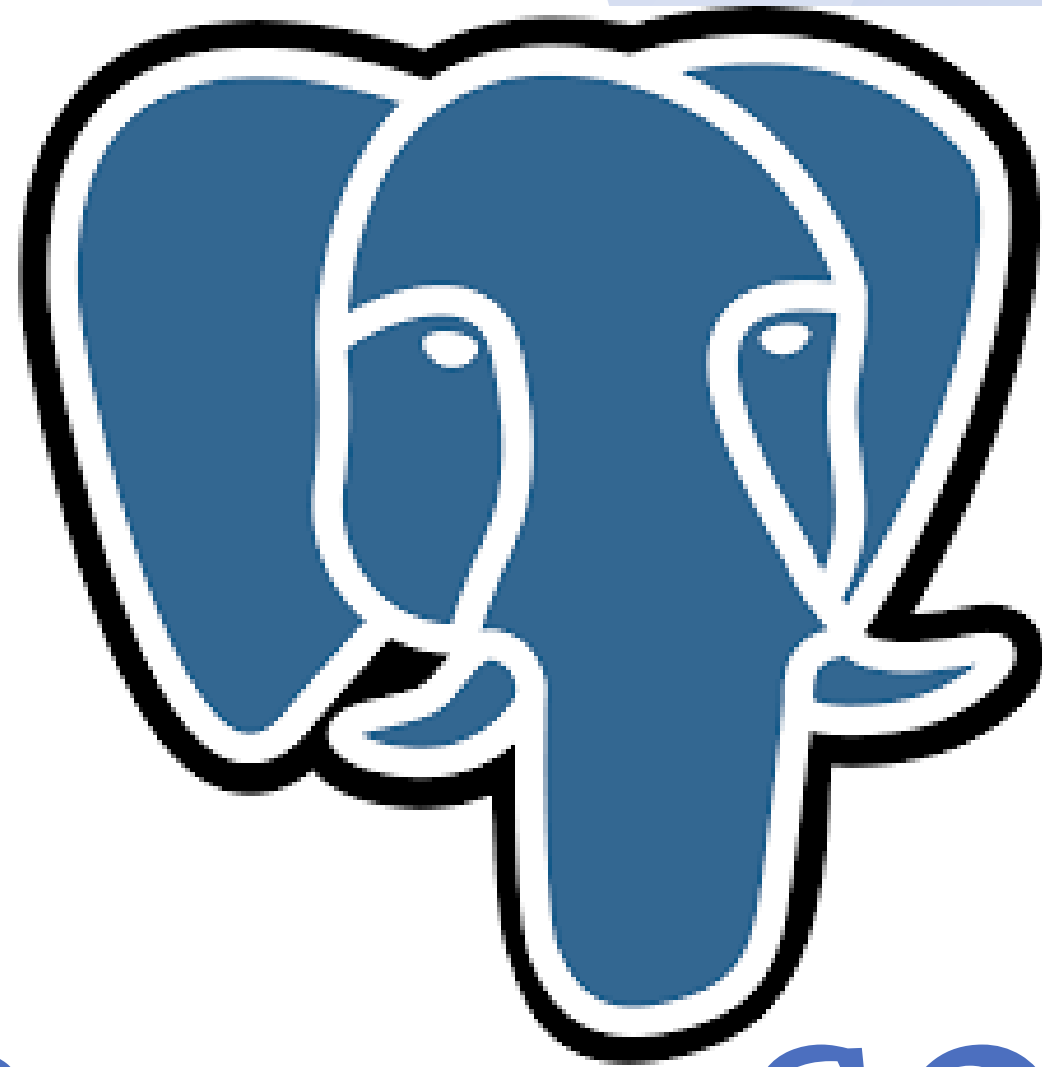


01



PostgreSQL

Backup e PITR

Discentes

- Allan Kaio
- Allany
- Lyziane
- Reinaldo
- Victor Manoel



O que é PostgreSQL?

- É um sistema de gerenciamento de banco de dados relacional de código aberto altamente avançado;
- Oferece suporte a uma ampla variedade de tipos de dados, funcionalidades ACID (Atomicidade, Consistência, Isolamento, Durabilidade), e é conhecido por sua robustez e flexibilidade.
- É amplamente utilizado por empresas e desenvolvedores.



Conceitos de Backup e Recuperação

Backup:

- É o processo de criar cópias de dados que podem ser usadas para restaurar o banco de dados em caso de falha, corrupção de dados ou desastres.
- O objetivo é garantir que, em caso de perda de dados, seja possível restaurar o banco de dados ao seu estado anterior de forma rápida e precisa.

Recuperação:

- Recuperação envolve o uso de backups e logs de transação para restaurar o banco de dados ao seu estado anterior.
- Pode incluir a recuperação completa do banco de dados ou a recuperação até um ponto específico no tempo (PITR).

Tipos de Backup

pg_dump

- É uma ferramenta usada para criar backups lógicos de um banco de dados PostgreSQL.
- **Vantagens:**
 - Flexível, permite backups parciais (por tabela ou por esquema).
 - Fácil de usar e configurar.
- **Desvantagens:**
 - Pode ser lento para bancos de dados muito grandes.
 - Não inclui dados dos logs WAL, então não é adequado para PITR.

pg_basebackup

- É uma ferramenta usada para criar backups físicos de um banco de dados PostgreSQL.
- **Vantagens:**
 - Rápido e eficiente para bancos de dados grandes.
 - Adequado para replicação e PITR, pois inclui os logs WAL.
- **Desvantagens:**
 - Menos flexível em comparação com pg_dump (não permite backups parciais).

PITR (Point In Time Recovery)

- PITR é o processo de restaurar o banco de dados até um ponto específico no tempo usando backups e logs de transação.
- **Como Funciona:**
 - Backup Inicial;
 - Logs WAL;
 - Restaurar e Aplicar Logs.
- **Benefícios do PITR:**
 - Minimização da Perda de Dados;
 - Flexibilidade;
 - Segurança;

07

Caso prático backup

1. Construção do Container

```
docker-compose up --build
```

2. Verificar Tabela

```
docker-compose exec postgres psql -U user -d testdb -c "SELECT * FROM users;"
```

id	name	email
1	Alice	alice@example.com
2	Bob	bob@example.com
3	Charlie	charlie@example.com

(3 rows)

3. Realizar Backup

```
docker-compose exec postgres pg_dump -U user -d testdb -F c -b -v -f /backups/testdb_backup.dump
```

```
✓ backups  
  └─ testdb_backup.dump
```

08

Caso práctico backup

4. Apagar Datos

```
docker-compose exec postgres psql -U user -d testdb -c "DELETE FROM users WHERE name='Alice';"
```

id	name	email
2	Bob	bob@example.com
3	Charlie	charlie@example.com

(2 rows)

5. Restaurar Datos

```
docker-compose exec postgres pg_restore -U user -d testdb -c -v /backups/testdb_backup.dump
```

id	name	email
1	Alice	alice@example.com
2	Bob	bob@example.com
3	Charlie	charlie@example.com

(3 rows)

09

PITR (POINT-IN-TIME RECOVERY)



Obrigado(a) pela atenção!