

Examen No. 1 – Valor 20%

Lógica Almacenada en Base de Datos

**Análisis de Recursos Humanos utilizados en Proyecto de Software de Nivel
Empresarial**

Fecha de Entrega: miércoles 26 de febrero de 2025, 10:00 pm (GMT -5)

Tiempo estimado de ejecución: 10 horas

Correo: mariana.osorioro@upb.edu.co

Estudiante: Mariana Osorio Rojas

ID: 464679

Universidad Pontificia Bolivariana

Facultad de Ingenierías - Ingeniería de Sistemas e Informática

Tópicos Avanzados en Bases de Datos

ABASTECIMIENTO POSTGRESQL CON INFRAESTRUCTURA EN NUBE AWS

1. Crear una Cuenta en AWS

- Ir a AWS Free Tier y registrarse.
- Seguir los pasos de verificación.
- Asegurarse que la cuenta esté habilitada para servicios gratuitos.

2. Crear una Instancia EC2

- Ir a la consola de AWS → AWS Console
- En el buscador, escribir "EC2" y seleccionar.
- Haz clic en "Launch Instance".
- Eligir una AMI: Seleccionar Ubuntu Server (gratuito).

Quick Start

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

SUSE Linux

Debian

aws

Mac

ubuntu

Microsoft

Red Hat

SUSE

debian

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type

ami-0cb91c7de36eed2cb (64-bit (x86)) / ami-0ac5d9e789dbb455a (64-bit (Arm))

Free tier eligible

Virtualization: hvm

ENA enabled: true

Root device type: ebs

Description

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture

AMI ID

Username

64-bit (x86)

ami-0cb91c7de36eed2cb

ubuntu

Verified provider

- Elegir el tipo de instancia: Seleccionar t2.micro (dentro del Free Tier).

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t2.micro

Free tier eligible

Family: t2

1 vCPU

1 GiB Memory

Current generation: true

On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour

On-Demand Linux base pricing: 0.0116 USD per Hour

On-Demand SUSE base pricing: 0.0116 USD per Hour

On-Demand Windows base pricing: 0.0162 USD per Hour

On-Demand RHEL base pricing: 0.026 USD per Hour

[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

- Configurar los detalles de la instancia: En este caso, dejar por defecto.
- Configurar el grupo de seguridad:**
 - SSH, puerto 22, solo mi IP (para la seguridad).
 - PostgreSQL, puerto 5432, solo mi IP (para la seguridad).

Type Info ssh	Protocol Info TCP	Port range Info 22
Source type Info Anywhere	Source Info Q Add CIDR, prefix list or security group 0.0.0.0/0 X	Description - optional Info e.g. SSH for admin desktop

▼ Security group rule 2 (TCP, 5432, 0.0.0.0/0) Remove



Type Info PostgreSQL	Protocol Info TCP	Port range Info 5432
Source type Info Anywhere	Source Info Q Add CIDR, prefix list or security group	Description - optional Info e.g. SSH for admin desktop

Si hay más IP's de confianza, agregarlas individualmente.

h. Descargar la llave .pem y lanzar la instancia.

3. Conectarse a la Instancia EC2

a. Copiar la IP Pública de nuestra instancia.

Public IPv4 address
 18.189.193.27 | [open address](#) 

b. Copiar la ruta de la llave .pem que previamente descargamos.

C:\Users\User\Downloads\rrhh_key.pem

c. Usar SSH para conectarse a la instancia desde nuestro terminal local:

```
C:\Users\User>ssh -i \Users\User\Downloads\rrhh_key.pem ubuntu@18.189.193.27
The authenticity of host '18.189.193.27 (18.189.193.27)' can't be established.
ED25519 key fingerprint is SHA256:oqgdgVj1MFdeEXAWFmaEQtyZAwY7f9R8GQokmdhS3t8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '18.189.193.27' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1021-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/pro

System information as of Tue Feb 25 04:12:10 UTC 2025

System load:  0.0          Processes:            105
Usage of /:   6.0% of 28.02GB Users logged in:       0
Memory usage: 19%         IPv4 address for enX0: 172.31.12.100
Swap usage:   0%
```

4. Instalar y Configurar PostgreSQL

- Instalar paquetes:
- Instalar PostgreSQL:
- Iniciar y habilitar PostgreSQL:
- Entra a PostgreSQL:

```
ubuntu@ip-172-31-12-100:~$ sudo apt update && sudo apt upgrade -y
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [619 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [871 kB]
Get:15 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [197 kB]

ubuntu@ip-172-31-12-100:~$ sudo apt install postgresql postgresql-contrib -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcommon-sense-perl libjson-perl libjson-xs-perl libllvm17t64 libpq5 libtypes-serialise
  postgresql-client-16 postgresql-client-common postgresql-common ssl-cert
Suggested packages:
  postgresql-doc postgresql-doc-16
The following NEW packages will be installed:
  libcommon-sense-perl libjson-perl libjson-xs-perl libllvm17t64 libpq5 libtypes-serialise
  postgresql-16 postgresql-client-16 postgresql-client-common postgresql-common postgresql
0 upgraded, 13 newly installed, 0 to remove and 0 not upgraded.
Need to get 43.5 MB of archives.
After this operation, 175 MB of additional disk space will be used.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-12-100:~$ sudo systemctl start postgresql
ubuntu@ip-172-31-12-100:~$ sudo systemctl enable postgresql
Synchronizing state of postgresql.service with SysV service script with /usr/
Executing: /usr/lib/systemd/systemd-sysv-install enable postgresql
ubuntu@ip-172-31-12-100:~$ sudo -u postgres psql
psql (16.6 (Ubuntu 16.6-0ubuntu0.24.04.1))
Type "help" for help.
```

- Crear la base de datos:
- Crear el usuario admin con contraseña *(diferente del default, para mejorar la seguridad de nuestro usuario con más privilegios)*

Por temas de seguridad, se recomienda una contraseña con cifrado SHA-256

```
postgres=# CREATE DATABASE rrhh_db;
CREATE DATABASE
postgres=# CREATE USER rrhh_admin WITH PASSWORD 'rrhh_pass';
CREATE ROLE
```

- g. Dar permisos de administrador solo para el usuario admin.
- h. Salir del entorno de PostgreSQL.

```
postgres=# GRANT ALL PRIVILEGES ON DATABASE rrhh_db TO rrhh_admin;
GRANT
postgres=# GRANT ALL PRIVILEGES ON SCHEMA public TO rrhh_admin;
GRANT
postgres=# ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT ALL ON TABLES TO rrhh_admin;
ALTER DEFAULT PRIVILEGES
postgres=# GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public TO rrhh_admin;
GRANT
postgres=# ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT ALL ON SEQUENCES TO rrhh_admin;
ALTER DEFAULT PRIVILEGES
postgres=# GRANT ALL PRIVILEGES ON ALL FUNCTIONS IN SCHEMA public TO rrhh_admin;
GRANT
postgres=# ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT ALL ON FUNCTIONS TO rrhh_admin;
ALTER DEFAULT PRIVILEGES
postgres=# ALTER USER rrhh_admin WITH SUPERUSER;
ALTER ROLE
postgres=# \q
```

- i. Configurar PostgreSQL para permitir conexiones remotas:

- i. Editar pg_hba.conf:

1. sudo nano /etc/postgresql/*/main/pg_hba.conf
2. Al final agregar *(permitimos solo las conexiones específicas de nuestros usuarios)*:
 - a. *host rrhh_db rrhh_admin TU_IP_PRIVADA/32 md5*
 - b. Guardar y cerrar (Ctrl + X, luego Y y Enter).

- ii. Editar postgresql.conf:

1. sudo nano /etc/postgresql/*/main/postgresql.conf
2. Buscar línea: listen_addresses = 'localhost'
3. Cambiarla:
 - a. listen_addresses = '*'
4. Guardar y cerrar (Ctrl + X, luego Y y Enter).

- j. Reiniciar PostgreSQL:

- i. sudo systemctl restart postgresql

CONEXIONES CIFRADAS PARA LA SEGURIDAD AL ACCESO DE DATOS

- 1. Configuración de puertos con IP's conocidas desde EC2 (ya hecho).**
- 2. Usuario diferente del default (ya hecho).**
- 3. Contraseñas fuertes con cifrado SHA-256 (ya hecho).**
- 4. Creación de varios usuarios con diferentes privilegios.**
 - a. Crear usuario de solo lectura:**
 - i. `CREATE USER rrhh_readonly WITH PASSWORD 'readonly_pass';`
 - b. Otorgar permisos de lectura sobre todas las tablas:**
 - i. `GRANT CONNECT ON DATABASE rrhh_db TO rrhh_readonly;`
 - ii. `GRANT USAGE ON SCHEMA public TO rrhh_readonly; GRANT`
 - iii. `SELECT ON ALL TABLES IN SCHEMA public TO rrhh_readonly;`
 - iv. `ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT`
`ON TABLES TO rrhh_readonly;`
- 5. Configuración de conexiones remotas específicas (ya hecho).**
- 6. Habilitar conexión cifrada con SSL.**
 - a. Generar certificados SSL en la instancia EC2:**
 - i. `sudo mkdir /etc/postgresql/certs`
 - ii. `sudo openssl req -new -x509 -days 365 -nodes -out`
`/etc/postgresql/certs/server.crt -keyout`
`/etc/postgresql/certs/server.key`
 - iii. `sudo chmod 600 /etc/postgresql/certs/server.key`
 - b. Edita postgresql.conf para habilitar SSL:**
 - i. `sudo nano /etc/postgresql/15/main/postgresql.conf`
 - ii. `ssl = on`
 - iii. `ssl_cert_file = '/etc/postgresql/certs/server.crt'`
 - iv. `ssl_key_file = '/etc/postgresql/certs/server.key'`
 - c. Reinicia PostgreSQL:**
 - i. `sudo systemctl restart postgresql`
- 7. Seguridad en Respaldo y Recuperación.**
 - a. respaldo manual con pg_dump:**

i. `pg_dump -U rrhh_admin -h localhost -p 5432 -F c -b -v -f
"/backups/rrhh_db.bak" rrhh_db`

b. **Automatizar esto** con un cronjob:

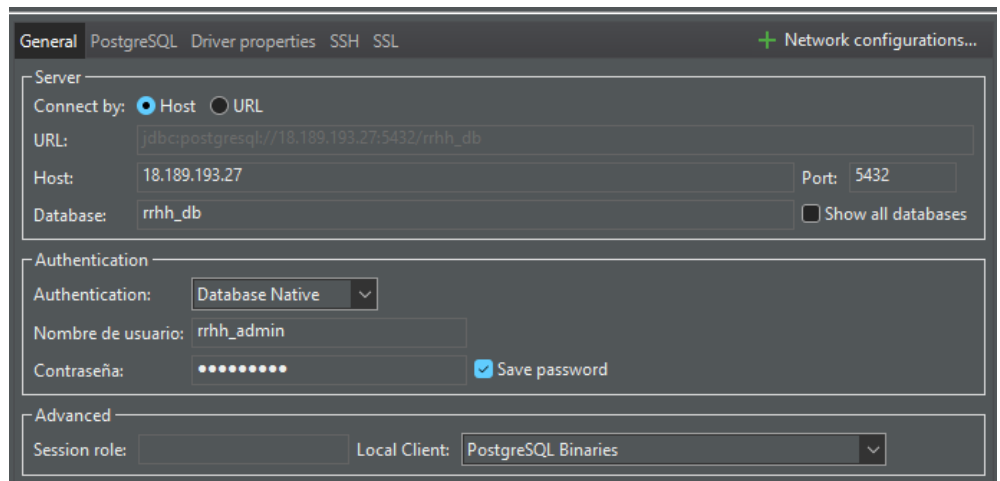
i. `crontab -e`

c. **Agregar esta línea para hacer backups diarios a las 2:00 AM:**

i. `0 2 * * * pg_dump -U rrhh_admin -h localhost -p 5432 -F c -b -v -f
"/backups/rrhh_db_$(date +%F).bak" rrhh_db`

CONEXIÓN REMOTA CON IDE DBEAVER

1. Instalar DBeaver.
2. **Configurar la conexión a PostgreSQL:**
 - a. Abrir DBeaver y dar click en “Nueva Conexión”.
 - b. Seleccionar PostgreSQL.
 - c. **Ingresar los datos:**
 - i. **Host:** 18.189.193.27 (IP Pública de la instancia EC2).
 - ii. **Puerto:** 5432.
 - iii. **Base de datos:** rrhh_db (Nombre de la bd).
 - iv. **Usuario:** rrhh_admin.
 - v. **Contraseña:** rrhh_pass.



- d. Click en “Test Connection”.
- e. Guardar Conexión.

