GW Gen 4 decommissioning

Last updated by | Hamza Aqel | Mar 31, 2023 at 12:42 AM PDT

As part of Gen4 decommissioning project, We will be decommissioning Gateway.PG for following regions . Region list and decomm date is captured below for your reference.

Region	Control Ring	Gateway.PG	Gateway.MySQL	Total	
canadaeast1-a	cr1		1	1	Bl
centralus1-a	cr2		2	2	
eastasia1-a	cr2		2	2	
westeurope1-a	cr2		4	4	
westus2-a	cr1	2	2	4	
koreasouth1-a	cr1		5	5	Server
uksouth1-a	cr1	4	1	5	
koreacentral1-a	cr1		7	7	
southcentralus1-a	cr2		7	7	Server
japaneast1-a	cr2	6	3	9	
northeurope1-a	cr2	5	5	10	
southeastasia1-a	cr2	7	3	10	
eastus1-a	cr2		11	11	
eastus2-a	cr3	6	7	13	

As part of this maintenance, the IP address(es) of your Azure Database for PostgreSQL (Single Server) will change. If you would like to learn more about connectivity architecture for PostgreSQL refer to this documentation Connectivity architecture in Azure Database for PostgreSQL. And take into consideration all the customers have been notified.

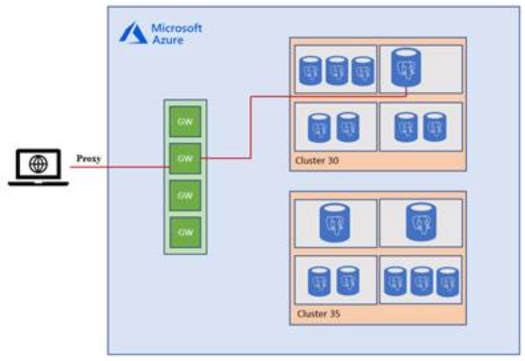
FAQs

- **What are the static IP addresses that will be decommissioned?**
  This decommissioning planned maintenance will cover four regions for now which has the following IP addresses:

refer to our Azure Database for PostgreSQL gateway IP addresses for this list.

GW Region	IP
Central US	13.67.215.62
East US	40.121.158.30
East US 2	52.177.185.181
Japan East	13.78.61.196
Japan West	104.214.148.156
Korea Central	52.231.32.42
Korea South	52.231.151.97
North Central US	23.96.178.199
North Europe	40.113.93.91
South Central US	13.66.62.124
South East Asia	104.43.15.0
West Europe	40.68.37.158
West US	13.66.62.124

When users connect to their servers, the first stop of the connection is to Gateway node, before connection is forwarded to server. We are decommissioning old Gateway rings (not tenant rings where the server is running) please refer to the following graph for more clarification.



This maintenance is simply DNS change, so it is transparent to clients. When the DNS cache is refreshed in the client (it is automatically done by operating systems), all the new connection will connect to the new IP address and all the existing connection will stay connected with no interruption until the decommissioning IP addresses are fully decommissioned, which usually several weeks later. Retry logic is not required for this case, but it is a best practice to have application retry logic configured.

- **How to know if my connections will be impacted?**

  Check if you are using FQDN in the connection string
  Make sure you are using FQDN (xxx.postgres.database.azure.com for example for PG) to connect to the database server and not static IP address to connect. Please review your application code (script or source code) and the setting file (PG bouncer/ProxySQL/K8s/connection poolers, etc)

  I am using a static IP address to connect
  Please review our public documentations for the latest '**Gateway IP addresses**' and ensure that your client application is not using any of the gateway IP addresses that are ***decommissioning***.

- Azure Database for PostgreSQL gateway IP addresses

- **How to test If I will be impacted?**

  You can test by simply PSPing or TCPPing the database server from your client application with port 5432 and ensure that return IP address is not one of the decommissioning IP addresses.

- **Is there any impact for my application connections?**

This maintenance is just a DNS change, so it is transparent to the client. When the DNS cache is refresh in the client (it is automatically done by operation system), all the new connection will connect to the new IP address and all the existing connection will still working fine until the old IP address fully get decommissioned, which usually several weeks later. And the retry logic is not required for this case, but it is good to see the application have retry logic configured. Please either use FQDN to connect to the database server or enable list the new 'Gateway IP addresses' in your application connection string.

Migration will not drop the existing connections. It only makes the new connection requests go to new gateway ring.

- **Can I request for a specific time window for the maintenance?**

As the migration should be transparent and no impact to customer's connectivity, we expect there will be no issue for majority of users. Please review your application proactively and ensure that you either use FQDN to connect to the database server or enable list the new 'Gateway IP addresses' in your application connection string.

- **I am using private link will my connection be impacted.**

In short, this is a gateway hardware decommission and have no relation to private IP addresses, only the public IP addresses mentioned under the decommissioning IP addresses.

Recommended action

- Connect to your Azure Database for PostgreSQL Single server using the DNS record **instead** of the IP address. The DNS record can be found in the Overview blade for you server in the Azure portal.
- If your client connects from behind a firewall, example: when you connect from a corporate network
  Ensure the firewall allows outbound traffic on port 5432 for PostgreSQL to the new IP address(es) documented here
  Azure Database for PostgreSQL gateway IP addresses

How to know if the customer is using a Static IP?

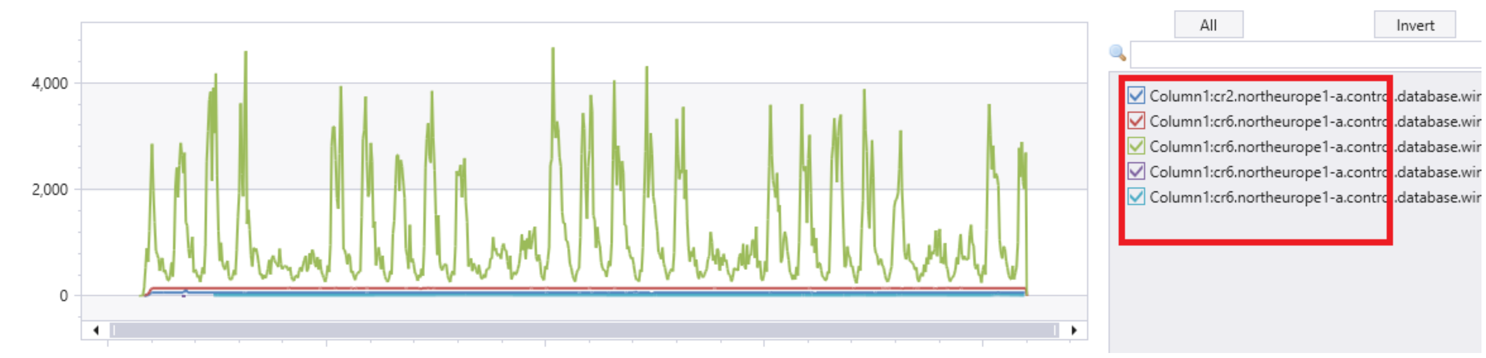
The below is a list of customers who are using the static IPs, so pay attention if you receive any cases related to these subscriptions:

Region	Ring Name	App Type	Customer Subscription	Server Count	Notified (
eastus2-a	cr3.eastus2-a.control.database.windows.net	Gateway.PG	0ac189eb-cc05-44d0-98e8-b3551ebe323b	1	2/13/20
eastus2-a	cr3.eastus2-a.control.database.windows.net	Gateway.PG	17841775-582d-4a8c-b134-fdb2442b8595	1	2/13/20
eastus2-a	cr3.eastus2-a.control.database.windows.net	Gateway.PG	1c888b95-02c1-4acd-aa67-3e2807275603	1	2/13/20
eastus2-a	cr3.eastus2-a.control.database.windows.net	Gateway.PG	40f5d89b-cca4-4f43-850b-46e831ef9da6	1	2/13/20
eastus2-a	cr3.eastus2-a.control.database.windows.net	Gateway.PG	ee691273-18af-4600-bc24-eb6768bf9cfa	2	2/13/20
japaneast1-a	cr2.japaneast1-a.control.database.windows.net	Gateway.PG	50b20f65-fcd6-4917-822c-89625d21450c	1	2/8/20
japaneast1-a	cr2.japaneast1-a.control.database.windows.net	Gateway.PG	96087b4e-f15c-4071-a421-d45eb9fb1101	2	2/8/20
japaneast1-a	cr2.japaneast1-a.control.database.windows.net	Gateway.PG	796fdec4-c9e2-46f4-afe2-e42a89838ab3	1	2/8/20
japaneast1-a	cr2.japaneast1-a.control.database.windows.net	Gateway.PG	800c0ef1-e3ea-465c-a09d-4a3a14a66600	1	2/8/20
northeurope1-a	cr2.northeurope1-a.control.database.windows.net	Gateway.PG	18b0d163-415c-4e22-b92b-61a85276890a	1	2/13/20
northeurope1-a	cr2.northeurope1-a.control.database.windows.net	Gateway.PG	5254b6cf-7fc4-4884-9a33-c09dcd209ee7	1	2/13/20
northeurope1-a	cr2.northeurope1-a.control.database.windows.net	Gateway.PG	532586b2-eac3-4ba5-9fe5-20c477c3f354	1	2/13/20
northeurope1-a	cr2.northeurope1-a.control.database.windows.net	Gateway.PG	89fc81d3-fdea-4a64-969e-1b8b910b6bed	1	2/13/20
northeurope1-a	cr2.northeurope1-a.control.database.windows.net	Gateway.PG	92781abb-bb77-41a1-bca6-902950d46feb	1	2/13/20
southeastasia1-a	cr2.southeastasia1-a.control.database.windows.net	Gateway.PG	515c1927-675c-4154-8c01-a324d0c98b86	1	2/8/20
southeastasia1-a	cr2.southeastasia1-a.control.database.windows.net	Gateway.PG	697dc9ec-d696-4962-94d9-c4e1fadb0455	1	2/8/20
southeastasia1-a	cr2.southeastasia1-a.control.database.windows.net	Gateway.PG	9c313952-cb65-4ed1-aefc-ff459f8926b2	1	2/8/20
southeastasia1-a	cr2.southeastasia1-a.control.database.windows.net	Gateway.PG	515c1927-675c-4154-8c01-a324d0c98b86	1	2/8/20
uksouth1-a	cr1.uksouth1-a.control.database.windows.net	Gateway.PG	ea31d63c-77dc-408b-9618-116f89640793	4	2/8/20
westus2-a	cr1.westus2-a.control.database.windows.net	Gateway.PG	c65f7514-12be-4cf6-a7d7-6294826d1da4	1	2/13/20
westus2-a	cr1.westus2-a.control.database.windows.net	Gateway.PG	fc01f0e2-eb2b-4d2f-a4d0-4041a604a369	1	2/13/20

Or you can use the below Kusto query to check:

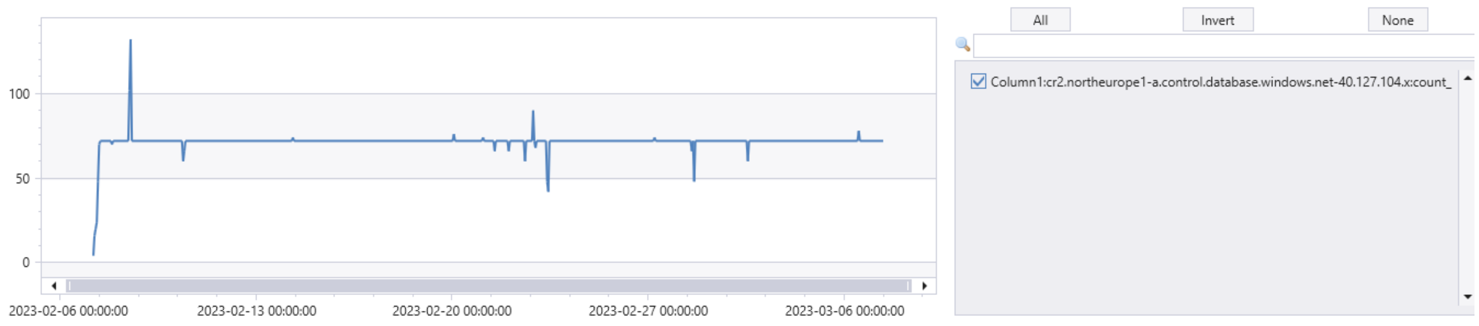
suppose for example a customer is not able to connect to his server, what I can do is to check the connection patterns to our Gateway using the below query:

```
MonLogin
| where AppTypeName == "Gateway.PG" and logical_server_name == "pgservername"
| summarize count() by strcat( ClusterName, "-", peer_address), bin(originalEventTimestamp, 1h)
| render timechart
```



If the connections as in the above example are using the old ring as in the table at the top of this page (cr2) and other rings, it means that the customer is not using a static IP.

if the connections like below:



It means he is using a static IP and we can share the recommended actions mentioned above.