

Crear grupo de recursos y base de datos

Screenshot of the Microsoft Azure 'Create resource group' wizard.

Step 1: Basic Information

Datos básicos

Grupo de recursos - Contenedor que incluye los recursos relacionados para una solución de Azure. El grupo de recursos puede contener todos los recursos de la solución o solamente los recursos que quiere administrar en grupo. Debe decidir cómo quiere asignar los recursos a los grupos de recursos según lo que resulte más pertinente para su organización. [Más información](#)

Suscripción *

Nombre del grupo de recursos *

Región *

[Anterior](#) [Siguiente](#) [Revisar y crear](#)

Screenshot of the Microsoft Azure 'Create flexible server for PostgreSQL' wizard.

Step 2: Basic Configuration

Básico

Detalles del producto

Azure Database para PostgreSQL de Microsoft
Términos de uso [?](#) | Directiva de privacidad [?](#)

Básico (Cambiar)

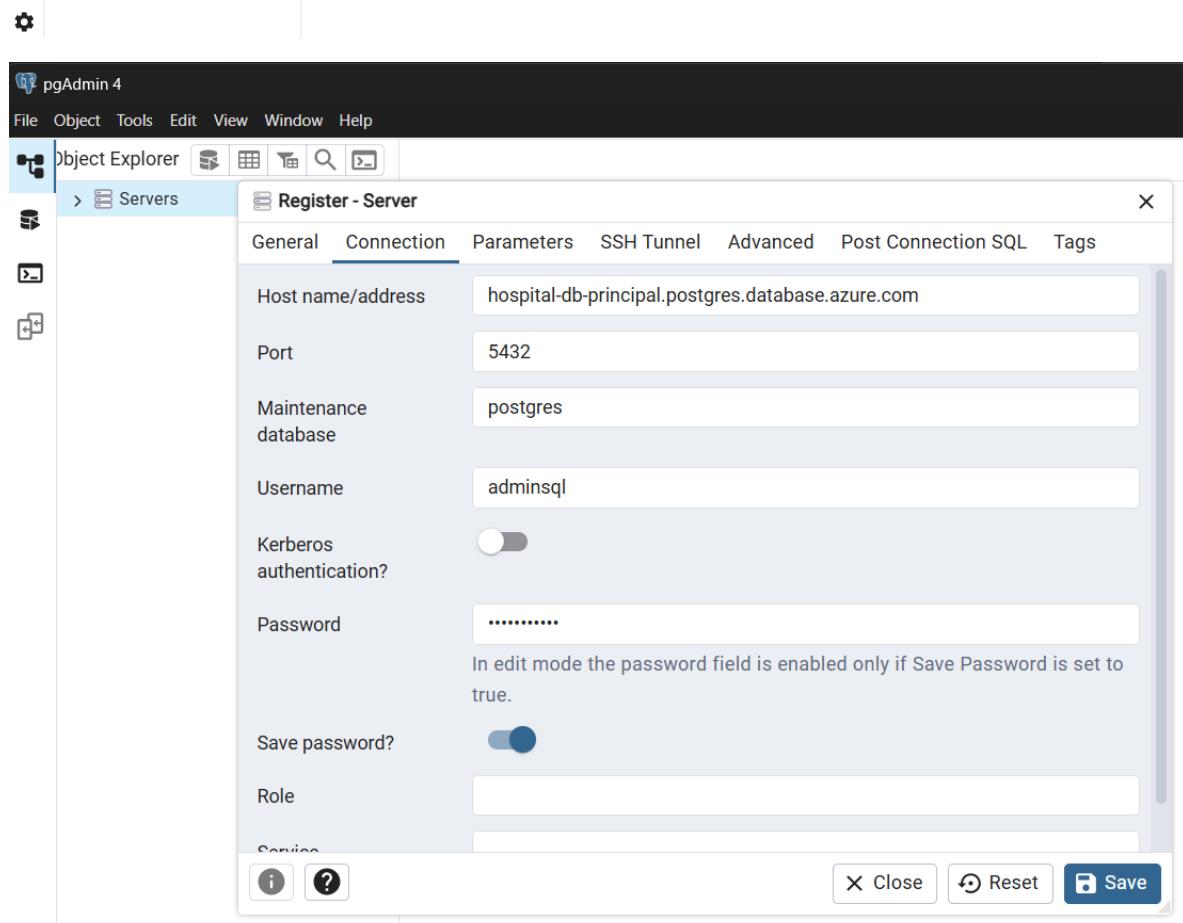
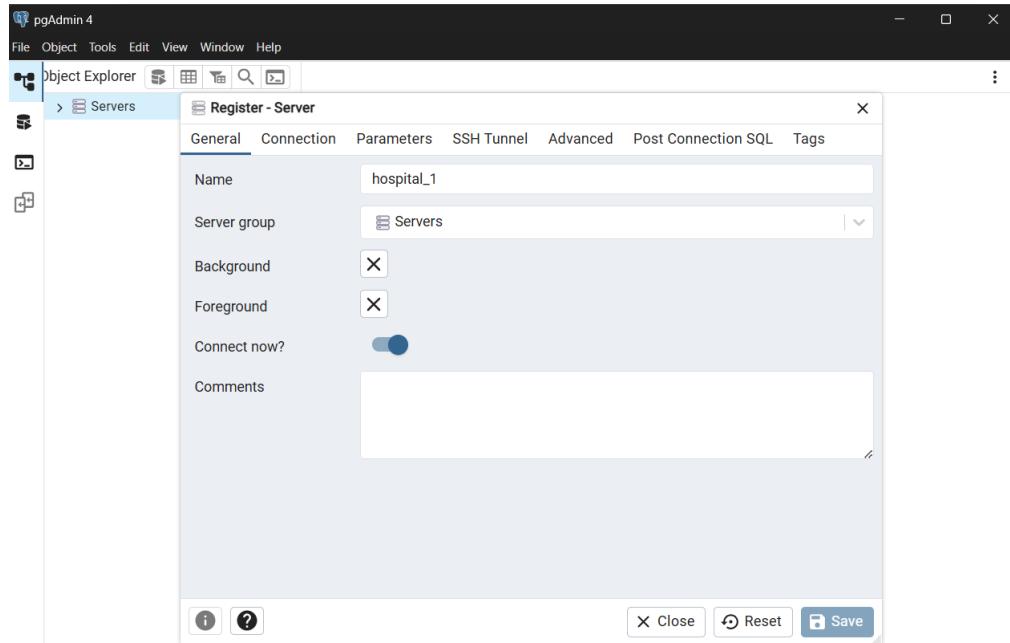
Suscripción	Azure for Students
Grupo de recursos	rg-hospital-bd
Nombre del servidor	hospital-db-principal
Inicio de sesión del administrador	admin\$ql
Ubicación	Brazil South
Zona de disponibilidad	Sin preferencias
Alta disponibilidad	No habilitado
Versión de PostgreSQL	17
Proceso y almacenamiento	Con capacidad de ráfaga, B2s, 2 núcleos virtuales, 4 GiB de RAM 32 almacenamiento GiB, P4 (120 IOPS) 7 días(s)
Periodo de retención de la copia de seguridad (en días)	No habilitado
Crecimiento automático del almacenamiento	No habilitado
Redundancia geográfica	No habilitado

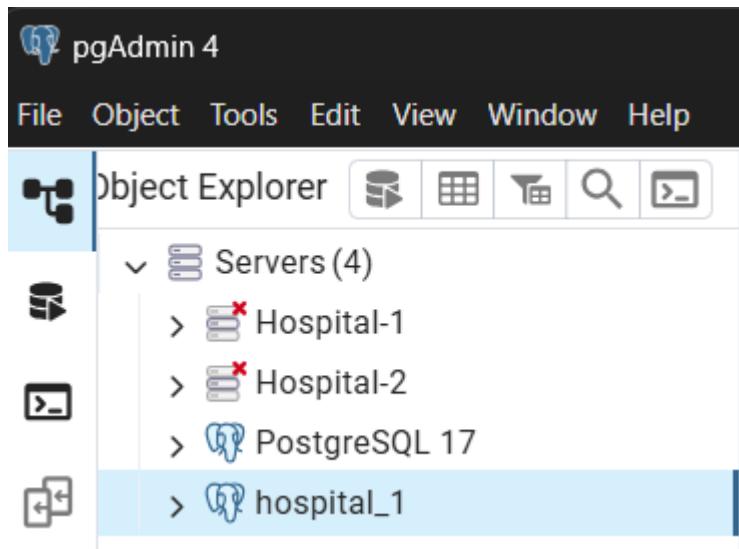
Costes estimados

Compute	USD 102.20/mes
Standard_B2s (2 Núcleos virtuales)	102.20
Almacenamiento	USD 7.01/mes
32 GiB (USD 0.22 por GiB)	32 x 0.22
Ancho de banda	
La transferencia de datos salientes entre servicios de distintas regiones conllevará cargos adicionales. Cualquier transferencia de datos entrantes es gratuita. Más información	
Total estimado	USD 109.21/mes

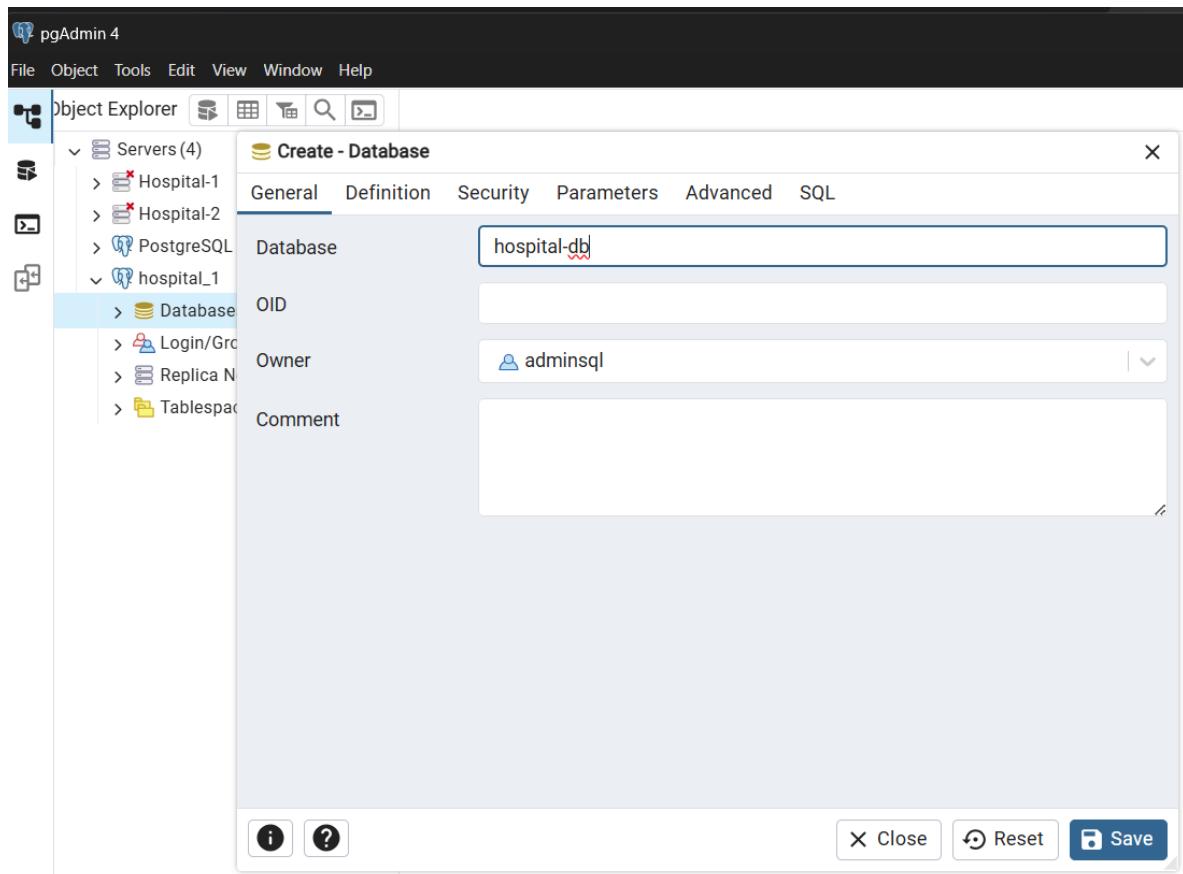
[Crear](#) [< Anterior](#) [Descargar una plantilla para la automatización](#)

Registro del servidor PostgreSQL





Ejercicio 1:



[Query](#) [Query History](#)

```
1  -- Tabla médicos
2  CREATE TABLE medico (
3      id SERIAL PRIMARY KEY,
4      nombre VARCHAR(100) NOT NULL,
5      especialidad VARCHAR(50),
6      telefono VARCHAR(20),
7      fecha_registro TIMESTAMP DEFAULT NOW()
8  );
9
10 -- Tabla pacientes
11 CREATE TABLE paciente (
12     id SERIAL PRIMARY KEY,
13     nombre VARCHAR(100) NOT NULL,
14     fecha_nacimiento DATE,
15     telefono VARCHAR(20),
16     direccion VARCHAR(200),
17     fecha_registro TIMESTAMP DEFAULT NOW()
18 );
19
20 -- Tabla consultas
21 CREATE TABLE consulta (
22     id SERIAL PRIMARY KEY,
23     medico_id INTEGER REFERENCES medico(id),
24     paciente_id INTEGER REFERENCES paciente(id),
25     fecha_consulta TIMESTAMP DEFAULT NOW(),
26     diagnostico TEXT,
27     tratamiento TEXT,
28     costo DECIMAL(10,2)
29 );
```

[Data Output](#) [Messages](#) [Notifications](#)

INSERT 0 3

Query returned successfully in 213 msec.

Query Query History

```
1 -- 2. Crear tabla de prueba con timestamps precisos
2 CREATE TABLE prueba_replicacion (
3     id SERIAL PRIMARY KEY,
4     accion VARCHAR(100),
5     timestamp_escritura TIMESTAMP(6) DEFAULT CLOCK_TIMESTAMP(),
6     zona_escritura VARCHAR(50)
7 );
8
9 -- 3. Insertar datos y medir el tiempo
10 -- Esto se replica SÍNCRONAMENTE a la standby
11 INSERT INTO prueba_replicacion (accion, zona_escritura)
12 VALUES ('Escritura en Primary', current_setting('cluster_name'));
13
14 -- 4. Ver el WAL (Write-Ahead Log) que se está replicando
15 SELECT
16     pg_current_wal_lsn() as posicion_actual,
17     pg_wal_lsn_diff(pg_current_wal_lsn(), '0/0') / 1024 / 1024 as mb_escritos;
18
19 -- 5. Ver lag de replicación (debe ser casi 0)
20 SELECT
21     CASE
22         WHEN pg_last_wal_receive_lsn() = pg_last_wal_replay_lsn()
23             THEN 'Replicación al día'
24         ELSE 'Replicación con retraso'
25     END as estado_replicacion,
26     pg_wal_lsn_diff(pg_last_wal_receive_lsn(), pg_last_wal_replay_lsn()) as bytes_pendientes;
```

Data Output Messages Notifications

Showing rows: 1 to 1

	estado_replicacion	bytes_pendientes
1	Replicación con retraso	[null]

[Query](#) [Query History](#)

```
1 -- 6. Crear procedimiento que simula carga hospitalaria
2 CREATE OR REPLACE FUNCTION simular_consultas(num_consultas INT)
3 RETURNS TABLE(consulta_id INT, tiempo_ms NUMERIC) AS $$ 
4 DECLARE
5     inicio TIMESTAMP;
6     fin TIMESTAMP;
7     nuevo_id INT;
8    BEGIN
9        FOR i IN 1..num_consultas LOOP
10            inicio := CLOCK_TIMESTAMP();
11
12            INSERT INTO consulta (medico_id, paciente_id, diagnostico, tratamiento, costo)
13            VALUES (
14                (RANDOM() * 2 + 1)::INT, -- Médico aleatorio 1-3
15                (RANDOM() * 2 + 1)::INT, -- Paciente aleatorio 1-3
16                'Consulta de prueba ' || i,
17                'Tratamiento ' || i,
18                (RANDOM() * 100 + 20)::NUMERIC(10,2)
19            )
20            RETURNING id INTO nuevo_id;
21
22            fin := CLOCK_TIMESTAMP();
23
24            consulta_id := nuevo_id;
25            tiempo_ms := EXTRACT(EPOCH FROM (fin - inicio)) * 1000;
26            RETURN NEXT;
27        END LOOP;
28    END;
29 $$ LANGUAGE plpgsql;
```

[Data Output](#) [Messages](#) [Notifications](#)

CREATE FUNCTION

Query returned successfully in 416 msec.

Query Query History

```
1 -- 7. Ejecutar simulación (verás que cada escritura espera la confirmación del standby)
2 SELECT * FROM simular_consultas(10);
```

Data Output Messages Notifications

Showing rows: 1 to 10

	consulta_id integer	tiempo_ms numeric
1	14	0.250000
2	15	0.078000
3	16	0.062000
4	17	0.663000
5	18	0.122000
6	19	0.062000
7	20	0.025000
8	21	0.022000
9	22	0.022000
10	23	0.022000

Query Query History

```
1 -- 8. ANTES del failover - Insertar registro marcado
2 INSERT INTO prueba_replicacion (accion, zona_escritura)
3 VALUES ('ANTES del Failover', inet_server_addr()::TEXT);
4
5 -- Anota la hora exacta
6 SELECT NOW(), pg_current_wal_lsn();
```

Data Output Messages Notifications

	now timestamp with time zone	pg_current_wal_lsn pg_lsn
1	2025-12-02 16:01:05.220516+00	0/D01BE40

alta

Guardar Descartar Commutación por error forzada Commutación por error planeada Comentarios Preguntas más frecuentes

alta

Configuración Proceso y almacenamiento Bases de datos Alta disponibilidad Retención a largo plazo (Copias de seguridad almacenadas) Rendimiento inteligente Información de rendimiento de consultas

Modo de alta disponibilidad Con redundancia de zona

Resistencia zonal Deshabilitado (Acuerdo de Nivel de Servicio del 99.9 %) Habilitado (Acuerdo de Nivel de Servicio del 99.99 %)

Estado de alta disponibilidad Correcto

Nombre del servidor principal (lectura/escritura) hospital-db-principal.postgres.database.azure.com

Zona de disponibilidad principal 1

Zona de disponibilidad en espera 2

Commutación por error forzada

¿Quiere continuar?

Iniciar commutación por error forzada Cancelar

✓ La operación de commutación por error forzada se ha completado correctamente

La commutación por error se ha realizado correctamente en el servidor en espera, que ahora es su servidor principal. Para reanudar las operaciones, vuelva a conectarse al servidor de bases de datos hospital-db-principal.

hace unos segundos

Query Query History

```
1
2 -- 9. DESPUÉS del failover - Reconnecta y verifica
3 -- Deberías reconectarte automáticamente
4
5 INSERT INTO prueba_replicacion (accion, zona_escritura)
6 VALUES ('DESPUÉS del Failover', inet_server_addr()::TEXT);
7
8 -- 10. Ver la evidencia completa
9 SELECT
10   id,
11   accion,
12   timestamp_escritura,
13   zona_escritura,
14   timestamp_escritura - LAG(timestamp_escritura) OVER (ORDER BY id) as tiempo_entre_escrituras;
15 FROM prueba_replicacion
16 ORDER BY id DESC
17 LIMIT 10;
```

Data Output Messages Notifications

Showing rows: 1 to 3 Page No:

	id [PK] integer <input type="button" value="Edit"/>	accion character varying (100) <input type="button" value="Edit"/>	timestamp_escritura timestamp without time zone (6) <input type="button" value="Edit"/>	zona_escritura character varying (50) <input type="button" value="Edit"/>	tiempo_entre_escrituras interval <input type="button" value="Edit"/>
1	34	DESPUÉS del Failover	2025-12-02 16:14:41.181718	10.3.0.5/32	00:13:35.960942
2	2	ANTES del Failover	2025-12-02 16:01:05.220776	10.3.0.4/32	00:03:07.566593
3	1	Escritura en Primary	2025-12-02 15:57:57.654183		[null]

Ejercicio 2:

Query Query History

```
1 -- EN LA INSTANCIA PRINCIPAL (Primary):
2
3 -- 1. Crear función que simula escrituras constantes
4 CREATE OR REPLACE FUNCTION registro_consultas_continuo()
5 RETURNS VOID AS $$
```

6 **BEGIN**

```
7     -- Simular registro de consultas cada segundo
8     INSERT INTO consulta (medico_id, paciente_id, diagnostico, tratamiento, costo)
9     SELECT
10         (RANDOM() * 2 + 1)::INT,
11         (RANDOM() * 2 + 1)::INT,
12         'Consulta automatizada',
13         'Tratamiento estándar',
14         (RANDOM() * 100 + 20)::NUMERIC(10,2)
15     FROM generate_series(1, 5);
16 
```

17 **END;**

```
18 $$ LANGUAGE plpgsql;
```

Data Output Messages Notifications

CREATE FUNCTION

Query returned successfully in 10 secs 587 msec.

Query Query History

```
1 -- 2. Insertar algunas consultas
2 SELECT registro_consultas_continuo();
3
4 -- 3. Verificar que hay datos
5 SELECT COUNT(*) FROM consulta;
6
```

Data Output Messages Notifications



	count
1	28

Puntos de conexión virtuales

Estos dominios lógicos simplifican las interacciones con los servidores subyacentes, garantizando un enrutamiento y una distribución de operaciones eficaces. [Más información](#)

+ Creación de puntos de conexión virtuales

Nombre base de los puntos de cone...	Puntos de conexión virtuales de escritor y lector	Puntos de conexión de destino
No se encontraron puntos de conexión virtuales.		

Réplicas

Las réplicas son copias de su base de datos que residen en un servidor lógico diferente del primario. Las réplicas protegen contra errores regionales o interrupciones prolongadas del centro de datos. [Más información](#)

+ Crear réplica

Nombre	Rol	Proceso y almacenamiento	Ubicación	Estado del servidor
hospital-db-principal.postgres.database.azure.com	Principal	De uso general, 2 núcleos virtuales, 32 GB	Brazil South	Listo

Microsoft Azure Buscar recursos, servicios y documentos (G+) Copiar

Inicio > hospital-db-principal | Replicación >

Incorporación de una réplica de lectura a Azure Database para un servidor flexible PostgreSQL

Microsoft

Básico Redes Seguridad Etiquetas Revisar y crear

Detalles del proyecto

Seleccione la suscripción para administrar recursos implementados y los costes. Use los grupos de recursos como carpetas para organizar y administrar todos los recursos.

Suscripción ⓘ Azure for Students

Grupo de recursos ⓘ rg-hospital-bd Crear nuevo

Detalles del servidor

Especifique la configuración necesaria para este servidor, incluida la selección de una ubicación y la configuración de los recursos de proceso y almacenamiento.

Nombre del servidor principal ⓘ hospital-db-principal

Nombre del servidor * ⓘ hospita-db-replica

Ubicación * ⓘ Brazil South

Proceso y almacenamiento ⓘ

Uso general, D2ds_v5
2 núcleos virtuales, 16 GiB de RAM, 32 GiB de almacenamiento, P4 (120 IOPS)
[Configurar servidor](#)

Zona de disponibilidad ⓘ 1

Costes estimados

Compute USD 175.20/mes

Standard_D2ds_v5 (2 núcleos virtuales, USD 87.60 por núcleo virtual) 2 x 87.60

Almacenamiento USD 7.01/mes

32 GiB (USD 0.22 por GiB) 32 x 0.22

Ancho de banda

La transferencia de datos salientes entre servicios de distintas regiones conllevará cargos adicionales. Cualquier transferencia de datos entrantes es gratuita. [Más información](#)

Total estimado USD 182.21/mes

Los precios son solo estimaciones y es posible que no reflejen el precio final. [Vea la calculadora de precios de Azure.](#) Los cargos finales aparecerán en la moneda local en las vistas de facturación y análisis de costos.

[Revisar y crear](#) [Siguiente: Redes >](#)

Inicio >

hosita-db-replica

Servidor flexible de Azure Database for PostgreSQL

Buscar Conectar desde VS Code Eliminar Promover Restablecer contraseña Restaurar Reiniciar Detener Actualizar CLI / PS Comentarios

Información general

^ Información esencial

Suscripción (move)	: Azure for Students
Id. de suscripción	: 2871f126-5604-43cf-9184-f07a1af0b35
Grupo de recursos (move)	: rg-hospital-bd
Nombre del servidor	: hospital-db-replica
Ubicación	: Brazil South
Estado	: Ready
Creado el	: 2025-12-02 16:23:53.2237770 UTC
Etiquetas (editar)	: Agregar etiquetas
Punto de conexión	: hosita-db-replica.postgres.database.azure.com
Punto de conexión virtual	: No habilitado
Inicio de sesión del administrador	: adminsql
Configuración	: Uso_general_D2ds_v5_2 núcleos virtuales, 8 GiB de RAM 32 almac...
Versión de PostgreSQL	: 17.6
Zona de disponibilidad	: 1
Alta disponibilidad	: No habilitado

Vista JSON

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

Servers (4)

- Hospital-1
- Hospital-2
- PostgreSQL 17
- hospital_1

hospital-db/adminsql@hospital_1*

Register - Server

General Connection Parameters SSH Tunnel Advanced Post Connection SQL Tags

Name: hospital_2

Server group: Servers

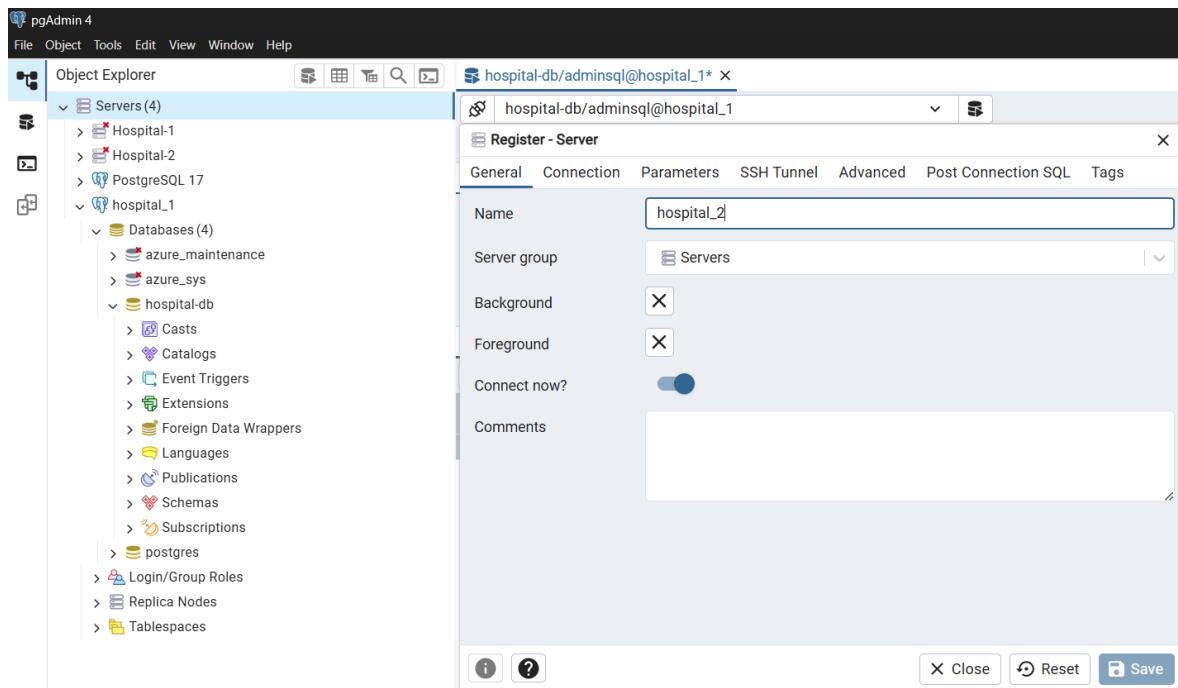
Background:

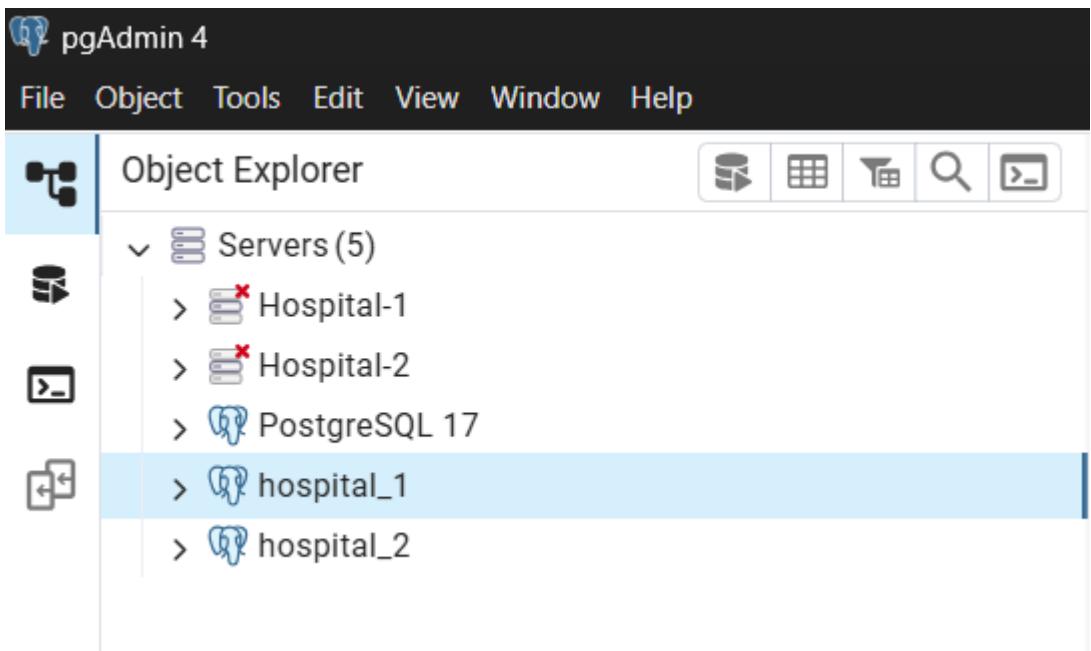
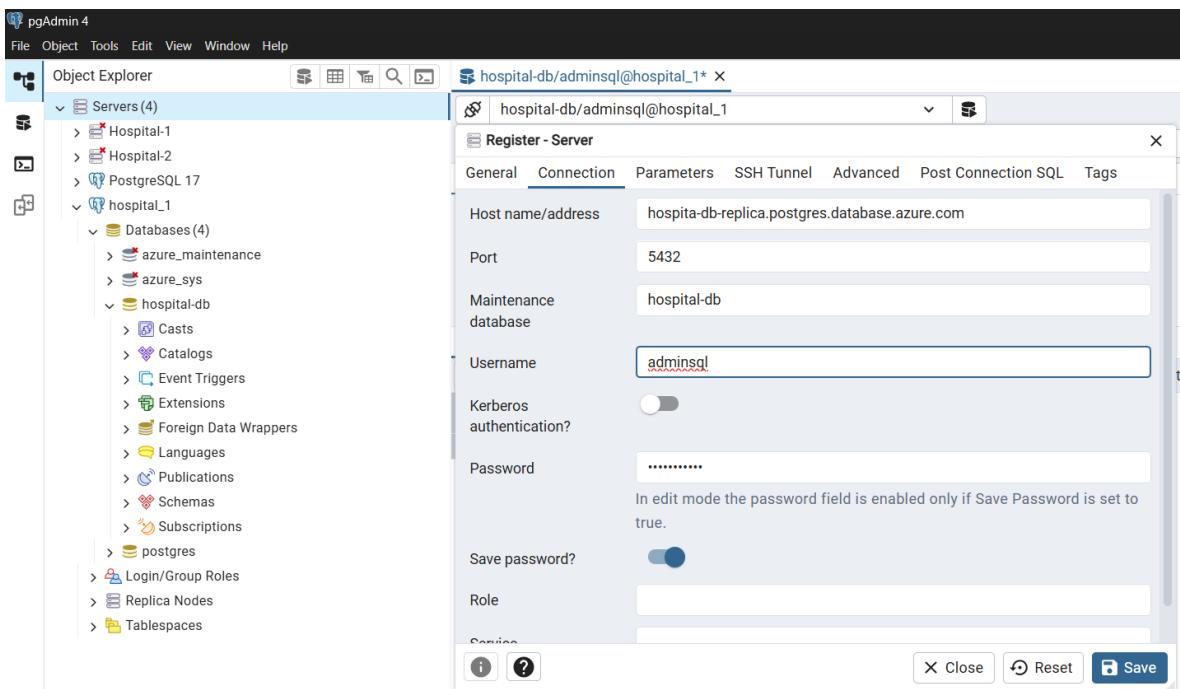
Foreground:

Connect now? (checkbox)

Comments:

Close Reset Save





Query Query History

```
1 -- EN LA READ REPLICA (solo lectura):  
2  
3 -- 4. Verificar que es read-only  
4 SELECT pg_is_in_recovery(); -- TRUE = es réplica
```

Data Output Messages Notifications

	pg_is_in_recovery	boolean
1	true	

Query Query History

```
1 -- 5. Intentar escribir (debe fallar)  
2 INSERT INTO paciente (nombre, fecha_nacimiento)  
3 VALUES ('Prueba', '2000-01-01');
```

Data Output Messages Notifications

ERROR: cannot execute INSERT in a read-only transaction

SQL state: 25006

Query Query History

```
1 -- 6. Queries de lectura pesadas (aquí se descarga el primary)
2 -- Reporte: Consultas por médico
3 SELECT
4     m.nombre as medico,
5     m.especialidad,
6     COUNT(c.id) as total_consultas,
7     AVG(c.costo) as costo_promedio,
8     SUM(c.costo) as ingreso_total
9 FROM medico m
10 LEFT JOIN consulta c ON m.id = c.medico_id
11 GROUP BY m.id, m.nombre, m.especialidad
12 ORDER BY total_consultas DESC;
```

Data Output

Messages Notifications

Showing rows: 1 to 3

	medico character varying (100)	especialidad character varying (50)	total_consultas bigint	costo_promedio numeric	ingreso_total numeric
1	Dr. Luis Rojas	Traumatología	12	72.749166666666667	872.99
2	Dra. Ana Flores	Pediatria	10	59.47300000000000	594.73
3	Dr. Carlos Méndez	Cardiología	6	72.451666666666667	434.71

Query Query History

```
1 -- Reporte: Pacientes frecuentes
2 SELECT
3     p.nombre,
4     p.telefono,
5     COUNT(c.id) as num_consultas,
6     MAX(c.fecha_consulta) as ultima_consulta,
7     SUM(c.costo) as gasto_total
8 FROM paciente p
9 LEFT JOIN consulta c ON p.id = c.paciente_id
10 GROUP BY p.id, p.nombre, p.telefono
11 HAVING COUNT(c.id) > 0
12 ORDER BY num_consultas DESC;
```

Data Output

Messages Notifications

Showing rows: 1 to 3

	nombre character varying (100)	telefono character varying (20)	num_consultas bigint	ultima_consulta timestamp without time zone	gasto_total numeric
1	José Pérez	0992-333-444	15	2025-12-02 16:16:50.091152	1091.41
2	Laura Sánchez	0993-555-666	9	2025-12-02 16:16:50.091152	576.76
3	Maria González	0991-111-222	4	2025-12-02 16:16:50.091152	234.26

Ejercicio 3:

Query Query History

```
1 -- 1. Crear tabla staging para datos externos
2 CREATE TABLE paciente_staging (
3     id INTEGER,
4     nombre VARCHAR(100),
5     fecha_nacimiento DATE,
6     telefono VARCHAR(20),
7     direccion VARCHAR(200),
8     fecha_actualizacion TIMESTAMP DEFAULT NOW()
9 );
10
11 CREATE TABLE consulta_staging (
12     id INTEGER,
13     medico_id INTEGER,
14     paciente_id INTEGER,
15     fecha_consulta TIMESTAMP,
16     diagnostico TEXT,
17     tratamiento TEXT,
18     costo DECIMAL(10,2)
19 );
20
21 -- 2. Simular datos del sistema externo
22 INSERT INTO paciente_staging (id, nombre, telefono, direccion) VALUES
23     (1, 'María González Pérez', '0991-111-999', 'Ambato, Av. Cevallos 123'), -- Actualización
24     (2, 'José Pérez', '0992-333-444', 'Ambato, Ficoa'), -- Sin cambios
25     (4, 'Pedro Morales', '0994-777-888', 'Ambato, Huachi'); -- Nuevo paciente
26
```

Data Output Messages Notifications

```
INSERT 0 2
```

Query returned successfully in 219 msec.

Query Query History

```
1
2 -- 3. Ver estado ANTES del MERGE
3 SELECT 'ANTES DEL MERGE - Pacientes:' as estado;
4 SELECT * FROM paciente ORDER BY id;
5
```

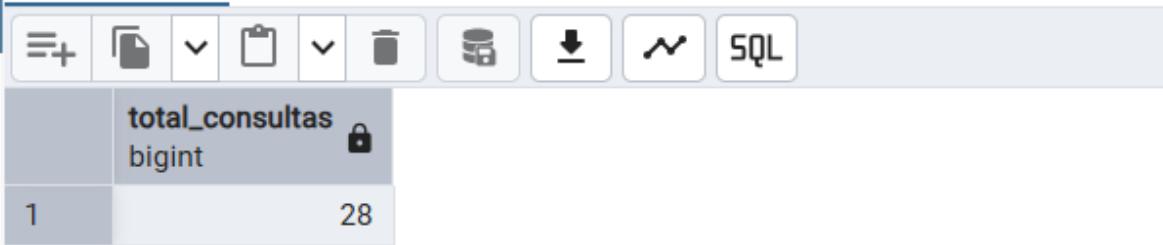
Data Output Messages Notifications

	id [PK] integer	nombre character varying (100)	fecha_nacimiento date	telefono character varying (20)	direccion character varying (200)	fecha_registro timestamp without time zone
1	1	María González	1985-03-15	0991-111-222	Ambato, Av. Cevallos	2025-12-02 15:54:58.519489
2	2	José Pérez	1990-07-22	0992-333-444	Ambato, Ficoa	2025-12-02 15:54:58.519489
3	3	Laura Sánchez	2015-11-10	0993-555-666	Ambato, La Merced	2025-12-02 15:54:58.519489

Query Query History

```
1 SELECT 'ANTES DEL MERGE - Consultas:' as estado;
2 SELECT COUNT(*) as total_consultas FROM consulta;
3
```

Data Output Messages Notifications



Query Query History

```
1 -- 4. Ejecutar MERGE para sincronizar pacientes
2 MERGE INTO paciente AS target
3 USING paciente_staging AS source
4 ON target.id = source.id
5 WHEN MATCHED THEN
6     UPDATE SET
7         nombre = source.nombre,
8         telefono = source.telefono,
9         direccion = source.direccion,
10        fecha_registro = CASE
11            WHEN target.nombre != source.nombre THEN NOW()
12            ELSE target.fecha_registro
13        END
14 WHEN NOT MATCHED THEN
15     INSERT (id, nombre, fecha_nacimiento, telefono, direccion, fecha_registro)
16     VALUES (source.id, source.nombre, source.fecha_nacimiento, source.telefono, source.direccion, NOW());
```

Data Output Messages Notifications

MERGE 3

Query returned successfully in 228 msec.

Query Query History

```

1 -- 5. Ver cambios
2 SELECT
3     id,
4     nombre,
5     telefono,
6     direccion,
7     fecha_registro,
8     CASE
9         WHEN fecha_registro > NOW() - INTERVAL '1 minute' THEN 'ACTUALIZADO'
10        ELSE 'Sin cambios'
11    END as estado
12 FROM paciente
13 ORDER BY id;
14
15

```

Data Output Messages Notifications

	id [PK] integer	nombre character varying (100)	telefono character varying (20)	direccion character varying (200)	fecha_registro timestamp without time zone	estado text
1	1	Maria González Pérez	0991-111-999	Ambato, Av. Cevallos 123	2025-12-02 16:42:16.778858	ACTUALIZADO
2	2	José Pérez	0992-333-444	Ambato, Ficoa	2025-12-02 15:54:58.519489	Sin cambios
3	3	Laura Sánchez	0993-555-666	Ambato, La Merced	2025-12-02 15:54:58.519489	Sin cambios
4	4	Pedro Morales	0994-777-888	Ambato, Huachi	2025-12-02 16:42:16.778858	ACTUALIZADO

Query Query History

```

1 -- 6. MERGE para CONSULTA
2 MERGE INTO consulta AS target
3 USING (
4     SELECT
5         cs.*,
6         p.id as paciente_existe,
7         m.id as medico_existe
8     FROM consulta_staging cs
9     LEFT JOIN paciente p ON cs.paciente_id = p.id
10    LEFT JOIN medico m ON cs.medico_id = m.id
11    WHERE p.id IS NOT NULL AND m.id IS NOT NULL -- Solo registros válidos
12 ) AS source
13 ON target.id = source.id
14 WHEN MATCHED THEN
15     UPDATE SET
16         diagnostico = source.diagnostico,
17         tratamiento = source.tratamiento,
18         costo = source.costo
19 WHEN NOT MATCHED THEN
20     INSERT (medico_id, paciente_id, fecha_consulta, diagnostico, tratamiento, costo)
21     VALUES (source.medico_id, source.paciente_id, source.fecha_consulta, source.diagnostico, source.tratamiento, source.costo)
22

```

Data Output Messages Notifications

```

MERGE 2

Query returned successfully in 197 msec.

```

Query

```

1 -- 7. Verificar resultado final
2
3 SELECT
4     c.id,
5     m.nombre as medico,
6     p.nombre as paciente,
7     c.diagnostico,
8     c.costo,
9     c.fecha_consulta
10    FROM consulta c
11    JOIN medico m ON c.medico_id = m.id
12    JOIN paciente p ON c.paciente_id = p.id
13    ORDER BY c.fecha_consulta DESC
14    LIMIT 10;

```

Data Output

	id integer	medico character varying (100)	paciente character varying (100)	diagnostico text	costo numeric (10,2)	fecha_consulta timestamp without time zone
1	43	Dra. Ana Flores	María González Pérez	Control pediátrico	40.00	[null]
2	42	Dr. Carlos Méndez	Pedro Morales	Chequeo general	30.00	[null]
3	38	Dra. Ana Flores	José Pérez	Consulta automatizada	37.82	2025-12-02 16:16:50.091152
4	41	Dr. Carlos Méndez	Laura Sánchez	Consulta automatizada	58.25	2025-12-02 16:16:50.091152
5	37	Dr. Luis Rojas	José Pérez	Consulta automatizada	79.03	2025-12-02 16:16:50.091152
6	40	Dr. Carlos Méndez	Maria González Pérez	Consulta automatizada	95.72	2025-12-02 16:16:50.091152
7	39	Dra. Ana Flores	José Pérez	Consulta automatizada	23.36	2025-12-02 16:16:50.091152
8	21	Dra. Ana Flores	José Pérez	Consulta de prueba 8	70.10	2025-12-02 15:59:51.592728
9	20	Dr. Luis Rojas	Laura Sánchez	Consulta de prueba 7	91.12	2025-12-02 15:59:51.592728
10	22	Dr. Carlos Méndez	Laura Sánchez	Consulta de prueba 9	70.61	2025-12-02 15:59:51.592728

Showing rows: 1 to 10 | | Page No: 1