Networking - Firewall rules, VNETs or Private Connectivity

Last updated by | Daniel Valero | Feb 21, 2023 at 9:40 AM PST

Contents

- Introduction
- Getting server networking configuration
  - 1. Is the server set for Public or Private access?
  - 2. Get firewall rules for servers that allow Public Access
    - A- Using orcasbreadth\orcasbreadth-adhoccmsquery.xts vi...
    - B - Using orcasbreadth\orcasbreadth servers.xts
  - 3. Get VNet/subnet, Private DNS Zone and private IP Addr...

Introduction

When it comes to establishing connection to an Azure PostgreSQL flexible server, Customer has two options, Firewall Rules or VNETs/Private Connectivity to enable access to the database.

Public Doc Reference

- https://docs.microsoft.com/en-us/azure/postgresql/flexible-server/concepts-networking
- https://docs.microsoft.com/en-us/azure/postgresql/flexible-server/how-to-manage-virtual-network-portal

Getting server networking configuration

To get network configuration information for the server you can use XTS

1. Is the server set for Public or Private access?

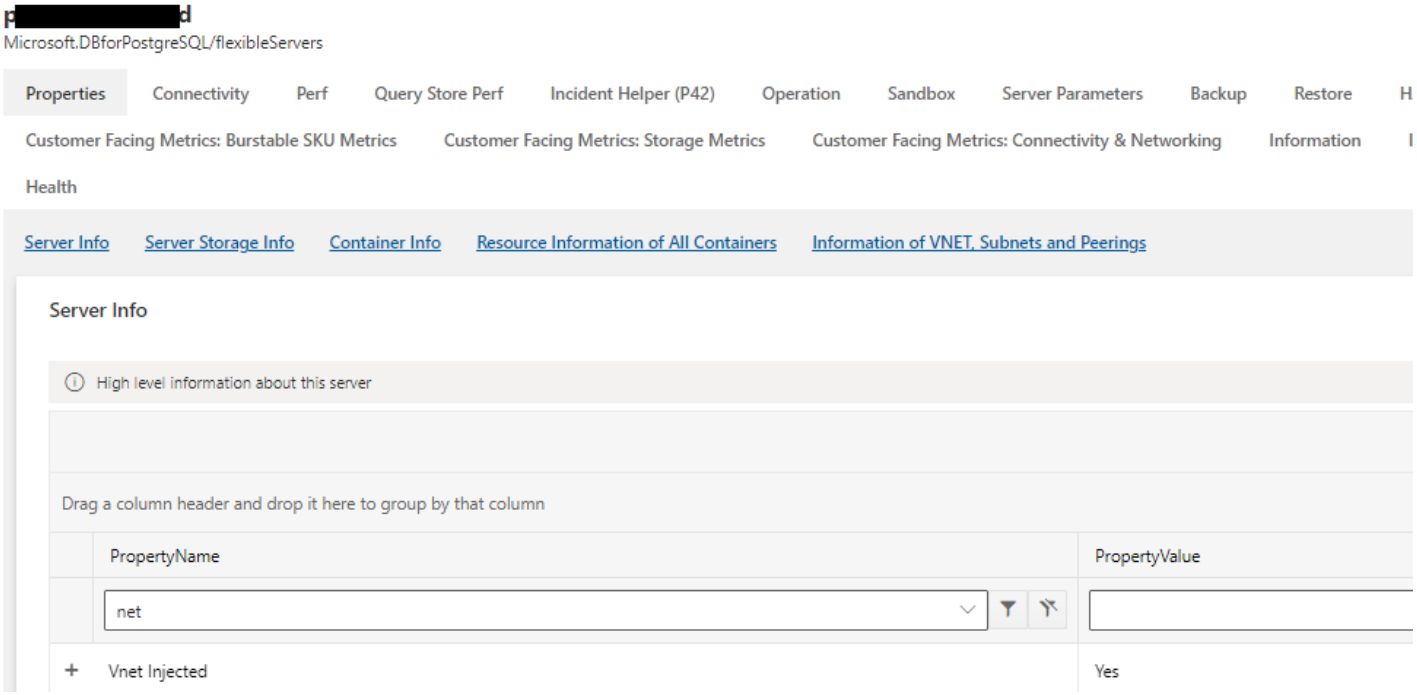
Check the server's connectivity method:

- Using ASC

  Go to the Properties Tab and filter properties with name that includes "net".

  - Vnet Injected at 0 indicates the server is set to Public Access
  - Vnet Injected at Yes indicates the server is set to Private Access (VNET integration). When Vnet Injected at 1, the other parameters provide information about the VNET and Subnet.
  - When "Vnet Injected" is empty, use XTS to check if the server set for Public or Private access

The image below shows a server set to Private Access (VNET integration):

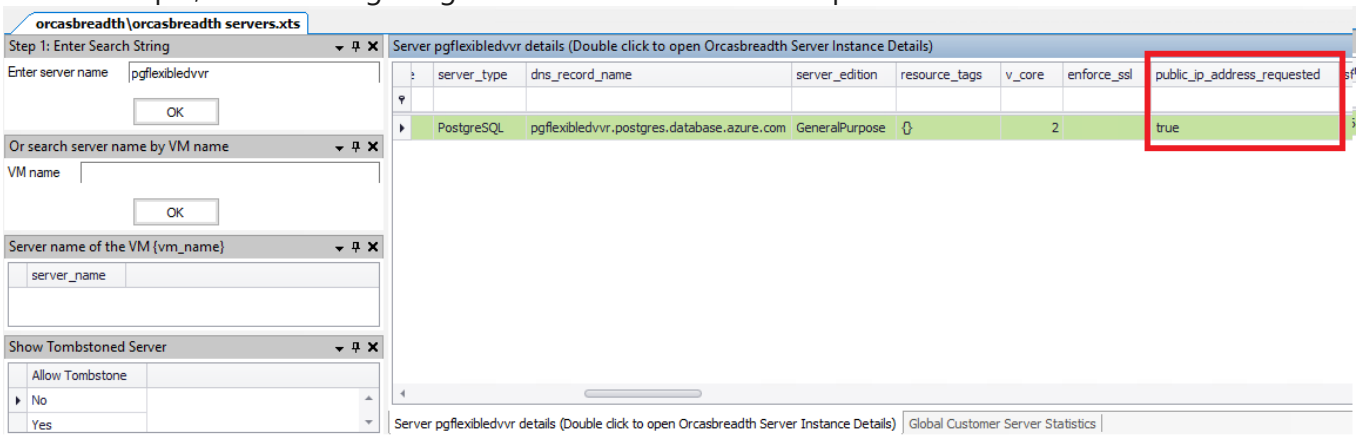


- Using XTS

  1. Open XTS
  2. Select the environment
  3. Open the view **orcasbreadth\orcasbreadth servers.xts** and look for the server
  4. Go to tab Server details and check the column *public_ip_address_requested*:

     - True: the server is set for public access
     - False: the server is set for private access (VNET integration)

  As an example, the following image shows a server that allows public access



2. Get firewall rules for servers that allow Public Access

There are two ways to get the firewall rules for a server that allow Public Access using XTS.

To interpret the information, use the following general guidelines:

- All rules where owner is *Customer* were set by the customer.

- All rules where owner is *Microsoft* were set by the Microsoft and cannot be controlled by the customer.
- If state is Dropped, the rule existed but was dropped by customer, so it is not considered to allow access to the server.
- If *Allow public access from any Azure service within Azure to this server* is enabled on the server, you will see a rule with a name that starts with *AllowAllAzureServicesAndResourcesWithinAzureIps_* and state *Succeeded*.
- If there is no rule with a name that start with *AllowAllAzureServicesAndResourcesWithinAzureIps_*, or it exists with a state other than *Succeeded* then "Allow public access from any Azure service within Azure to this server" is disabled on the server.
- Any other rule where owner is *Customer* allows access to a specific IP or IP range.

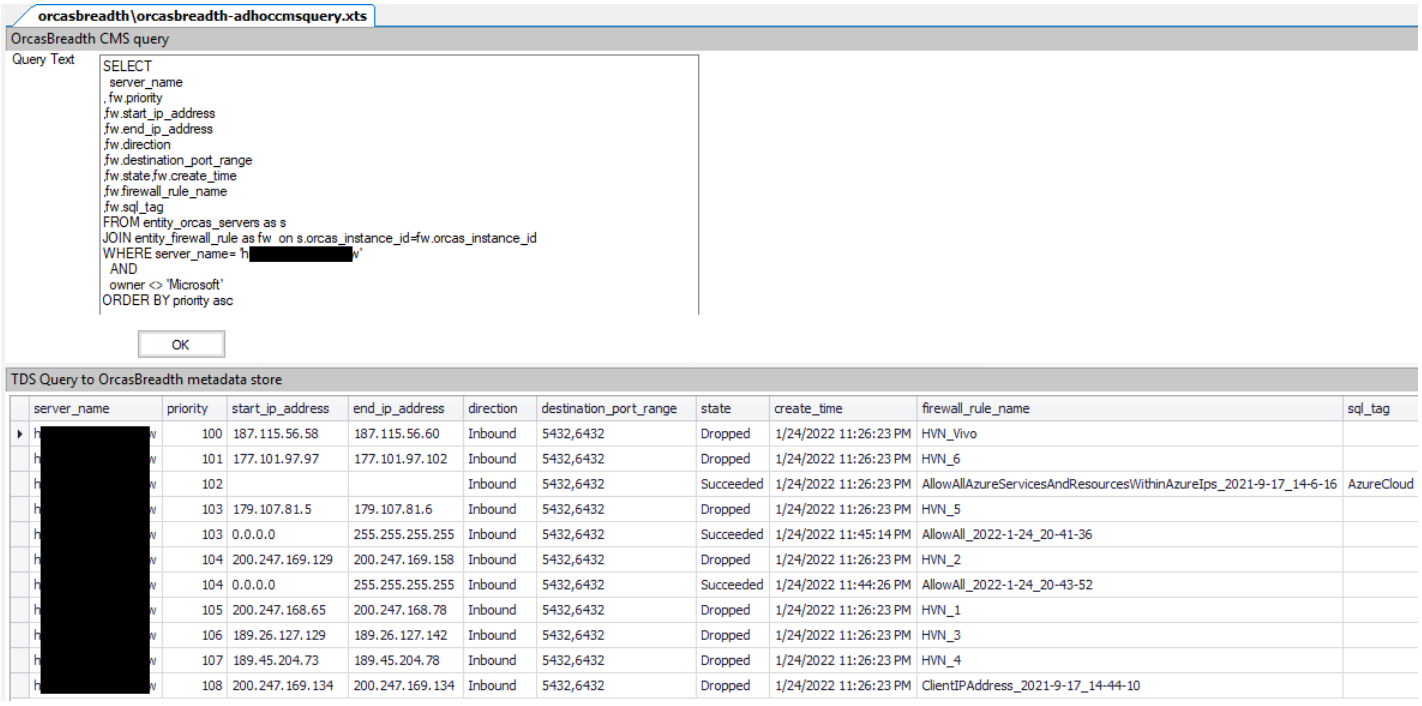
A- Using **orcasbreadth\orcasbreadth-adhoccmsquery.xts** view

1. Select the environment

2. Open the view **orcasbreadth\orcasbreadth-adhoccmsquery.xts** and run the following query:

```
SELECT
  server_name
, fw.priority
,fw.start_ip_address
,fw.end_ip_address
,fw.direction
,fw.destination_port_range
,fw.state,fw.create_time
,fw.firewall_rule_name
,fw.sql_tag
FROM entity_orcas_servers as s
JOIN entity_firewall_rule as fw on s.orcas_instance_id=fw.orcas_instance_id
WHERE server_name= '<YOUR-SERVER-NAME>'
 AND
 owner <> 'Microsoft'
ORDER BY priority asc
```

As an example, in the following image, there is a server with *Allow public access from any Azure service within Azure to this server* enabled, has two active rules: *AllowAll_2022-1-24_20-41-36* and *AllowAll_2022-1-24_20-43-52*, and there are eight rules that existed but were deleted

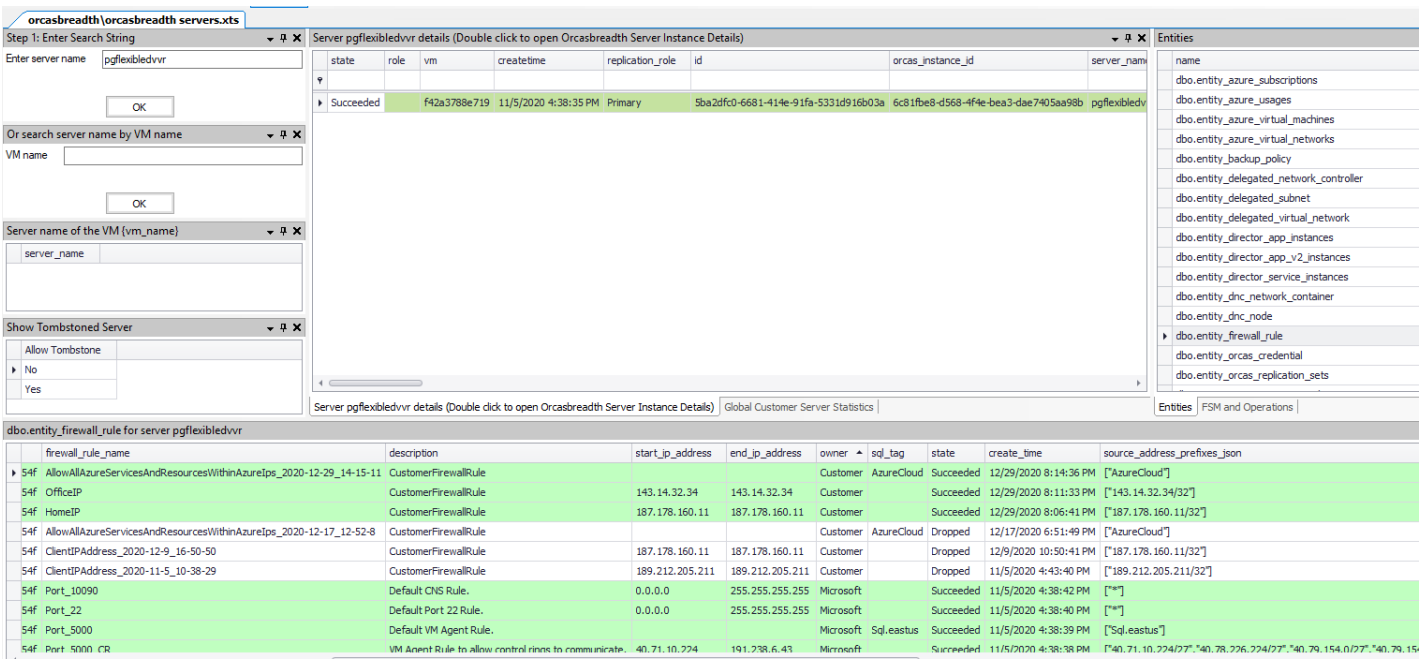


server_name	priority	start_ip_address	end_ip_address	direction	destination_port_range	state	create_time	firewall_rule_name	sql_tag
h[redacted]w	100	187.115.56.58	187.115.56.60	Inbound	5432,6432	Dropped	1/24/2022 11:26:23 PM	HVN_Vivo	
h[redacted]w	101	177.101.97.97	177.101.97.102	Inbound	5432,6432	Dropped	1/24/2022 11:26:23 PM	HVN_6	
h[redacted]w	102			Inbound	5432,6432	Succeeded	1/24/2022 11:26:23 PM	AllowAllAzureServicesAndResourcesWithinAzureIps_2021-9-17_14-6-16	AzureCloud
h[redacted]w	103	179.107.81.5	179.107.81.6	Inbound	5432,6432	Dropped	1/24/2022 11:26:23 PM	HVN_5	
h[redacted]w	103	0.0.0.0	255.255.255.255	Inbound	5432,6432	Succeeded	1/24/2022 11:45:14 PM	AllowAll_2022-1-24_20-41-36	
h[redacted]w	104	200.247.169.129	200.247.169.158	Inbound	5432,6432	Dropped	1/24/2022 11:26:23 PM	HVN_2	
h[redacted]w	104	0.0.0.0	255.255.255.255	Inbound	5432,6432	Succeeded	1/24/2022 11:44:26 PM	AllowAll_2022-1-24_20-43-52	
h[redacted]w	105	200.247.168.65	200.247.168.78	Inbound	5432,6432	Dropped	1/24/2022 11:26:23 PM	HVN_1	
h[redacted]w	106	189.26.127.129	189.26.127.142	Inbound	5432,6432	Dropped	1/24/2022 11:26:23 PM	HVN_3	
h[redacted]w	107	189.45.204.73	189.45.204.78	Inbound	5432,6432	Dropped	1/24/2022 11:26:23 PM	HVN_4	
h[redacted]w	108	200.247.169.134	200.247.169.134	Inbound	5432,6432	Dropped	1/24/2022 11:26:23 PM	ClientIPAddress_2021-9-17_14-44-10	

B - Using **orcasbreadth\orcasbreadth servers.xts**

1. Select the environment
2. Open the view **orcasbreadth\orcasbreadth servers.xts** and look for the server
3. Go to entity **dbo.entity_firewall_rule**

   As an example, in the following image, there is a server with *Allow public access from any Azure service within Azure to this server* enabled, has two rules: OfficeIP and HomeIP, and there are three rules that existed but were deleted



	firewall_rule_name	description	start_ip_address	end_ip_address	owner	sql_tag	state	create_time	source_address_prefixes_json
54f	AllowAllAzureServicesAndResourcesWithinAzureIps_2020-12-29_14-15-11	CustomerFirewallRule			Customer	AzureCloud	Succeeded	12/29/2020 8:14:36 PM	["AzureCloud"]
54f	OfficeIP	CustomerFirewallRule	143.14.32.34	143.14.32.34	Customer		Succeeded	12/29/2020 8:11:33 PM	["143.14.32.34/32"]
54f	HomeIP	CustomerFirewallRule	187.178.160.11	187.178.160.11	Customer		Succeeded	12/29/2020 8:06:41 PM	["187.178.160.11/32"]
54f	AllowAllAzureServicesAndResourcesWithinAzureIps_2020-12-17_12-52-8	CustomerFirewallRule			Customer	AzureCloud	Dropped	12/17/2020 6:51:49 PM	["AzureCloud"]
54f	ClientIPAddress_2020-12-9_16-50-50	CustomerFirewallRule	187.178.160.11	187.178.160.11	Customer		Dropped	12/9/2020 10:50:41 PM	["187.178.160.11/32"]
54f	ClientIPAddress_2020-11-5_10-38-29	CustomerFirewallRule	189.212.205.211	189.212.205.211	Customer		Dropped	11/5/2020 4:43:40 PM	["189.212.205.211/32"]
54f	Port_10090	Default CNS Rule.	0.0.0.0	255.255.255.255	Microsoft		Succeeded	11/5/2020 4:38:42 PM	["*"]
54f	Port_22	Default Port 22 Rule.	0.0.0.0	255.255.255.255	Microsoft		Succeeded	11/5/2020 4:38:40 PM	["*"]
54f	Port_5000	Default VM Agent Rule.			Microsoft	Sql.eastus	Succeeded	11/5/2020 4:38:39 PM	["Sql.eastus"]
54f	Port_5000_CR	VM Agent Rule to allow control rings to communicate.	40.71.10.224	191.238.6.43	Microsoft		Succeeded	11/5/2020 4:38:38 PM	["40.71.10.224/27","40.78.226.224/27","40.79.154.0/27","40.79.15

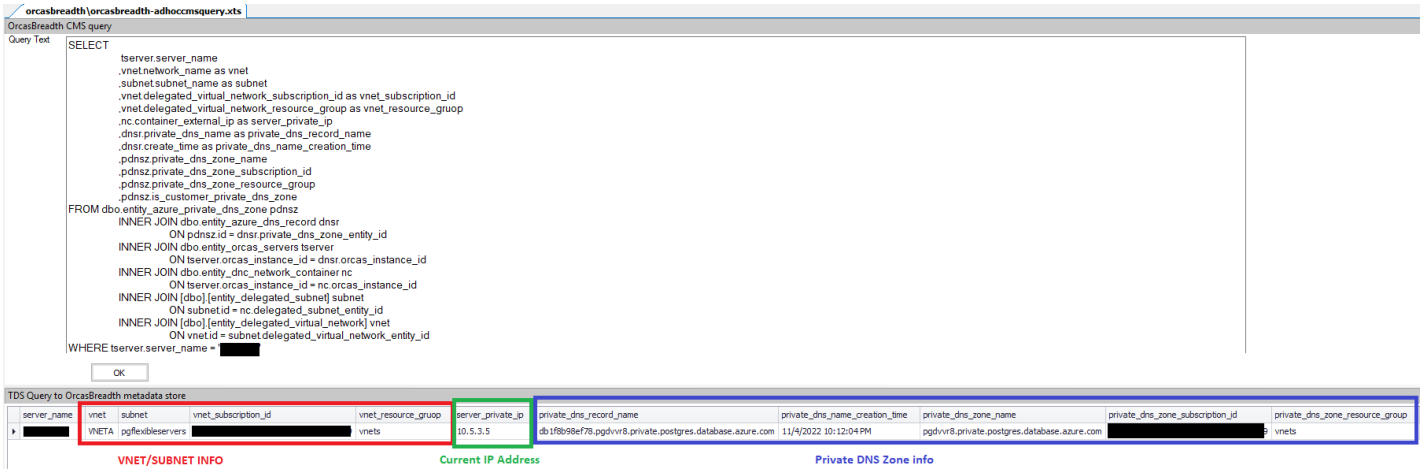
3. Get VNet/subnet, Private DNS Zone and private IP Address for a server set for Private Access (VNET Integration)

NOTE: Using the XTS method, provides more information that ASC as it shows the private IP for the server, and information about the subscriptions and resource group for the VNET and the Private DNS Zone as they can be different from those used by the db server, and it also shows the DNS record name used by the server in the Private DNS Zone.

- Using XTS

  1. Open XTS
  2. Select the environment
  3. Open the view **orcasbreadth\orcasbreadth-adhoccmsquery.xts** and execute the query below:

```
SELECT
  tserver.server_name
 ,vnet.network_name as vnet
 ,subnet.subnet_name as subnet
 ,vnet.delegated_virtual_network_subscription_id as vnet_subscription_id
 ,vnet.delegated_virtual_network_resource_group as vnet_resource_gruop
 ,nc.container_external_ip as server_private_ip
 ,dnsr.private_dns_name as private_dns_record_name
 ,dnsr.create_time as private_dns_name_creation_time
 ,pdnsz.private_dns_zone_name
 ,pdnsz.private_dns_zone_subscription_id
 ,pdnsz.private_dns_zone_resource_group
 ,pdnsz.is_customer_private_dns_zone
FROM dbo.entity_azure_private_dns_zone pdnsz
 INNER JOIN dbo.entity_azure_dns_record dnsr
        ON pdnsz.id = dnsr.private_dns_zone_entity_id
 INNER JOIN dbo.entity_orcas_servers tserver
        ON tserver.orcas_instance_id = dnsr.orcas_instance_id
 INNER JOIN dbo.entity_dnc_network_container nc
        ON tserver.orcas_instance_id = nc.orcas_instance_id
 INNER JOIN [dbo].[entity_delegated_subnet] subnet
        ON subnet.id = nc.delegated_subnet_entity_id
 INNER JOIN [dbo].[entity_delegated_virtual_network] vnet
        ON vnet.id = subnet.delegated_virtual_network_entity_id
WHERE tserver.server_name = '<YOUR-SERVER-NAME>'
```



- Using ASC

  Go to the Properties Tab and filter properties with name that includes "net".

  - When "Vnet Injected" is Yes, the other parameters provide information about the VNET and Subnet.
  - When "Vnet Injected" is empty, use XTS to check if the server set for Public or Private access and get vnet/subnet information

p[redacted]d
Microsoft.DBforPostgreSQL/flexibleServers

Properties | Connectivity | Perf | Query Store Perf | Incident Helper (P42) | Operation | Sandbox | Server Parameters | Backup | Restore | HA
Customer Facing Metrics: Burstable SKU Metrics | Customer Facing Metrics: Storage Metrics | Customer Facing Metrics: Connectivity & Networking | Information | Insig
Health

Server Info | Server Storage Info | Container Info | Resource Information of All Containers | Information of VNET, Subnets and Peerings

Server Info

High level information about this server

Drag a column header and drop it here to group by that column

PropertyName	PropertyValue
net	
Vnet Injected	Yes
Vnet Name	vn-odpr-eus2-db
Vnet Resource Group	rg[redacted]rk
Vnet Subscription ID	7[redacted]7F
Vnet State	Succeeded
Subnet Name	sn-odpr-eus2-db-pg-babelfish
Subnet State	Succeeded