ]
```

Crear la imagen

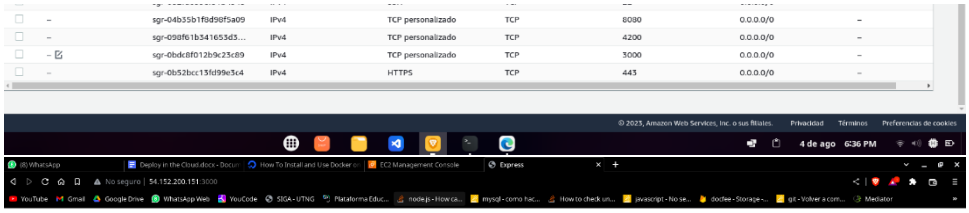
sudo docker build -t myapp_node:1.0 .

Ejecutar el contenedor

sudo docker run -p 3000:3000 myapp_node:1.0

```
ubuntu@ip-172-31-81-131:~/express/ejemplo-node$ sudo docker run -p 3000:3000 myapp_node:1.0

> ejemplo-node@0.0.0 start
> node ./bin/www
```

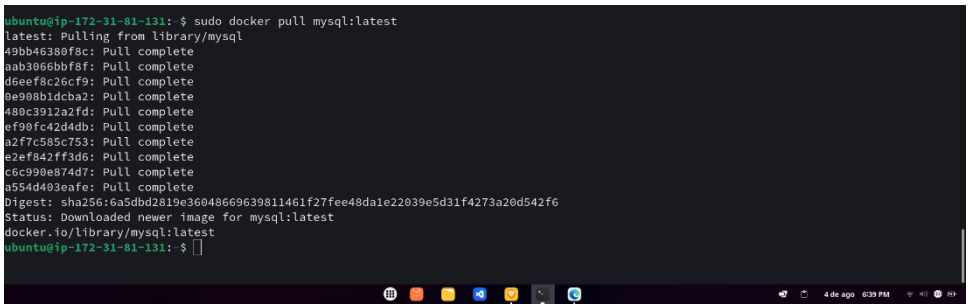
Agregando regla en aws



Docker MySQL

Extraer la imagen de mysql

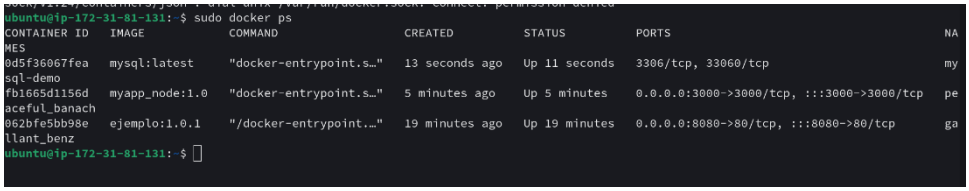
docker pull mysql:latest



Ejecutar el siguiente comando donde crearà automàticamente la imagen con una contraseña y el volumen para almacenar la informaciòn

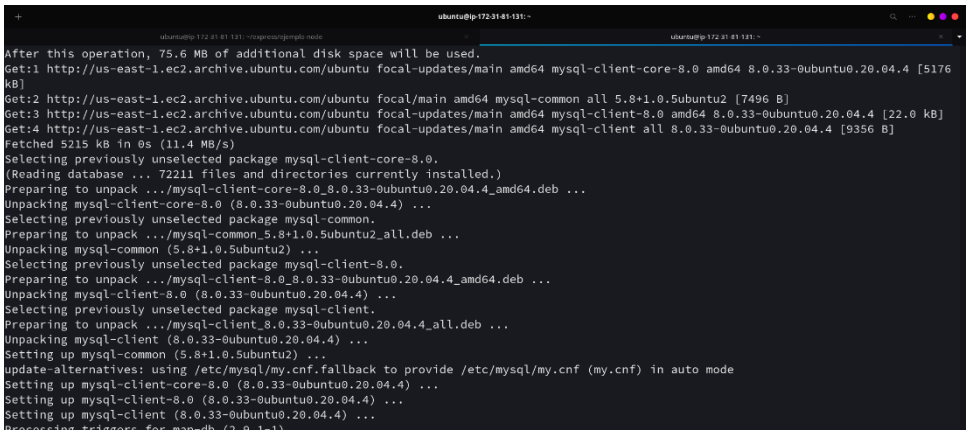
docker run --name mysql-demo -e MYSQL_ROOT_PASSWORD=demo -d -v mysql-demo:/var/lib/mysql mysql:latest

docker ps



Instalar cliente de MySql

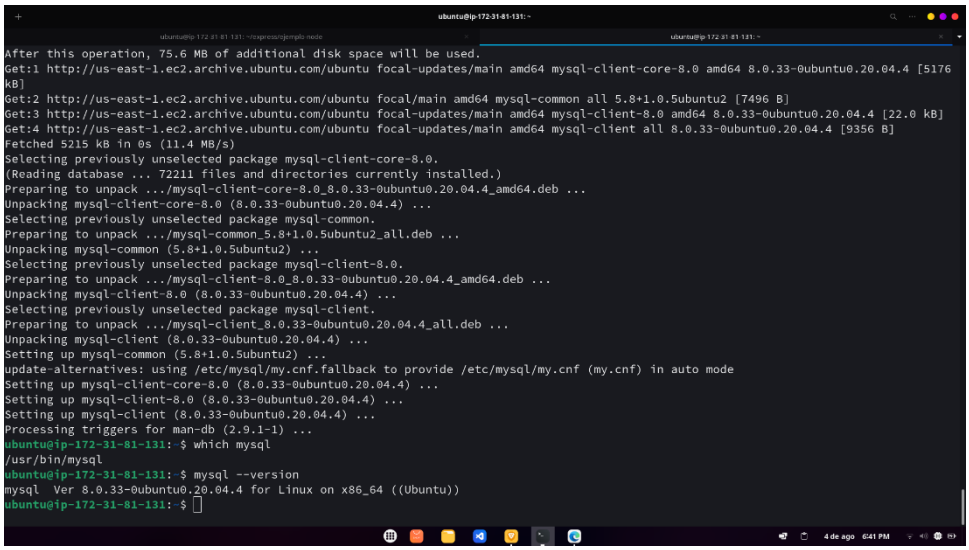
sudo apt install mysql-client -y



Habilitar la extracción de instalaciòn y versiòn de MySQL

which mysql

mysql --version

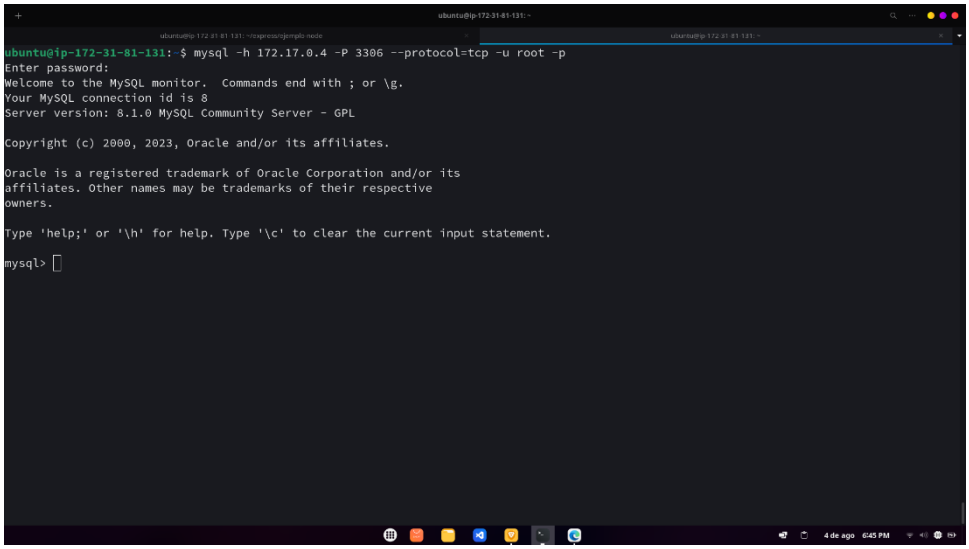


Establecer comunicaciòn

mysql -u root -p

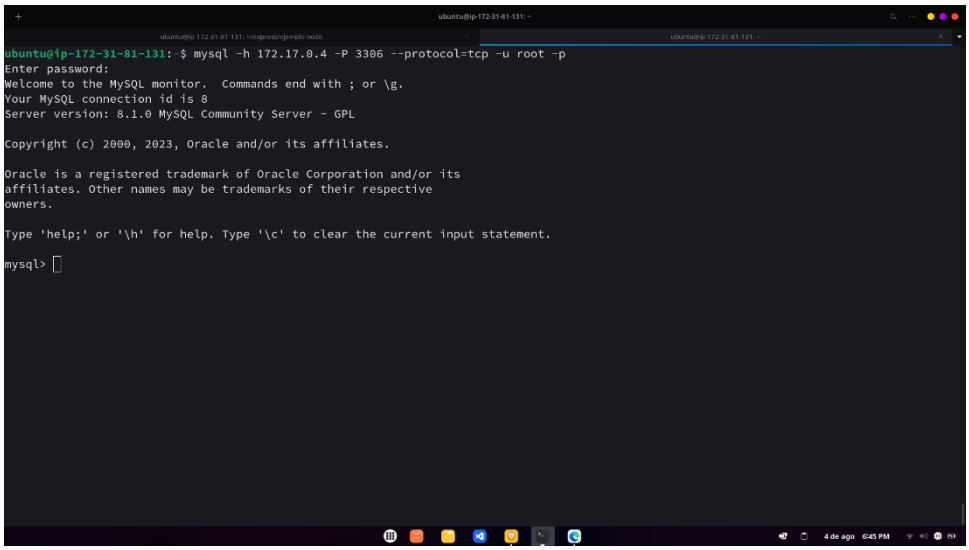
El comando de inspección ayuda a asignar una dirección IP a la instancia del servidor MySQL:

docker inspect -f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' mysql-demo



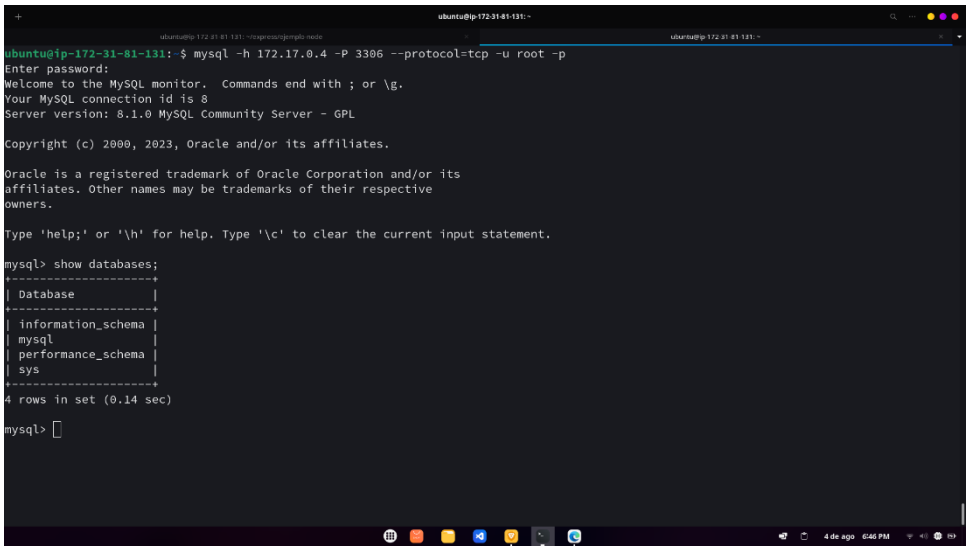
Proporcionar la dirección IP anterior en la opción de host del cliente, con el número de puerto predeterminado y el tipo de protocolo como TCP:

mysql -h 172.17.0.2 -P 3306 --protocol=tcp -u root -p



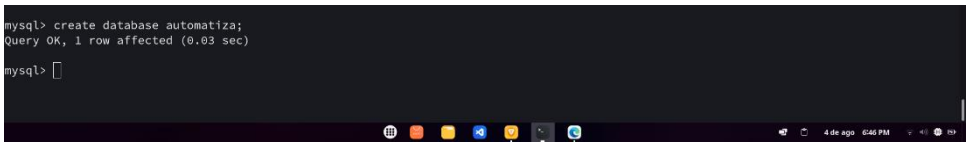
Ingresar el siguiente comando para mostrar las base de datos

show databases;



Crear una base de datos

create database automatiza;



Cambiarse a la base de datos

use automatiza;

Crear una tabla llamada usuarios con los siguientes campos

Campo	Tipo de datos	Indices
username	varchar(30) PK	
password	varchar(20)	
email	varchar(80)	UNIQUE
telefono	varchar(10)	UNIQUE

```
mysql> create database automatiza;
Query OK, 1 row affected (0.03 sec)

mysql> use automatiza;
Database changed
mysql> CREATE TABLE usuarios (
    ->     username VARCHAR(30) PRIMARY KEY,
    ->     password VARCHAR(20),
    ->     email VARCHAR(80) UNIQUE,
    ->     telefono VARCHAR(10) UNIQUE
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql>
```

Ejecutar comando quit para salir de cliente mysql

quit

Para el contenedor

sudo docker stop mysql-demo

```
ubuntu@ip-172-31-81-131:~$ sudo docker stop mysql-demo
mysql-demo
ubuntu@ip-172-31-81-131:~$
```

Eliminar el contendor

sudo docker rm mysql-demo

Creando un contenedor de red

docker network create network-mysql

Modificar archivo para crear contenedor y agregar comando

**--network network-mysql **