**📘 Objetivo**

Probar que **Azure PostgreSQL Flexible Server** permite restaurar una base de datos a un punto en el tiempo con un RPO máximo de 5 minutos, como se indica en la documentación oficial.

✅ **Referencia oficial**:  
https://learn.microsoft.com/en-us/azure/postgresql/flexible-server/concepts-backup-restore

**🛠️ Requisitos previos**

1. Tener un servidor PostgreSQL Flexible Server **con backups automáticos habilitados**.
2. Haber realizado **operaciones de escritura recientes** en la base de datos (INSERT, UPDATE).
3. Contar con permisos de **Contributor o superior** en el recurso de Azure.
4. Acceso a **Azure CLI**, **Azure Portal**, o scripts de automatización (opcional: Azure DevOps).

**🧪 Estrategia para probar el RPO de 5 minutos**

**1. 🕐 Generar cambios en la base de datos**

* Conéctate a tu servidor con psql, DBeaver u otro cliente y ejecuta:

sql

CREATE TABLE rpo\_test (

id SERIAL PRIMARY KEY,

info TEXT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

sql

INSERT INTO rpo\_test (info) VALUES ('Prueba RPO - 5 minutos');

📌 Marca el **timestamp exacto del cambio** (por ejemplo, 2025-07-31 12:50:00).

**2. 🛑 Esperar 5–10 minutos**

Esto garantiza que el punto de restauración esté disponible en la ventana de recuperación de 5 minutos.

**3. 🔄 Restaurar la base de datos a un punto en el tiempo**

Usa el portal o Azure CLI para hacer una restauración:

**Opción A: Azure Portal**

1. Ve a tu servidor de PostgreSQL Flexible Server.
2. Haz clic en **Restore**.
3. Selecciona:
   * **Nuevo nombre de servidor** (ej: test-rpo-server)
   * **Punto en el tiempo** justo antes del INSERT (por ejemplo: 12:49:30).
4. Ejecuta la restauración.

**Opción B: Azure CLI**

bash

az postgres flexible-server restore \

--name test-rpo-server \

--source-server <nombre\_original> \

--restore-time "2025-07-31T12:49:30Z" \

--resource-group <tu\_rg> \

--location <ubicacion>

**4. 🔍 Verificar resultados**

* Conéctate al nuevo servidor restaurado (test-rpo-server).
* Ejecuta:

sql

SELECT \* FROM rpo\_test;

**Resultados esperados:**

* ✅ Si **no aparece** el registro de la prueba, el RPO de 5 minutos se cumple.
* ❌ Si **aparece el registro**, repite la restauración a un tiempo ligeramente anterior.

**🧾 Documentación sugerida**

Incluye los siguientes puntos:

1. 📅 Fecha y hora de la prueba.
2. 🧪 Operación realizada y timestamp.
3. 🛠️ Tiempo de restauración solicitado.
4. 📊 Resultado de la verificación.
5. 📉 Conclusión sobre el cumplimiento del RPO.

**📦 Bonus: Automatización para pruebas frecuentes**

Puedes usar Azure Functions o Azure DevOps para:

* Insertar un dato con timestamp cada hora.
* Programar una restauración automática cada día.
* Verificar automáticamente si el dato existe en la restauración.

**1. Conectarte a PostgreSQL Flexible Server desde la CLI**

**🔹 Requisitos:**

* Tener instalado **psql** (cliente de PostgreSQL).
* Conocer:
  + El **nombre del servidor**.
  + El **nombre de la base de datos**.
  + El **usuario administrador** (tiene formato usuario@servidor).
  + La **contraseña**.
  + Que el servidor **permita tu IP pública** (puedes agregarla desde el portal).

**🔹 Comando para conectarte:**

bash

psql "host=<nombre-del-servidor>.postgres.database.azure.com port=5432 dbname=<nombre-de-la-bd> user=<usuario>@<nombre-del-servidor> password=<tu-contraseña> sslmode=require"

**🔸 Ejemplo:**

bash

psql "host=mi-servidor.postgres.database.azure.com port=5432 dbname=postgres user=adminuser@mi-servidor password=MiClave123 sslmode=require"

📌 *Tip:* Si quieres evitar escribir la contraseña cada vez, puedes exportarla:

bash

export PGPASSWORD='MiClave123'

Y luego conectarte con:

bash

psql -h mi-servidor.postgres.database.azure.com -U adminuser@mi-servidor -d postgres -p 5432 -v sslmode=require

**🧩 2. Crear tabla y datos de prueba para RPO**

sql

CREATE TABLE rpo\_test (

id SERIAL PRIMARY KEY,

info TEXT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

sql

INSERT INTO rpo\_test (info) VALUES ('Registro de prueba para RPO');

📌 Guarda el timestamp del insert con:

sql

SELECT NOW();

**🔄 3. Restaurar manualmente desde el portal de Azure**

**Pasos para hacer el restore manual:**

1. Ingresa al **Portal de Azure**: https://portal.azure.com/
2. Ve a **"Azure Database for PostgreSQL Flexible Server"** y selecciona tu servidor.
3. En el panel izquierdo, haz clic en **"Restore"**.
4. Llena los siguientes campos:
   * **Target server name**: el nuevo nombre del servidor (ej. mi-servidor-rpo)
   * **Restore point in time**: escoge un momento **unos segundos antes del INSERT**
   * **Location**: debe ser la misma ubicación de tu servidor original.
5. Haz clic en **"Review + Restore"** y confirma.

⏱️ El proceso puede tardar algunos minutos.

**✅ 4. Verifica la restauración**

Cuando el servidor nuevo esté listo:

1. Conéctate con el mismo comando psql, usando el nuevo nombre de servidor.
2. Verifica si existe la tabla y si contiene datos:

sql

SELECT \* FROM rpo\_test;

* Si **no está el registro**, el RPO de 5 minutos se cumplió.
* Si **sí está**, intenta con un punto de restauración más cercano al INSERT.

**🧾 1. Script Bash (rpo\_test.sh) – Inserta datos y guarda timestamp**

Este script:

* Se conecta a tu servidor PostgreSQL Flexible Server
* Crea una tabla si no existe
* Inserta un registro con marca de tiempo
* Muestra el timestamp exacto del INSERT para usarlo en la restauración

**🔹 rpo\_test.sh**

bash

#!/bin/bash

# Configura tus variables

SERVER="mi-servidor.postgres.database.azure.com"

DB="postgres"

USER="adminuser@mi-servidor"

PASSWORD="MiClave123"

SSL="require"

# Exportar la contraseña para usar con psql

export PGPASSWORD="$PASSWORD"

# Crear tabla si no existe e insertar registro

psql -h "$SERVER" -U "$USER" -d "$DB" -p 5432 -v sslmode="$SSL" <<EOF

CREATE TABLE IF NOT EXISTS rpo\_test (

id SERIAL PRIMARY KEY,

info TEXT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

INSERT INTO rpo\_test (info) VALUES ('Registro de prueba RPO');

SELECT NOW() AS "Timestamp\_del\_Insert";

EOF

✅ Guarda este archivo como rpo\_test.sh y ejecútalo con bash rpo\_test.sh.

**🧾 2. Script SQL para verificación (verificar\_restore.sql)**

Este script sirve para ejecutar en el servidor restaurado y verificar si el registro insertado aparece o no:

**🔹 verificar\_restore.sql**

sql

\c postgres

SELECT \* FROM rpo\_test ORDER BY created\_at DESC LIMIT 5;

📌 Si no aparece ningún registro con la marca de tiempo original, el **RPO de 5 minutos se cumple**.

**🚀 Extra: Script para conectarse al servidor restaurado**

Cuando ya se haya creado el servidor restaurado (ej. mi-servidor-rpo), puedes usar este comando en bash:

bash

PGPASSWORD="MiClave123" psql \

-h mi-servidor-rpo.postgres.database.azure.com \

-U adminuser@mi-servidor-rpo \

-d postgres \

-p 5432 \

-v sslmode=require \

-f verificar\_restore.sql

**Pasos para conectarte a Azure PostgreSQL Flexible Server con DBeaver**

**✅ Paso 1: Instalar DBeaver**

* Descarga DBeaver Community Edition:  
  🔗 https://dbeaver.io/download/

Instálalo como cualquier otra aplicación (compatible con Windows, macOS y Linux).

**✅ Paso 2: Crear una conexión nueva**

1. Abre DBeaver
2. Haz clic en **"Archivo" > "Nueva conexión"** (o el ícono de conexión en la parte superior)
3. Selecciona **PostgreSQL** y haz clic en **Siguiente**

**✅ Paso 3: Completar los datos de conexión**

En la pantalla de conexión, llena los siguientes campos:

| **Campo** | **Valor** |
| --- | --- |
| **Host** | mi-servidor.postgres.database.azure.com |
| **Port** | 5432 |
| **Database** | postgres (o el nombre de tu base de datos si ya tienes otra) |
| **Username** | usuario@mi-servidor (importante: incluye el @servidor) |
| **Password** | Tu contraseña de administrador |

**✅ Paso 4: Configurar conexión SSL**

1. Haz clic en la pestaña **"SSL"**
2. Marca la casilla **"Use SSL"**
3. En el campo **SSL mode**, selecciona: require
4. Deja los demás campos en blanco

✅ Así Azure acepta la conexión sin necesidad de certificados locales.

**✅ Paso 5: Probar y guardar**

1. Haz clic en **"Test Connection"**
2. Si todo está bien, verás un mensaje de conexión exitosa
3. Haz clic en **"Finalizar"**

**🗃️ Crear una nueva base de datos (opcional)**

Una vez conectado:

1. En el panel izquierdo, haz clic derecho sobre el servidor PostgreSQL
2. Selecciona **SQL Editor > Nueva pestaña SQL**
3. Escribe el siguiente comando para crear una base nueva:

sql

CopiarEditar

CREATE DATABASE rpo\_pruebas;

1. Ejecuta con el botón ▶️ o Ctrl+Enter
2. Luego, en el panel izquierdo, haz clic derecho en el servidor > **"Conectar a base de datos..."**
3. Escribe rpo\_pruebas y conéctate a ella para realizar tus pruebas de RPO.

**✅ Resumen rápido**

* Puedes usar la base postgres sin problema
* Para mantener tus pruebas organizadas, puedes crear una base llamada rpo\_pruebas
* No necesitas certificados para Azure si configuras SSL como require