

# Capturas de pantalla trabajo AWS Next Digital: Primera parte

1. **Creando la instancia EC2:** Creamos una instancia EC2 que servirá como el servidor de la aplicación web.

### Lanzar una instancia

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

#### Nombre y etiquetas

Nombre

[Agregar etiquetas adicionales](#)

#### Imágenes de aplicaciones y sistemas operativos (Imagen de máquina de Amazon)

Una AMI es una plantilla que contiene la configuración de software (sistema operativo, servidor de aplicaciones y aplicaciones) necesaria para lanzar la instancia. Busque o examine las AMI si no ve lo que busca a continuación.

Inicio rápido

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

SUSE Linux

Debian

[Buscar más AMI](#)  
Inclusión de AMI de AWS, Marketplace y la comunidad

#### Resumen

Número de instancias: 1

Imagen de software (AMI)  
Amazon Linux 2023 AMI 2023.7.2...[más información](#)  
ami-0e449927258d45bc4

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

Firewall (grupo de seguridad)  
Nuevo grupo de seguridad

Almacenamiento (volúmenes)  
Volúmenes: 1 (8 GiB)

Nivel gratuito: Durante el primer año que abra una cuenta de AWS, obtiene 750 horas al mes de uso de instancias t2.micro (o t3.micro cuando t2.micro no esté disponible) si se utiliza

Cancelar [Lanzar instancia](#)

[Código de versión preliminar](#)

### Par de claves (inicio de sesión)

Puede utilizar un par de claves para conectarse de forma segura a la instancia. Asegúrese de que de lanzar la instancia.

Nombre del par de claves - obligatorio

2. **Configuramos el grupo de seguridad de la EC2:** Para que puedan acceder a la página web a través de HTTP.

Detalles Estado y alarmas Monitoreo Seguridad Redes Almacenamiento Etiquetas

#### Detalles de seguridad

|  |                              |  |
|--|------------------------------|--|
| Rol de IAM   | ID del propietario           | Hora de lanzamiento  |
| -  | <a href="#">032164870763</a> | Tue Apr 22 2025 16:05:27 GMT+0200 (hora de verano de Europa central) |
| Grupos de seguridad                                    |                              |  |
| <a href="#">sg-0871192aec03b1df8 (launch-wizard-1)</a> |                              |  |

Reglas de entrada Reglas de salida Compartiendo : novedad Asociaciones de VPC : novedad Etiquetas

#### Reglas de entrada (1)

| <input type="checkbox"/> | Name | ID de la regla del gr... | Versión de IP | Tipo | Protocolo | Intervalo de puerto |
|--------------------------|------|--------------------------|---------------|------|-----------|---------------------|
| <input type="checkbox"/> | -    | sgr-060d7cfb6895c7771    | IPv4          | SSH  | TCP       | 22                  |

(Instalamos nginx como servidor web)

```
[ec2-user@ip-172-31-25-125 ~]$ sudo dnf install git
Last metadata expiration check: 0:02:34 ago on Tue Apr 22 14:09:49 2025.
Dependencies resolved.
=====
Package                                Architecture      Version           Repository        Size
=====
Installing:
git                                     x86_64            2.47.1-1.amzn2023.0.2  amazonlinux      54 k
Installing dependencies:
git-core                               x86_64            2.47.1-1.amzn2023.0.2  amazonlinux      4.7 M
git-core-doc                           noarch            2.47.1-1.amzn2023.0.2  amazonlinux      2.8 M
perl-Error                             noarch            1:0.17029-5.amzn2023.0.2  amazonlinux      41 k
perl-File-Find                         noarch            1.37-477.amzn2023.0.6    amazonlinux      26 k
perl-Git                               noarch            2.47.1-1.amzn2023.0.2  amazonlinux      42 k
perl-TermReadKey                       x86_64            2.38-9.amzn2023.0.2     amazonlinux      36 k
perl-lib                               x86_64            0.65-477.amzn2023.0.6    amazonlinux      15 k
=====
Transaction Summary
-----
Install      8 Packages

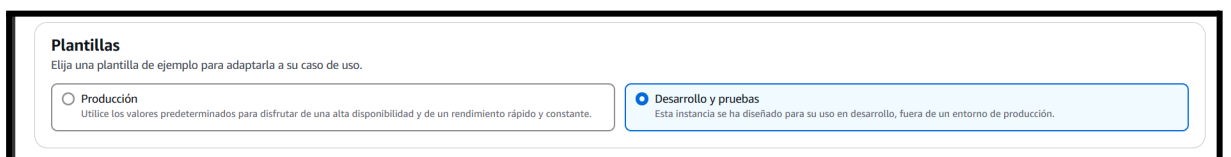
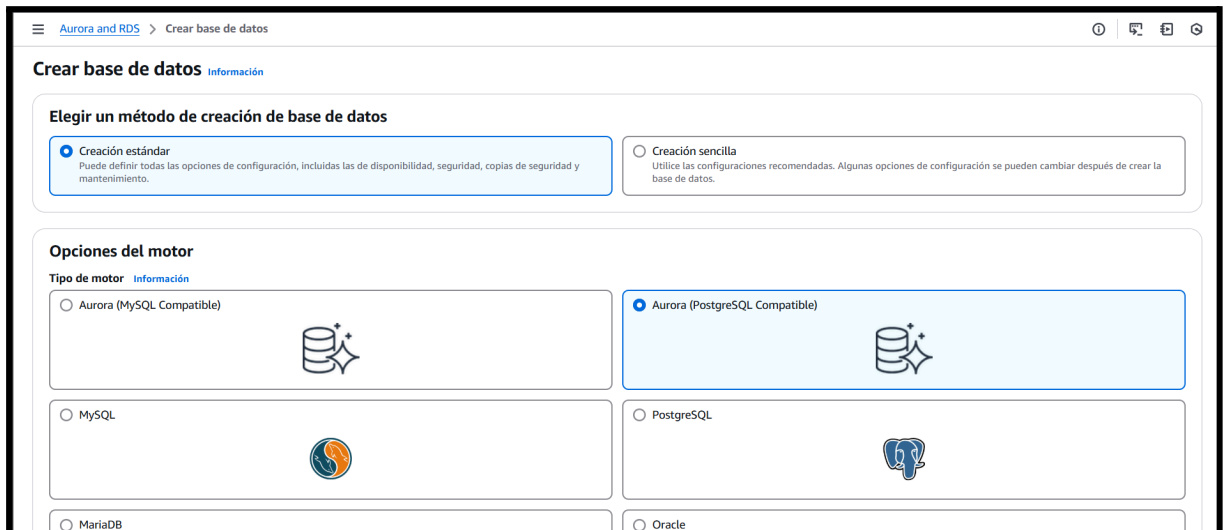
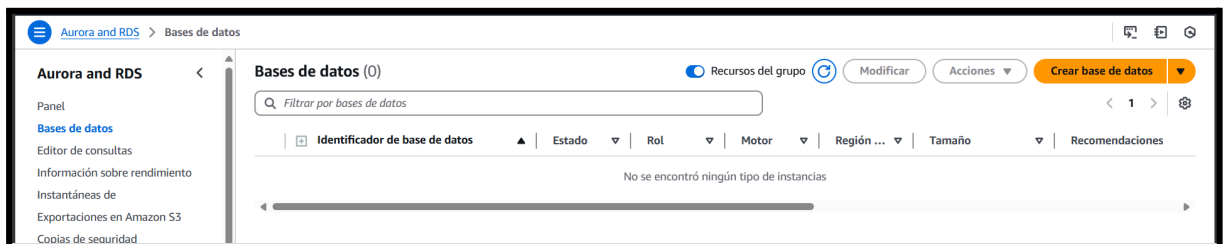
Total download size: 7.7 M
Installed size: 37 M
```

(Instalamos git para clonar el repo, es que hacer uno desde la consola es más pesado. Mejor lo hacemos en local y lo descargamos de github)

```
Complete!
[ec2-user@ip-172-31-25-125 ~]$ git clone 'https://github.com/ManuMedinal/AWS-parte-uno-APP-Web.git'
Cloning into 'AWS-parte-uno-APP-Web'...
remote: Enumerating objects: 23, done.
remote: Counting objects: 100% (23/23), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 23 (delta 0), reused 17 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (23/23), 6.29 KiB | 6.29 MiB/s, done.
[ec2-user@ip-172-31-25-125 ~]$ ls
AWS-parte-uno-APP-Web
```

(Clonamos el repo, de momento, dejamos aquí EC2)

#### 4. Creamos la BBDD con RDS: Será la base de datos donde se guardarán los datos.



### Configuración

**Identificador del clúster de base de datos** [Información](#)

Ingrese un nombre para el clúster de base de datos. El nombre debe ser único entre todos los clústeres de base de datos de la cuenta de AWS de la región de AWS actual.

El identificador del clúster de base de datos no distingue entre mayúsculas y minúsculas, pero se almacena todo en minúsculas (por ejemplo, "mydbcluster"). Restricciones: de 1 a 63 caracteres alfanuméricos o guiones. El primer carácter debe ser una letra. No puede contener dos guiones consecutivos. No puede terminar con un guion.

▼ Configuración de credenciales

**Nombre de usuario maestro** [Información](#)

Escriba un ID de inicio de sesión para el usuario maestro de la instancia de base de datos.

1 a 16 caracteres alfanuméricos. El primer carácter debe ser una letra.

**Administración de credenciales**

Puede usar AWS Secrets Manager o administrar sus credenciales de usuario maestro.

☐ Administrado en AWS Secrets Manager - *más seguro*  
RDS genera una contraseña y la administra durante todo su ciclo de vida mediante AWS Secrets Manager.

☒ Autoadministrado  
Cree su propia contraseña o pida a RDS que cree una contraseña para que pueda administrarla.

☐ Generar contraseña automáticamente  
Amazon RDS puede generar una contraseña en su nombre, o bien puede especificar su propia contraseña.

**Contraseña maestra** [Información](#)

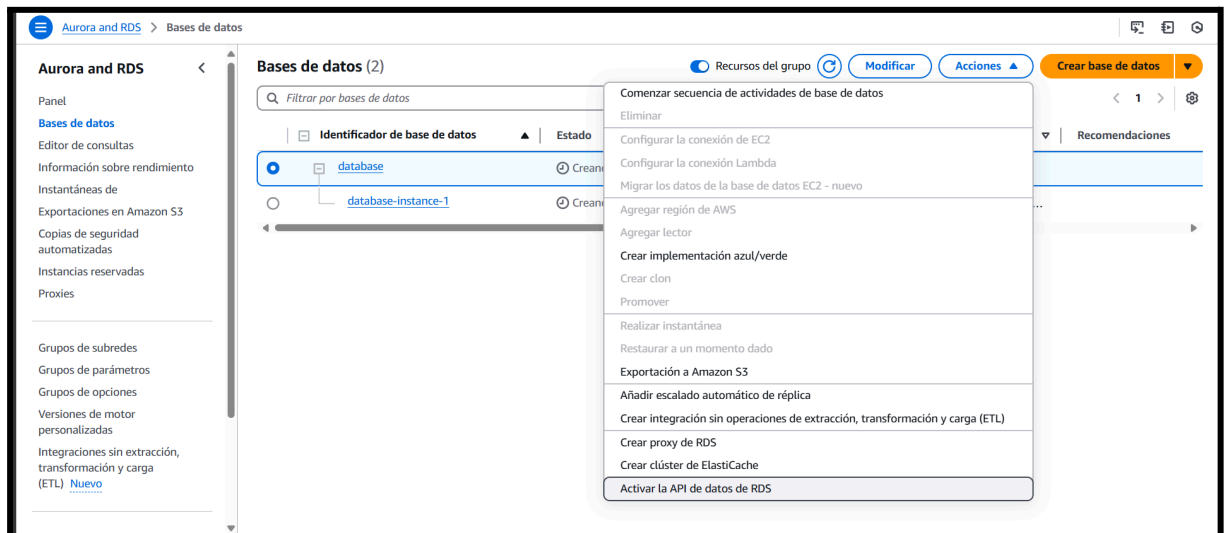
**Seguridad de la contraseña** Muy débil

**Información sobre rendimiento**

☐ Habilitar información sobre rendimiento

Con el panel de Performance Insights, puede visualizar la carga de la base de datos en la carga de la instancia de base de datos de Amazon RDS y filtrar la carga por esperas, sentencias SQL, hosts o usuarios.

## 5. Activamos la API de datos de la RDS: Para poder lanzar consultas SQL desde AWS.



## 6. Conectamos a la BBDD: Para crear las tablas y los ejemplos.

## Conectar con la base de datos

Debe elegir una base de datos e introducir las credenciales de la base de datos para utilizar el editor de consultas. Almacenaremos sus credenciales y la conexión en el servicio AWS Secrets Manager. [Más información](#)

**Instancia o clúster de base de datos**

database

**Nombre de usuario de la base de datos**

Añadir nuevas credenciales de base de datos

**Escriba el nombre de usuario de la base de datos**

manu

**Escriba la contraseña de la base de datos**

.....

**Escriba el nombre de la base de datos**

postgre

C Conectando...

[Cancelar](#) [C Conectar con la base de datos](#)

7. Creamos la BBDD y los ejemplos: Ejecutando sentencias SQL en la AWS console.

```
1 CREATE DATABASE mi_base_de_datos;
2 CREATE TABLE fechas_importantes (
3     id SERIAL PRIMARY KEY,
4     nombre VARCHAR(100) NOT NULL,
5     fecha DATE NOT NULL
6 );
7 INSERT INTO fechas_importantes (nombre, fecha) VALUES ('Cumpleaños de Manuel', '2025-06-15');
8 INSERT INTO fechas_importantes (nombre, fecha) VALUES ('Navidad', '2025-12-25');
9 SELECT * from fechas_importantes;
```

[Ejecutar](#) [Guardar](#) [Borrar](#) [Cambiar base de datos](#)

Resultado:

8. Conectamos la RDS con la EC2: Donde desde la EC2, vamos a conectarnos a la RDS.

Aurora and RDS

Bases de datos

Configurar conexión de EC2

Paso 1

Configurar conexión de EC2

Paso 2

Revisar y confirmar

Configurar conexión de EC2

Información

Seleccionar instancia de EC2

Base de datos  
database

Instancia de EC2  
Elija la instancia de EC2 para conectarse a esta base de datos. Solo se muestran las instancias de EC2 de la misma VPC que la base de datos. Si no hay ninguna instancia de EC2 disponible en la misma VPC, puede crear una nueva instancia de EC2.

i-09f9f37e1b2d83cb7

mi-servidor-web-aws-next-digital

us-east-1d

Crear instancia EC2

Cancelar

Continuar

Cambios en la base de datos de RDS: database

| Atributo           | Valor actual | Valor nuevo        |
|--------------------|--------------|--------------------|
| Grupo de seguridad | default      | default, rds-ec2-1 |

Cambios en la instancia de EC2: i-09f9f37e1b2d83cb7

| Atributo           | Valor actual    | Valor nuevo                |
|--------------------|-----------------|----------------------------|
| Grupo de seguridad | launch-wizard-1 | launch-wizard-1, ec2-rds-1 |

⚠ Pueden aplicarse cargos entre zonas de disponibilidad (AZ)

La base de datos de RDS database (us-east-1b) y la instancia de EC2 i-09f9f37e1b2d83cb7 (us-east-1d) se encuentran en zonas de disponibilidad diferentes. Es posible que se apliquen cargos entre zonas de disponibilidad. [Transferencia de datos dentro de la misma Región](#)

Cancelar

Anterior

Configurar

9. Configuramos los grupos de seguridad: Para poder conectarnos sin problemas desde la EC2 hacia la RDS.

<

Conectividad y seguridad

Supervisión

Registros y eventos

Configuración

Mantenimiento y copias de seguridad

Etiquetas

>

Conectividad y seguridad

Punto de enlace y puerto

Punto de enlace  
database-instance-1.cdru4ykthknh.us-east-1.rds.amazonaws.com

Puerto  
5432

Redes

Zona de disponibilidad  
us-east-1c

VPC  
vpc-02affb47634ccf9e7

Seguridad

Grupos de seguridad de la VPC  
default (sg-0fcf2203752a9ad2a)

Activo

Accesible públicamente  
No

Reglas de entrada
Reglas de salida
Compartiendo : *novedad*
Asociaciones de VPC : *novedad*
Etiquetas

Reglas de entrada (1)
Administrar etiquetas
Editar reglas de entrada

| <input type="checkbox"/> | Name | ID de la regla del gr... | Versión de IP | Tipo            | Protocolo | Intervalo de puerto |
|--------------------------|------|--------------------------|---------------|-----------------|-----------|---------------------|
| <input type="checkbox"/> | -    | sgr-09324e29c58d4cfa1    | -             | Todo el tráfico | Todo      | Todo                |

Editar reglas de entrada *Información*

Las reglas de entrada controlan el tráfico entrante que puede llegar a la instancia.

Reglas de entrada *Información*

| ID de la regla del grupo de seguridad | Tipo <i>Información</i> | Protocolo <i>Información</i> | Intervalo de puertos <i>Información</i> | Origen <i>Información</i> | Descripción: opcional <i>Información</i> |          |
|---------------------------------------|-------------------------|------------------------------|---|---------------------------|--|----------|
| sgr-09324e29c58d4cfa1                 | Todo el tráfico         | Todo                         | Todo                                    | Perso...                  |  | Eliminar |
| -                                     | TCP personalizado       | TCP                          | 5432                                    | Perso...                  | sg-04b6404093e4455                       | Eliminar |

Agregar regla

Cancelar
Previsualizar los cambios
Guardar reglas

10. Comprobamos que se puede leer los datos de la BBDD: Vamos a la instancia EC2 para ver si todo esta bien.

```
[ec2-user@ip-172-31-25-125 ~]$ sudo dnf install -y postgresql16
Last metadata expiration check: 0:25:55 ago on Tue Apr 22 14:09:49 2025.
Dependencies resolved.
=====
Package                                Architecture      Version           Repository        Size
-----
Installing:
 postgresql16                                x86_64            16.8-1.amzn2023.0.2    amazonlinux        1.9 M
Installing dependencies:
 postgresql16-private-libs                  x86_64            16.8-1.amzn2023.0.2    amazonlinux        144 k
Transaction Summary
-----
Install 2 Packages
Total download size: 2.0 M
Installed size: 8.3 M
Downloading Packages:
(1/2): postgresql16-private-libs-16.8-1.amzn2023.0.2.x86_64.rpm    1.7 MB/s | 144 kB    00:00
(2/2): postgresql16-16.8-1.amzn2023.0.2.x86_64.rpm                14 MB/s | 1.9 MB    00:00
-----
Total                                                                    11 MB/s | 2.0 MB    00:00
Running transaction check
Transaction check succeeded.
Running transaction test
```

(Instalamos postgresql 16)

Conectividad y seguridad
Supervisión
Registros y eventos
Configuración
Integraciones sin extracción, transformación y carga

Puntos de conexión (2)
Acciones
Crear un punto de conexión personalizado

|  | Nombre del punto de enlace                                  | Estado     | Tipo     | Puerto |
|--|---|------------|----------|--------|
|  | database.cluster-cdru4ythknh.us-east-1.rds.amazonaws.com    | Disponible | Escritor | 5432   |
|  | database.cluster-ro-cdru4ythknh.us-east-1.rds.amazonaws.com | Disponible | Lector   | 5432   |

(Buscamos el punto de conexión de la BBDD)

11. Conectamos la RDS con la EC2: Donde desde la EC2, vamos a conectarnos a la RDS.

```
[ec2-user@ip-172-31-25-125 ~]$ psql -h database.cluster-ro-cdr4ykthknh.us-east-1.rds.amazonaws.com -U manu -d postgres -p 5432
Password for user manu:
psql (16.8, server 16.6)
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off)
Type "help" for help.
```

(Conectamos a la BBDD)

```
postgres=> select * from fechas_importantes;
 id |          nombre          |      fecha
-----+-----+-----
  1 | Cumpleaños de Manuel    | 2025-06-15
  2 | Navidad                 | 2025-12-25
  3 | Cumpleaños de Manuel    | 2025-06-15
  4 | Navidad                 | 2025-12-25
(4 rows)

postgres=>
```

(Hacemos una consulta SQL de ejemplo para ver si funciona)

**Ahora, estos pasos son para arrancar nuestra app web:**

**1. Instalamos pip:** Para poder instalar después las dependencias.

```
[ec2-user@ip-172-31-25-125 AWS-parte-uno-APP-Web]$ sudo dnf install python3-pip -y
Last metadata expiration check: 0:46:30 ago on Tue Apr 22 14:09:49 2025.
Dependencies resolved.

Package                               Architecture      Version              Repository           Size
-----
Installing:
python3-pip                           noarch            21.3.1-2.amzn2023.0.11  amazonlinux          1.8 M
Installing weak dependencies:
libxcrypt-compat                      x86_64            4.4.33-7.amzn2023      amazonlinux           92 k
```

**2. Instalamos las dependencias para ejecutar la APP:** Todas están en el archivo requirements.txt.

```
[ec2-user@ip-172-31-25-125 AWS-parte-uno-APP-Web]$ pip install -r requirements.txt
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: psycopg2-binary==2.9.10 in /home/ec2-user/.local/lib/python3.9/site-packages (from -r requirements.txt (line 1)) (2.9.10)
Collecting flask==3.1.0
  Downloading flask-3.1.0-py3-none-any.whl (102 kB)
  102 kB 15.6 MB/s
Collecting importlib-metadata>=3.6
  Downloading importlib_metadata-8.6.1-py3-none-any.whl (26 kB)
Collecting Werkzeug==3.1
  Downloading werkzeug-3.1.3-py3-none-any.whl (224 kB)
  224 kB 14.7 MB/s
Collecting itsdangerous>=2.2
  Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Collecting Jinja2>=3.1.2
  Downloading Jinja2-3.1.6-py3-none-any.whl (134 kB)
  134 kB 15.2 MB/s
Collecting click>=8.1.3
  Downloading click-8.1.8-py3-none-any.whl (98 kB)
  98 kB 12.2 MB/s
Collecting blinker>=1.9
  Downloading blinker-1.9.0-py3-none-any.whl (8.5 kB)
Collecting zipp>=3.20
  Downloading zipp-3.21.0-py3-none-any.whl (9.6 kB)
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-3.0.2-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (20 kB)
Installing collected packages: zipp, MarkupSafe, Werkzeug, Jinja2, itsdangerous, importlib-metadata, click, blinker, Flask
Successfully installed Flask-3.1.0 Jinja2-3.1.6 MarkupSafe-3.0.2 Werkzeug-3.1.3 blinker-1.9.0 click-8.1.8 importlib-metadata-8.6.1 itsdangerous-2.2.0 zipp-3.21.0
```

**3. Ejecutamos:** Para poner en marcha nuestra APP.

```
[ec2-user@ip-172-31-25-125 AWS-parte-uno-APP-Web]$ python3 app.py
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.31.25.125:5000
Press CTRL+C to quit
```



4. **Configuramos el archivo /etc/nginx/nginx.conf:** Para que nginx redirija las peticiones HTTP hacia la APP web.

```
GNU nano 2.9.3 /etc/nginx/nginx.conf
# For more information on configuration, see:
#   * Official English Documentation: http://nginx.org/en/docs/
#   * Official Russian Documentation: http://nginx.org/ru/docs/

user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

http {
    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for"';

    access_log /var/log/nginx/access.log main;

    sendfile            on;
    tcp_nopush          on;

    keepalive_timeout   65;

    include              conf.d/*.conf;
}
```

Este es el codigo por si acaso:

```
# For more information on configuration, see:
# * Official English Documentation: http://nginx.org/en/docs/
# * Official Russian Documentation: http://nginx.org/ru/docs/
```

```
user nginx;  
worker_processes auto;  
error_log /var/log/nginx/error.log notice;  
pid /run/nginx.pid;
```

```
# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;
```

```
events {
    worker_connections 1024;
}
```

```
http {
    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for";
```

```
access_log /var/log/nginx/access.log main;
```

```
sendfile      on;
tcp_nopush    on;
keepalive_timeout 65;
types_hash_max_size 4096;
```

```
include      /etc/nginx/mime.types;
```

```
default_type    application/octet-stream;

# Load modular configuration files from the /etc/nginx/conf.d directory.
# See http://nginx.org/en/docs/nginx_core_module.html#include
# for more information.
include /etc/nginx/conf.d/*.conf;

server {
    listen 80;
    server_name _;

    location / {
        proxy_pass http://127.0.0.1:5000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}
```

5. **¡Listo!** A través de nuestra DNS de AWS, debería de verse nuestra APP web:



Agregar una Fecha Importante

Nombre: Aniversario de Next Digital

Fecha: 09/02/2026

Guardar

Volver a la lista

Lista de Fechas Importantes

Cumpleaños de Manuel - 2025-06-15

Cumpleaños de Manuel - 2025-06-15

Navidad - 2025-12-25

Navidad - 2025-12-25

Aniversario Next Digital - 2026-02-09

Agregar nueva fecha

(No me mateis si esta mal la fecha del aniversario de Next, no tengo buena memoria xd)

Por último, vamos a comprobar si se guardó la nueva fecha:

```
[ec2-user@ip-172-31-25-125 AWS-partes-uno-APP-Web]$ psql -h database.cluster-ro-cdr4ytkthknh.us-east-1.rds.amazonaws.com -U manu -d postgres -p 5432
Password for user manu:
psql (16.8, server 16.6)
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off)
Type "help" for help.

postgres=> select * from fechas importantes;
 id | nombre | fecha
----+-----+-----
  1 | Cumpleaños de Manuel | 2025-06-15
  2 | Navidad | 2025-12-25
  3 | Cumpleaños de Manuel | 2025-06-15
  4 | Navidad | 2025-12-25
  5 | Aniversario Next Digital | 2026-02-09
(5 rows)
```

En efecto, la APP web si funciona. ¡Ya podemos guardar las fechas importantes!