Bases de Datos

Juárez Huerta Enrique

Tarea 5: Usuarios

En postgresql se pueden 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.

Los usuarios, grupos y roles son lo mismo en PostgreSQL, y la única diferencia es que los usuarios tienen permiso para iniciar sesión de forma predeterminada. Las instrucciones CREATE USER y CREATE GROUP son en realidad alias de la instrucción CREATE ROLE. Para crear un usuario de PostgreSQL, utilizaremos la siguiente instrucción SQL:

Creacion de usuario:

CREATE USER myuser WITH PASSWORD 'passwd';

Creación de roles:

CREATE ROLE myuser WITH LOGIN PASSWORD 'passwd';

También unos de los comandos más comunes en posgresql para la aasignación de propiedades son:

- CREATEDB: para crear bases de datos
- LOGIN: Permite a un rol iniciar sesión en la base de datos
- SUPERUSER: permite otorgar todos los permisos administrativos
- NOSUPERUSER: Niega todos los permisos administrativos
- REPLICATION: permite utilizar aquellas funciones de replicación
- CONNECTION LIMIT: Limita el numero de conexiones simutaneaas que el rol puede tener

Roles:

Utilice roles de base de datos para gestionar con mayor facilidad los privilegios de los grupos de usuarios.

Los roles de base de datos simplifican el proceso de gestión de privilegios, ya que se pueden otorgar privilegios a un rol y luego otorgar el rol a usuarios. Cuando desee revocar privilegios para un usuario, simplemente tiene que revocar la autorización de rol del usuario, en vez de revocar cada privilegio individual.

Los roles más comunes que se crean son:

- Rol de solo lectura
- Rol de lectura y escritura

Tipos de privilegio:

Hay dos tipos de privilegio que puede otorgar:

Administrador

Los privilegios de administrador controlan la creación de objetos y la administración de sistema.

Ejemplo:

- Copia seguridad: Permite al usuario crear copias de seguridad. El usuario puede ejecutar el comando nzbackup.
- [Crear] Base de datos: Permite al usuario crear bases de datos. El permiso para operar en bases de datos existentes está controlado por los privilegios de objeto.
- [Crear] Biblioteca: Permite al usuario crear bibliotecas compartidas. El permiso para operar en bibliotecas compartidas existentes está controlado por los privilegios de objeto.

Entre muchos otros más.

Objeto

Los privilegios de objeto acceder a objetos de base de datos específicos.

Algunos privilegios de administrador son globales en cuanto al alcance, independientemente de la base de datos actual. Por ejemplo, los privilegios de base de datos, usuario, grupo, sistema y administrador de hardware son privilegios globales. Los demás privilegios administrativos pueden ser globales o locales dependiendo de la base de datos actual.

Ejemplos:

- Abort: Permite al usuario anular sesiones. Se aplica a grupos y usuarios.
- Alter: Permite al usuario modificar atributos de objeto. Se aplica a todos los objetos.
- Delete: Permite al usuario suprimir filas de tabla. Sólo se aplica solo a tablas.
- Execute: Permite al usuario ejecutar funciones definidas por el usuario, agregados definidos por el usuario o procedimientos almacenados.

Entre muchos otros más.

Otorgar y remover privilegios a usuarios:

Para poder asignar los privilegios deseados a un usuario utilizaremos el comando GRANT.

Ejemplo:

GRANT CONNECT ON DATABASE mydatabase TO readonly;

Para poder revocar los privilegios deseados a un usuario utilizaremos el comando REVOKE.

Ejemplo:

REVOKE CREATE ON SCHEMA public FROM PUBLIC;

https://www.qualoom.es/blog/administracion-usuarios-roles-postgresql/

https://www.ibm.com/docs/es/psfa/7.1.0?topic=language-types-privileges