Tarea V

19 de agosto de 2024

1. Como Crear Usuarios y roles

Con PostgreSQL, puede crear usuarios y roles con permisos de acceso granulares. Al nuevo usuario o rol se les debe conceder selectivamente los permisos necesarios para cada objeto de base de datos. Esto da mucho poder al usuario final, pero al mismo tiempo, dificulta potencialmente el proceso de creación de usuarios y roles con los permisos correctos.

PostgreSQL le permite conceder permisos directamente a los usuarios de la base de datos. Sin embargo, como práctica recomendada, se recomienda crear varios roles con conjuntos específicos de permisos basados en los requisitos de aplicación y acceso. Paso seguido, asigne el rol apropiado a cada usuario. Los roles deben utilizarse para aplicar un modelo de privilegios mínimos para acceder a objetos de base de datos. El usuario maestro que se crea durante la creación de instancias de Amazon RDS y Aurora PostgreSQL solo debe utilizarse para tareas de administración de bases de datos, como la creación de otros usuarios, roles y bases de datos. El usuario maestro nunca debe ser utilizado por la aplicación.

- El método recomendado para configurar un control de acceso detallado en PostgreSQL es el siguiente:

- Utilice el usuario maestro para crear roles por aplicación o caso de uso, como readonly (solo lectura) y readwrite (escritura).
Agregue permisos para permitir que estos roles tengan acceso a varios objetos de base de datos. Por ejemplo, el rol de readonly solo puede ejecutar consultas SELECT.

Conceda a los roles los menos permisos posibles necesarios para la funcionalidad.

- Cree nuevos usuarios para cada aplicación o funcionalidad distinta, como app$_u ser(usuario de aplicación) y reporting_u se$

- Asigne los roles aplicables a estos usuarios para otorgarles rápidamente los mismos permisos que el rol. Por ejemplo, conceda el rol readwrite aapp$_u$ser y conceda el rol de readonly a reporting$_u$ser.

- En cualquier momento, puede quitar el rol del usuario para revocar los permisos.

En PostgreSQL, se pueden asignar diversas propiedades a roles y usuarios. Aquí algunas de las más comunes:

- **CREATEDB**: Permite crear bases de datos.
- **CREATEROLE**: Permite crear nuevos roles.
- **LOGIN**: Permite al rol iniciar sesión en la base de datos.
- **SUPERUSER**: Otorga todos los privilegios administrativos.
- **INHERIT**: El rol hereda privilegios de roles que le han sido asignados.
- **NOCREATEDB**: Niega la capacidad de crear bases de datos.

- **NOCREATEROLE**: Niega la capacidad de crear roles.
- **NOSUPERUSER**: Niega todos los privilegios administrativos.
- **REPLICATION**: Permite usar las funciones de replicación.
- **CONNECTION LIMIT**: Limita el número de conexiones simultáneas que el rol puede tener.

2. Privilegios a nivel objeto y a nivel sistema

En PostgreSQL, los privilegios se dividen en dos tipos principales: **privilegios a nivel de sistema** y **privilegios a nivel de objeto**.

Privilegios a Nivel de Sistema

Estos privilegios se refieren a permisos que afectan a la base de datos en su conjunto o a la capacidad de un usuario para ejecutar ciertas operaciones a nivel de base de datos. Algunos ejemplos incluyen:

- **CREATEDB**: Permite crear nuevas bases de datos.
- **CREATEROLE**: Permite crear nuevos roles (usuarios o grupos).
- **SUPERUSER**: Concede todos los privilegios posibles en PostgreSQL.
- **LOGIN**: Permite que el rol inicie sesión en la base de datos.
- **REPLICATION**: Permite al rol iniciar la replicación de bases de datos.

Privilegios a Nivel de Objeto

Estos privilegios se aplican a objetos específicos dentro de la base de datos, como tablas, vistas, secuencias, funciones, etc. Ejemplos incluyen:

- **SELECT**: Permite leer datos de una tabla, vista o secuencia.
- **INSERT**: Permite insertar datos en una tabla o vista.
- **UPDATE**: Permite actualizar datos en una tabla o vista.
- **DELETE**: Permite borrar datos de una tabla o vista.
- **USAGE**: Permite usar secuencias y esquemas.

Estos privilegios se asignan a roles (usuarios o grupos) para controlar el acceso y las operaciones permitidas en la base de datos.

3. Como otrorgar y quitar privilegios a un usuario

Resulta muy fácil revocar los privilegios de un usuario. Por ejemplo, puede quitar el permiso de lectura y escritura (readwrite) de myuser1 utilizando la siguiente instrucción SQL:

```
REVOKE readwrite FROM myuser1;
```

Del mismo modo, puede otorgar un nuevo rol de la siguiente manera:

```
GRANT readonly TO myuser1;
```

Referencias

[1] Yaser Raja. “Administración de usuarios y roles de PostgreSQL — Amazon Web Services”. Amazon Web Services. Accedido el 19 de agosto de 2024. [En línea]. Disponible: https://aws.amazon.com/es/blogs/aws-spanish/managing-postgresql-users-and-roles/

[2] R. Domínguez. “Gestión de roles y privilegios en PostgreSQL 12”. Medium. Accedido el 19 de agosto de 2024. [En línea]. Disponible: https://medium.com/@dgzraul.web/gestión-de-roles-y-privilegios-en-postgresql-12-dc6897445a29