

Cisco Azure DB for PostgreSQL Backup & Restore Demo

Sridhar Kothalanka (CSA-Data&AI)

Jan 2020



**Single server backup and restore
using “Locally Redundant” backup option**

Create an Azure DB for PostgreSQL resource

The screenshot shows the Microsoft Azure (Preview) portal interface. The top navigation bar includes the 'Microsoft Azure (Preview)' logo, a search bar ('Search resources, services, and docs (G+/)'), and various account and settings icons. The user is signed in as 'srkothal@microsoft.com'.

The main area displays a 'New' blade with a sidebar on the left containing links like 'Create a resource', 'Home', 'Dashboard', 'All services', 'FAVORITES' (with 'All resources' selected), and others. A search bar at the top of the blade says 'Search the Marketplace'.

The central area has two tabs: 'Azure Marketplace' (selected) and 'Featured'. Under 'Azure Marketplace', there's a list of categories: Get started, Recently created, AI + Machine Learning, Analytics, Blockchain, Compute, Containers, Databases (which is highlighted with a red box), Developer Tools, DevOps, Identity, Integration, Internet of Things, Media, and Mixed Reality. To the right of these categories, under the 'Featured' tab, are several service cards:

- Azure SQL Managed Instance ([Quickstart tutorial](#))
- SQL Database ([Quickstart tutorial](#))
- Azure Synapse Analytics (formerly SQL DW) ([Quickstart tutorial](#))
- Azure Database for MariaDB ([Learn more](#))
- Azure Database for MySQL ([Quickstart tutorial](#))
- Azure Database for PostgreSQL ([Quickstart tutorial](#)) (which is highlighted with a red box)
- Azure Cosmos DB ([Quickstart tutorial](#))

Pick the locally redundant server option

Microsoft Azure (Preview) Report a bug Search resources, services, and docs (G+ /)

Home > New > Select Azure Database for PostgreSQL deployment option > Single server

Single server

Basics Tags Review + create

Create an Azure Database for PostgreSQL server. [Learn more](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * sridhar's internal subscription

Resource group * Demo

[Create new](#)

Server details

Enter required settings for this server, including picking a location and configuring the compute and storage resources.

Server name * mypostgresserver

Data source * None Backup

Admin username * sridhar

Password * Confirm password *

Location * (US) South Central US

Version * 10

Compute + storage General Purpose 4 vCores, 100 GB storage [Configure server](#)

Click on configure server to choose backup option

Review + create Next : Tags >



Microsoft Azure (Preview) Report a bug Search resources, services, and docs (G+ /)

Home > New > Select Azure Database for PostgreSQL deployment option > Single server > Configure

Configure

Variable IO performance (1-2 vCores) predictable IO performance (2-64 vCores)

Please note that changing to and from the Basic pricing tier or changing the backup redundancy options after server creation is not supported.

Compute Generation - [Learn more about compute generation](#) Gen 5

vCore - [What is a vCore?](#)

Storage (type: General Purpose Storage)

Auto-growth - [Learn More](#) Yes No

Backup Retention Period

Backup Redundancy Options - [Learn more details](#)

Locally Redundant Recover from data loss within region

Geo-Redundant Recover from regional outage or disaster

OK

Cannot be changed once server is created

Note Server name & username

The screenshot shows the Microsoft Azure (Preview) interface. The top navigation bar includes the Microsoft logo, a search bar, and user information (srkothal@microsoft.com). Below the navigation bar, the breadcrumb path indicates the current location: Home > All resources > mypostgresserver. The main content area displays the 'mypostgresserver' resource, which is an Azure Database for PostgreSQL server.

The left sidebar contains a navigation menu with various icons and links, including:

- Home
- All resources
- mypostgresserver
- Search (Ctrl+ /)
- Overview (selected)
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Settings
 - Connection security
 - Connection strings
 - Server parameters
 - Replication
 - Active Directory admin
 - Pricing tier
 - Properties
 - Locks
 - Export template

The main content area has several sections:

- Resource group (change): Demo**
- Server name:** mypostgresserver.postgres.database.azure.com
- Admin username:** sridhar@mypostgresserver
- PostgreSQL version:** 10
- Status:** Available
- Location:** South Central US
- Subscription (change):** sridhar's internal subscription
- Subscription ID:** (not explicitly shown)
- Tags (change):** Click here to add tags
- Performance configuration:** General Purpose, 2 vCore(s), 100 GB
- SSL enforce status:** ENABLED

A yellow callout bubble with the text "Make a note of the server name and the username" points to the Server name field.

At the bottom, there is a chart titled "Resource utilization (mypostgresserver)" showing usage over the last 1 hour. The Y-axis ranges from 1.5% to 4.5%. The X-axis shows time intervals. The chart shows several peaks, notably around 4%, 3.5%, and 3% usage.

Configure the server-level firewall rule

Microsoft Azure (Preview) Search resources, services, and docs (G+) Home > All resources > mypostgresserver - Connection security Save Discard Add client IP

mypostgresserver - Connection security Azure Database for PostgreSQL server

Search (Ctrl+/) Overview Activity log Access control (IAM) Tags Diagnose and solve problems

Settings Connection security Connection strings Server parameters Replication Active Directory admin Pricing tier Properties Locks Export template

Firewall rules

Some network environments may not report the actual public-facing IP address needed to access your server. Contact your network administrator if adding your IP address does not allow access to your server.

Allow access to Azure services ON OFF

Rule name	Start IP address	End IP address	...
ClientIPAddress_2020-1-17_10-56-9	75.8.100.91	75.8.100.91	...

VNET rules [+ Adding existing virtual network](#) [+ Create new virtual network](#)

Rule name	Virtual network	Subnet	Address range	Endpoint status	Resource group	Si
No results						

SSL settings

Enforcing SSL connections on your server may require additional configuration to your applications connecting to the server. Click here to learn more.

Connect to PostgreSQL server using Cloud Shell

The screenshot shows the Microsoft Azure Cloud Shell interface. At the top, there's a navigation bar with 'Microsoft Azure (Preview)', a search bar, and a user profile. A red box highlights the 'New' button icon. Below the navigation bar, the title is 'mypostgresserver - Connection security'. On the left, there's a sidebar with icons for Home, All resources, Overview, and Firewall rules. The main area is a terminal window titled 'PowerShell'. The terminal output shows:

```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

MOTD: Read more about PowerShell in CloudShell: https://aka.ms/pscloudshell/docs

VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
Azure:/
PS Azure:\> psql --host=mypostgresserver.postgres.database.azure.com --port=5432 --username=sridhar@mypostgresserver --dbname=postgres
Password for user sridhar@mypostgresserver:
psql (12.1 (Ubuntu 12.1-1.pgdg16.04+1), server 10.11)
SSL connection (protocol: TLSV1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
Type "help" for help.

postgres=> create database demodb;
ERROR:  database "demodb" already exists
postgres=> \c demodb
psql (12.1 (Ubuntu 12.1-1.pgdg16.04+1), server 10.11)
SSL connection (protocol: TLSV1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
You are now connected to database "demodb" as user "sridhar@mypostgresserver".
demodb=>
```

A yellow callout bubble points to the 'Create a database' command with the text 'Create a database'. Another yellow callout bubble points to the '\c demodb' command with the text 'Connect to the database'. A yellow arrow points from the 'Connect to the database' text to the '\c demodb' command. A yellow speech bubble in the top right corner says 'Connect to the server'.

Connect using pg_admin

1. Launch pg_admin and add a server using the tool menu (Object -> Create -> Server...)
2. Input the server name in the "general" tab
3. Input the connection string, port, database name and username in "Connection" tab

The figure consists of three screenshots illustrating the steps to create a new server in pgAdmin:

- Screenshot 1:** The pgAdmin interface with the "Create" menu open. The "Server..." option is highlighted.
- Screenshot 2:** The "General" tab of the server configuration dialog. The "Name" field is set to "mypostgresserver".
- Screenshot 3:** The "Connection" tab of the server configuration dialog. The fields are filled with the following values:
 - Host name/address: mypostgresserver.postgres.database.azure.com
 - Port: 5432
 - Maintenance database: postgres
 - Username: sridhar@mypostgresserver
 - Role: (empty)
 - Service: (empty)

Open pg_admin client

pgAdmin File Object Tools Help

Browser Dashboard Properties SQL Statistics Dependencies Dependents

Servers (1) mypostgresserver Databases (4) azure_maintenance azure_svs demodb postgres Login/Group Roles Tablespaces

Server sessions

Transactions per second

Tuples in

Tuples out

Block I/O

Server activity

Sessions Locks Prepared Transactions Configuration Search

	PID	Database	User	Application	Client	Backend start	State	Wait event
x	124	postgres	azure_superuser					
x	136	postgres	azure_superuser					
x	144	postgres	azure_superuser					
x	192							

2020-01-17 16:08:10 UTC Activity: CheckpointerMain

The screenshot shows the pgAdmin 4 interface with the 'Dashboard' tab selected. The left sidebar shows a single server 'mypostgresserver' with four databases: 'azure_maintenance', 'azure_svs', 'demodb' (highlighted with a red box), and 'postgres'. The main area contains five cards: 'Server sessions' (line chart for Total, Active, and Idle sessions), 'Transactions per second' (line chart for Transactions, Commits, and Rollbacks), 'Tuples in' (line chart for Inserts, Updates, and Deletes), 'Tuples out' (line chart for Fetched and Returned tuples), and 'Block I/O' (line chart for Reads and Hits). Below these is a 'Server activity' section with tabs for Sessions, Locks, Prepared Transactions, and Configuration, and a search bar. A table at the bottom lists active sessions with columns for PID, Database, User, Application, Client, Backend start, State, and Wait event. The table shows three sessions for the 'postgres' database and 'azure_superuser' user, all in the 'idle' state.

Populate the server with databases and a table

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the 'Servers' tree, with 'mypostgresserver' selected. Under 'mypostgresserver', 'Databases' is expanded, showing 'azure_maintenance', 'azure_sys', and 'demodb'. The 'demodb' database is currently selected. The main window contains a 'Query Editor' tab with the following SQL code:

```
1 select * from pg_catalog.pg_tables where tableowner='sridhar';
2 --create database demodb
3 create table inventory (id integer, name varchar(25), count integer);
4 insert into inventory (id, name, count) value (1, 'banana', 100);
5 insert into inventory (id, name, count) value (2, 'apple', 200);
6 insert into inventory (id, name, count) value (3, 'orange', 300);
7 select * from inventory;
8 select timeofday(); -- Fri Jan 17 23:29:33.095417 2020 UTC
9 create database demodb_2;
10 create database demodb_3;
11
12 --drop database demodb_2;
13 --drop database demodb_3;
```

The 'Data Output' tab at the bottom shows the result of the timestamp query:

timeofday	text
1	Fri Jan 17 23:29:33.095417 2020 UTC

Yellow callout boxes provide the following annotations:

- A callout points to the 'inventory' table creation line: "Create a table called 'inventory' in demodb and populate it with 3 rows".
- A callout points to the timestamp query line: "Make a note of the timestamp. This is to capture a point in time".
- A callout points to the database creation lines: "Create 2 additional databases".
- A callout points to the drop database lines: "Drop the database".

Populate the server with databases and tables

The screenshot shows the pgAdmin interface for managing PostgreSQL databases. The left sidebar displays the 'Servers' tree, with 'mypostgresserver' selected. Under 'mypostgresserver', 'Databases' are listed, including 'demodb_1', 'demodb_2', and 'demodb_3', which are highlighted with a red box. The main area is the 'Query Editor' tab, showing the following SQL code:

```
1 create table inventory1 (id integer, name varchar(25), count integer);
2 insert into inventory1 (id,name,count) values (1,'banana',100);
3 insert into inventory1 (id,name,count) values (2,'apple',200);
4 insert into inventory1 (id,name,count) values (3,'orange',300);
5 select * from inventory1;
6
7 select timeofday();
```

A yellow callout points from the text 'Make a note of the point-in-time when the server was healthy' to the output of the 'timeofday()' function in the 'Data Output' tab, which shows the value 'Sat Jan 18 15:09:27.336847 2020 UTC'. This value is also highlighted with a red box.

Query Editor Query History Scratch Pad

```
1 create table inventory1 (id integer, name varchar(25), count integer);
2 insert into inventory1 (id,name,count) values (1,'banana',100);
3 insert into inventory1 (id,name,count) values (2,'apple',200);
4 insert into inventory1 (id,name,count) values (3,'orange',300);
5 select * from inventory1;
6
7 select timeofday();
```

Make a note of the point-in-time when the server was healthy

timeofday	text
1	Sat Jan 18 15:09:27.336847 2020 UTC

Simulate some unexpected changes

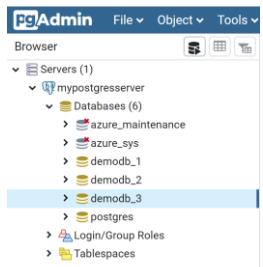
- State of DB at

Sat Jan 18 15:09:27.336847 2020 UTC

- Database: demodb_1
 - Table: inventory1
- Database: demodb_2
 - Table: inventory2
- Database: demodb_3
 - Table: inventory3

15 <code>select * from inventory1;</code>			
Data Output Explain Messages Notifications			
	<code>id</code> integer	<code>name</code> character varying (25)	<code>count</code> integer
1	1	banana	100
2	2	apple	200
3	3	orange	300

15 <code>select * from inventory2;</code>			
Data Output Explain Messages Notifications			
	<code>id</code> integer	<code>name</code> character varying (25)	<code>count</code> integer
1	1	banana	100



15 <code>select * from inventory3;</code>			
Data Output Explain Messages Notifications			
	<code>id</code> integer	<code>name</code> character varying (25)	<code>count</code> integer
1	1	banana	100
2	2	apple	200
3	3	orange	300

- Current state of DB

- Database: demodb_1
 - Table: inventory1
- Database: demodb_2
 - Table: inventory2
- Database: demodb_3
 - Table: inventory3

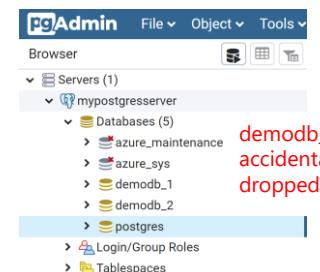
Database is good to use

Inventory table should have only 1 row but has multiple rows

Database got accidentally deleted

15 <code>select * from inventory1;</code>			
Data Output Explain Messages Notifications			
	<code>id</code> integer	<code>name</code> character varying (25)	<code>count</code> integer
1	1	banana	demodb_1
2	2	apple	inventory1
3	3	orange	table is good

18 <code>select * from inventory2;</code>			
Data Output Explain Messages Notifications			
	<code>id</code> integer	<code>name</code> character varying (25)	<code>count</code> integer
1	1	banana	demodb_2
2	2	banana	inventory2
3	3	banana	table has duplicates
4	4	banana	100



Restore the server to a point-in-time

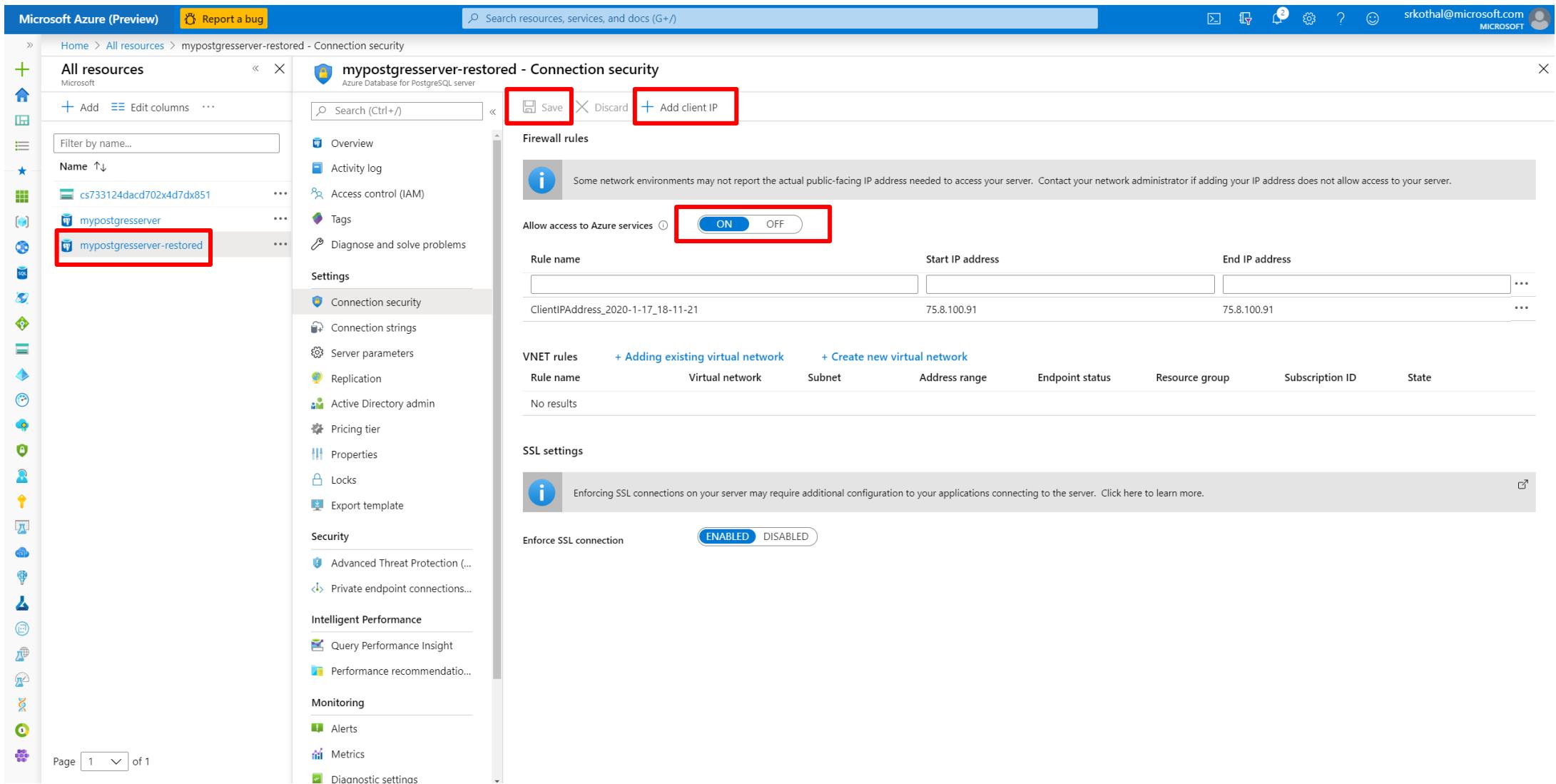
```
az postgres server restore --resource-group demo --name mypostgresserver-restored --restore-point-in-time 2020-01-18T15:09:27Z --source-server mypostgresserver
```

The screenshot shows the Microsoft Azure (Preview) portal interface. At the top, there is a search bar and a user profile. Below it, the 'All resources' section is displayed, with a red box highlighting the refresh button (labeled 3). In the main table, three resources are listed: a storage account ('cloud-shell-storage-southcentralus') and two PostgreSQL servers ('mypostgresserver' and 'mypostgresserver-restored'). The 'mypostgresserver-restored' row is highlighted with a red box and labeled 4. The PowerShell terminal window at the bottom contains the command used to perform the restore.

Name	Type	Resource group	Location	Subscription
cs733124dacd702x4d7dx851	Storage account	cloud-shell-storage-southcentralus	South Central US	sridhar's internal subscription
mypostgresserver	Azure Database for PostgreSQL server	Demo	South Central US	sridhar's internal subscription
mypostgresserver-restored	Azure Database for PostgreSQL server	demo	South Central US	sridhar's internal subscription

```
PS Azure:> az postgres server restore --resource-group demo --name mypostgresserver-restored --restore-point-in-time 2020-01-18T15:09:27Z --source-server mypostgresserver
{
  "administratorLogin": "sridhar",
  "earliestRestoreDate": "2020-01-18T15:46:51.277000+00:00",
  "fullyQualifiedDomainName": "mypostgresserver-restored.postgres.database.azure.com",
  "id": "subscriptions/33124dac-d702-4d7d-851b-0e1159f01295/resourceGroups/demo/providers/Microsoft.DBforPostgreSQL/servers/mypostgresserver-restored",
  "location": "southcentralus",
  "masterServerId": "",
  "name": "mypostgresserver-restored",
  "replicaCapacity": 5,
  "replicationRole": "None",
  "resourceGroup": "demo",
  "sku": {
    "capacity": 2,
    "family": "Gen5",
    "name": "GP_Gen5_2",
    "size": null,
    "tier": "GeneralPurpose"
  },
  "sslEnforcement": "Enabled",
  "storageProfile": {
    "backupRetentionDays": 7,
    "geoRedundantBackup": "Disabled",
    "storageAutoGrow": "Enabled",
    "storageAutoGrow": null
  }
}
```

Firewall settings on the restored database



The screenshot shows the Microsoft Azure (Preview) portal interface for managing a PostgreSQL database. The left sidebar lists various resources, and the main panel displays the 'Connection security' settings for the database 'mypostgresserver-restored'. Key elements include:

- Toolbar:** Includes 'Save' and 'Discard' buttons, both highlighted with red boxes.
- Information Bar:** A message states: "Some network environments may not report the actual public-facing IP address needed to access your server. Contact your network administrator if adding your IP address does not allow access to your server."
- Switch:** 'Allow access to Azure services' is set to **ON**, also highlighted with a red box.
- Table:** Shows a single rule entry: Rule name 'ClientIPAddress_2020-1-17_18-11-21', Start IP address '75.8.100.91', and End IP address '75.8.100.91'.
- VNET rules:** A section for managing virtual network rules, currently showing 'No results'.
- SSL settings:** A note: "Enforcing SSL connections on your server may require additional configuration to your applications connecting to the server. Click here to learn more." The 'Enforce SSL connection' switch is set to **ENABLED**.

Connect to the restored database

The screenshot shows the pgAdmin interface with a successful connection to a restored database. The browser pane on the left lists two servers: 'mypostgresserver' and 'mypostgresserver-restored'. The 'mypostgresserver-restored' server is selected and highlighted with a red box. Its databases are listed: 'azure_maintenance', 'azure_sys', 'demodb', 'demodb_2', 'demodb_3', and 'postgres'. The 'demodb' database is currently selected and highlighted with a blue bar at the bottom of the list. The main pane shows the 'Query Editor' tab with the query 'select * from inventory;'. Below the editor is a 'Data Output' tab displaying the results of the query:

	id integer	name character varying (25)	count integer
1	1	banana	100
2	2	apple	200
3	3	orange	300

Verify data in restored server data is good

Server: mypostgresserver-restored

pgAdmin

File Object Tools Help

Browser

Servers (2)

mypostgresserver

Databases (5)

azure_maintenance

azure_sys

demodb_1

demodb_2

postgres

Login/Group Roles

Tablespaces

mypostgresserver-restored

Databases (6)

azure_maintenance

azure_sys

demodb_1

demodb_2

demodb_3

postgres

Login/Group Roles

Tablespaces

demodb_1/sridhar@mypostgresserver-restored*

Query Editor

```
1 select * from inventory1;
```

Data Output Explain Messages Notifications

	<code>id</code>	<code>name</code>	<code>count</code>
1	1	banana	100
2	2	apple	200
3	3	orange	300

demodb_1

pgAdmin

File Object Tools Help

Browser

Servers (2)

mypostgresserver

Databases (5)

azure_maintenance

azure_sys

demodb_1

demodb_2

postgres

Login/Group Roles

Tablespaces

mypostgresserver-restored

Databases (6)

azure_maintenance

azure_sys

demodb_1

demodb_2

demodb_3

postgres

Login/Group Roles

Tablespaces

demodb_2/sridhar@mypostgresserver-restored*

Query Editor

```
1 select * from inventory2;
```

Data Output Explain Messages Notifications

	<code>id</code>	<code>name</code>	<code>count</code>
1	1	banana	100

demodb_2

pgAdmin

File Object Tools Help

Browser

Servers (2)

mypostgresserver

Databases (5)

azure_maintenance

azure_sys

demodb_1

demodb_2

postgres

Login/Group Roles

Tablespaces

mypostgresserver-restored

Databases (6)

azure_maintenance

azure_sys

demodb_1

demodb_2

demodb_3

postgres

Login/Group Roles

Tablespaces

demodb_3/sridhar@mypostgresserver-restored*

Query Editor

```
1 select * from inventory3;
```

Data Output Explain Messages Notifications

	<code>id</code>	<code>name</code>	<code>count</code>
1	1	banana	100
2	2	apple	200
3	3	orange	300

demodb_3

Restore the inventory2 table to demodb_2

- Export the inventory2 table from demodb_2 database from mypostgresserver-restored server
 - `pg_dump --host=mypostgresserver-restored.postgres.database.azure.com --username=sridhar@mypostgresserver-restored --dbname=demodb_2 --file=demodb_2.sql --table=inventory2;`
- Drop the table “inventory1” from mypostgresserver server
 - `drop inventory2 from demodb_2`
- Import the inventory2 table to original server
 - `psql --file=demodb_2.sql --host=mypostgresserver.database.windows.net --port=5432 --username=sridhar@mypostgresserver --dbname=demodb_2 --table=inventory2;`

```
PS Azure:\> pg_dump --host=mypostgresserver-restored.postgres.database.azure.com --username=sridhar@mypostgresserver-restored --dbname=demodb_2 --file=demodb_2.sql --table=inventory2;
Password:
Azure:/ 
PS Azure:\> ls
clouddrive demodb_2.sql
Azure:/ 
PS Azure:\> psql --host=mypostgresserver.postgres.database.azure.com --port=5432 --username=sridhar@mypostgresserver --dbname=postgres
Password for user sridhar@mypostgresserver:
psql (12.1 (Ubuntu 12.1-1.pgdg16.04+1), server 10.11)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
Type "help" for help.

postgres=> \c demodb_2
psql (12.1 (Ubuntu 12.1-1.pgdg16.04+1), server 10.11)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
You are now connected to database "demodb_2" as user "sridhar@mypostgresserver".
demodb_2=> drop table inventory2;
DROP TABLE
demodb_2=> \q
Azure:/ 
PS Azure:\> psql --file=demodb_2.sql --host=mypostgresserver.database.windows.net --port=5432 --username=sridhar@mypostgresserver --dbname=demodb_2 --table=inventory2;
Password for user sridhar@mypostgresserver:
```

Restore the demodb_3 database

- Export the demodb_3 database from mypostgresserver-restored server
 - `pg_dump --host=mypostgresserver-restored.postgres.database.azure.com --username=sridhar@mypostgresserver-restored --dbname=demodb_3 --file=demodb_3.sql;`
- Create an empty database demodb_3 in mypostgresserver
 - `create database demodb_3;`
- Import the demodb_3 to mypostgresserver server
 - `psql --file=demodb_3.sql --host=mypostgresserver.database.windows.net --port=5432 --username=sridhar@mypostgresserver --dbname=demodb_3;`

```
PS Azure:\> pg_dump --host=mypostgresserver-restored.postgres.database.azure.com --username=sridhar@mypostgresserver-restored --dbname=demodb_3 --file=demodb_3.sql;
Password:
Azure:/
PS Azure:\> ls
clouddrive  demodb_2.sql  demodb_3.sql
Azure:/
PS Azure:\> psql --host=mypostgresserver.postgres.database.azure.com --port=5432 --username=sridhar@mypostgresserver --dbname=postgres;
Password for user sridhar@mypostgresserver:
psql (12.1 (Ubuntu 12.1-1.pgdg16.04+1), server 10.11)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
Type "help" for help.

postgres=> create database demodb_3;
CREATE DATABASE
postgres=> \q
Azure:/
PS Azure:\> psql --file=demodb_3.sql --host=mypostgresserver.database.windows.net --port=5432 --username=sridhar@mypostgresserver --dbname=demodb_3;
Password for user sridhar@mypostgresserver:
```

Verify the data on mypostgresserver after restore

The screenshot shows the pgAdmin interface with the 'demodb_2' database selected. The left sidebar lists servers and databases. The 'Query Editor' tab contains the SQL query: `select * from inventory2;`. The 'Data Output' tab displays the results:

id	name	count
1	banana	100

demodb_2

The screenshot shows the pgAdmin interface with the 'demodb_3' database selected. The left sidebar lists servers and databases. The 'Query Editor' tab contains the SQL query: `select * from inventory3;`. The 'Data Output' tab displays the results:

id	name	count
1	banana	100
2	apple	200
3	orange	300

demodb_3

**Single server backup and restore
using “Geo Redundant” backup option**

Pick the locally redundant server option

Microsoft Azure (Preview) Report a bug Search resources, services, and docs (G+)

Home > New > Select Azure Database for PostgreSQL deployment option > Single server

Single server

Microsoft

Basics Tags Review + create

Create an Azure Database for PostgreSQL server. [Learn more](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * sridhar's internal subscription

Resource group * Demo Create new

Server details

Enter required settings for this server, including picking a location and configuring the compute and storage resources.

Server name * mypostgresserver

Data source * None Backup

Admin username * sridhar

Password * Confirm password *

Location * (US) South Central US

Version * 10

Compute + storage General Purpose 4 vCores, 100 GB storage [Configure server](#)

Review + create Next : Tags >



Microsoft Azure (Preview) Report a bug Search resources, services, and docs (G+)

Home > New > Select Azure Database for PostgreSQL deployment option > Single server > Configure

Configure

Variable IO performance (1-2 vCores) predictable IO performance (2-64 vCores)

Please note that changing to and from the Basic pricing tier or changing the backup redundancy options after server creation is not supported.

Compute Generation - [Learn more about compute generation](#)

Gen 5

vCore - [What is a vCore?](#)

Storage (type: General Purpose Storage)

Auto-growth - [Learn More](#)

Yes No

Backup Retention Period

Backup Redundancy Options - [Learn more details](#)

Locally Redundant Recover from data loss within region

Geo-Redundant Recover from regional outage or disaster

OK

Cannot be changed once server is created

Restore the server to a point-in-time

- Restore the server in the same resource group as the backup
 - az postgres server georestore --resource-group demo --name mypostgresserver-georestored --source-server mypostgresserver --location eastus
- Restore the server in a different resource group from the backup
 - az postgres server georestore --resource-group newresourcegroup --name mypostgresserver-georestored --source-server "/subscriptions/\$<subscription ID>/resourceGroups/\$<resource group ID>/providers/Microsoft.DBforPostgreSQL/servers/mypostgresserver" --location eastus



Q&A

