Virtual Network Interface Endpoints and rules for Azure SQL Database

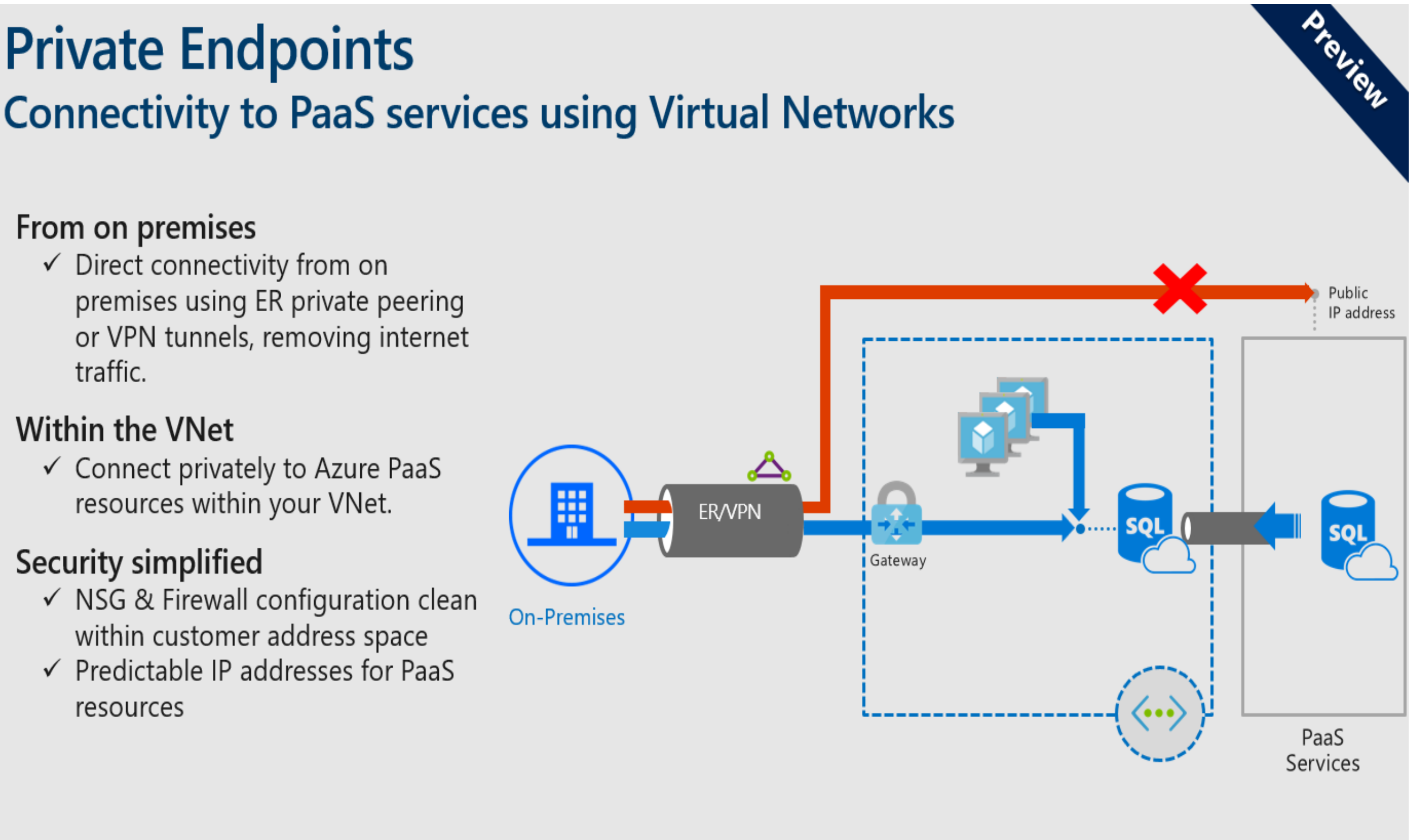
Last updated by | Vitor Tomaz | Aug 5, 2020 at 12:40 PM PDT

Contents

Virtual Network Interface Endpoints and rules for Azure SQL Database

This feature is currently in Preview to enroll into the preview please contact dmalik@microsoft.com

This article explains how to use Virtual Network Interface Endpoints with Azure SQL Database to enable connectivity on a Private IP to your Azure SQLDB or Azure SQLDW. Connectivity on the Azure SQLDB is enforced with the help of Virtual Network Rules that govern which Virtual Network/Sub Networks can connect.



Private Endpoints

Connectivity to PaaS services using Virtual Networks

From on premises

✓ Direct connectivity from on premises using ER private peering or VPN tunnels, removing internet traffic.

Within the VNet

✓ Connect privately to Azure PaaS resources within your VNet.

Security simplified

✓ NSG & Firewall configuration clean within customer address space

✓ Predictable IP addresses for PaaS resources

Terminology and Description

Virtual network: You can have virtual networks associated with your Azure subscription.

Subnet: A virtual network contains subnets. Any Azure virtual machines (VMs) that you have are assigned to subnets. One subnet can contain multiple VMs or other compute nodes. Compute nodes that are outside of your virtual network cannot access your virtual network unless you configure your security to allow access.

Virtual Network Interface Endpoint: A Virtual Network Interface Endpoint is an endpoint created inside a Virtual Network, which is associated with a given Azure SQLDB logical server. The Interface Endpoint has properties associated with it like a Private IP, VNET, Subnet, these properties are then inherited by your Azure SQLDB Logical Server. For more information on Interface Endpoints please refer to *this* document. Creation of an Interface Endpoint alone is not enough to enable connectivity to your Azure SQLDB logical server for that you must create a Virtual Network Rule.

Virtual network rule: A virtual network rule for your SQL Database server is a subnet that is listed in the access control list (ACL) of your SQL Database server. A virtual network rule tells your SQL Database server to accept communications from every node that is on the subnet.

NOTE: You must create a Virtual Network Rule on your server to enable connectivity on the private IP.

Benefits

Configure a Private IP on your Azure SQLDB

This feature will allow you to associate an IP from within your VNET/Subnet to your Azure SQLDB. When connecting from within a VNET you will not have to allow outbound on your NSGs to Azure SQLDB public IPs.

Allow VPN and Express Route Connectivity

As your Azure SQLDB will now be accessible from a private IP within your subnet you will now be able to connect to it from on premises using VPN or Express Route private peering.

Limitations

iDNS support is not yet available

On configuring an Interface Endpoint for your Azure SQLDB logical server you must first populate the DNS entry for your logical server in a DNS server that resides within the VNET.

Only one geographic region

You cannot map an Interface Endpoints cross region, the Interface Endpoint and the Azure SQLDB logical server must be in the same region.

Server level not Database level

The interface endpoint maps to a given server not a specific database within that server. So, when connecting to any database in the server the same IP will be used.

Public IP connectivity does not get terminated

Assigning your Azure SQLDB a private IP via interface endpoints does not disable the public IP that the database has. You can still allow client public IPs to connect if they are allowed in the firewall rules, same applies to ‘Allow all Azure Services’.

Cannot have Server in multiple VNETs

At present an Interface Endpoint can only be mapped to a given VNET.

Cannot Delete Server with Interface Endpoint Associated (Preview limitation only)

At this point you cannot delete a SQLDB logical server that has an interface endpoint associated with it. You can delete the server after you have removed the interface endpoint, this is a limitation for preview only and will be lifted after preview.

Detailed Workflow

There are a number of steps involved in enabling connectivity to your Azure SQLDB when using Interface Endpoints and securing your Azure SQLDB logical server. These are as follows:

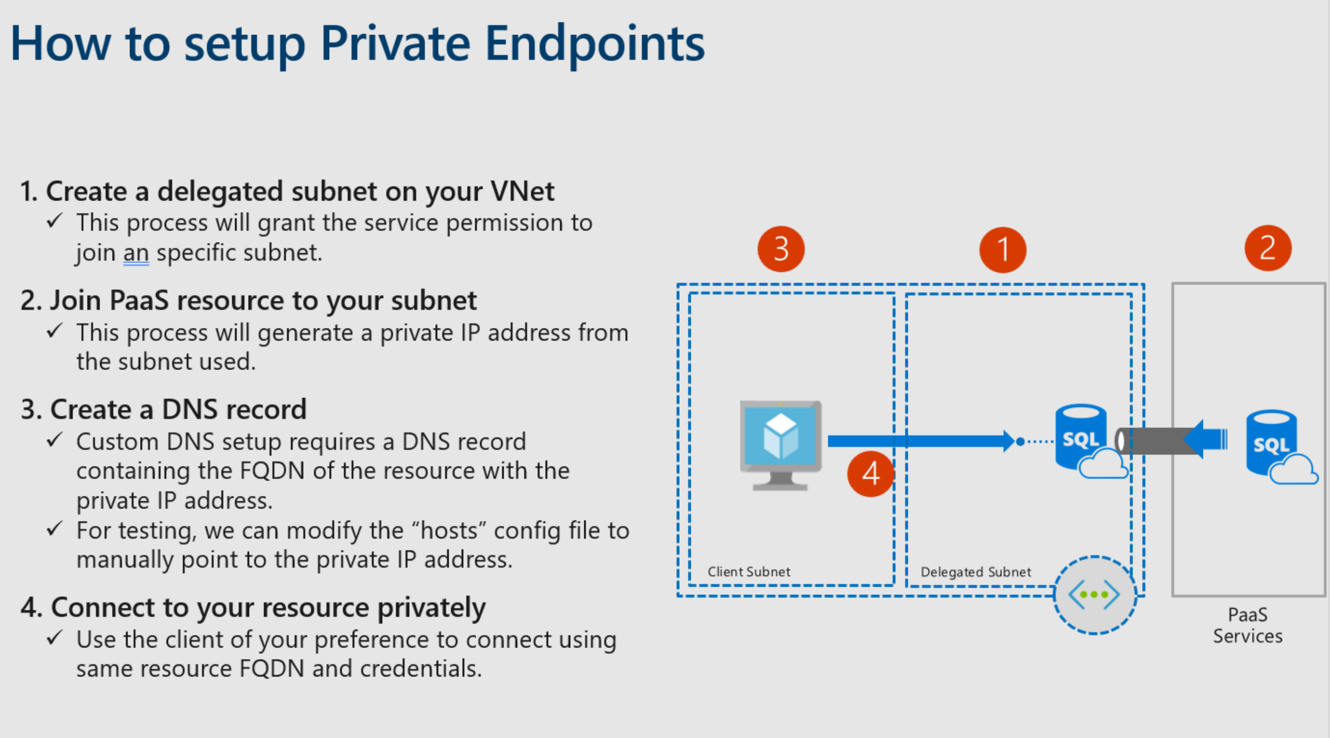
1. Create a delegated subnet in the VNET where you want to enable connectivity to your Azure SQLDB Server. More details can be found in *this* document.
2. Create an Interface Endpoint on your Azure SQLDB logical server.
3. Update your custom DNS settings to reflect the new IP assigned to the Azure SQLDB logical server.
4. Create a virtual network rule specifying the subnet that you want to allow connectivity from.

Migrating from Public IP connectivity

If you have an app connecting to Azure SQLDB on the Public IP and you want to switch over to private IPs using interface endpoints then this might result in loss of connectivity in the following scenarios:

1. DNS entries not populated and outbound to SQLDB Public IP removed: one of the scenarios for using a private IP is to remove outbound to SQLDB Public IPs. If you map your SQLDB Logical Server to an Interface Endpoint and remove outbound to the public IP then unless a new DNS entry is made on the custom DNS that you have in your VNET, connectivity to your SQLDB Logical Server will break. A way to mitigate this is to remove outbound to SQLDB public IP only after you have the DNS entry in the custom DNS.
2. VNET Firewall rule is not specified on the SQLDB Logical Server: just mapping your SQLDB Logical Server to an Interface Endpoint is not enough to ensure connectivity you must allow the specific VNET/Subnet to connect to your logical server. As a mitigation you can put the VNET Firewall rule on the logical server before you assign it an Interface Endpoint.

Getting Started with PowerShell



How to setup Private Endpoints

1. **Create a delegated subnet on your VNet**
   - ✓ This process will grant the service permission to join an specific subnet.
2. **Join PaaS resource to your subnet**
   - ✓ This process will generate a private IP address from the subnet used.
3. **Create a DNS record**
   - ✓ Custom DNS setup requires a DNS record containing the FQDN of the resource with the private IP address.
   - ✓ For testing, we can modify the “hosts” config file to manually point to the private IP address.
4. **Connect to your resource privately**
   - ✓ Use the client of your preference to connect using same resource FQDN and credentials.

Prerequisites

To onboard for private preview, please look at the onboarding instructions doc to enable this feature.

To use the MSI provided here you must have PS 5.0 or above on your machine.

Getting Started with Powershell

Please follow below instructions to get started with Powershell:

1. Download the Powershell MSI and scripts in this folder.
2. Install the Private Powershell MSI
   1. Because this is a Private PowerShell build, you'll need to uninstall any existing Microsoft Azure PowerShell bits on your system.

3. Run PSH_ACL_SkipSNCheck.reg. This bypasses strong name verification bits for the preview assemblies.
4. Launch Powershell and follow the instructions below to get started. You can also modify Powershell_Demo_Commands.PS1.

To sign into Azure and select your subscription, please follow the instructions at How to Install and Configure Azure PowerShell.

Scenario: Put Azure SQLDB logical server inside my VNet on a Private IP

The steps below help you create a Delegated Subnet within a

Assign SQL delegation to a pre-created Subnet

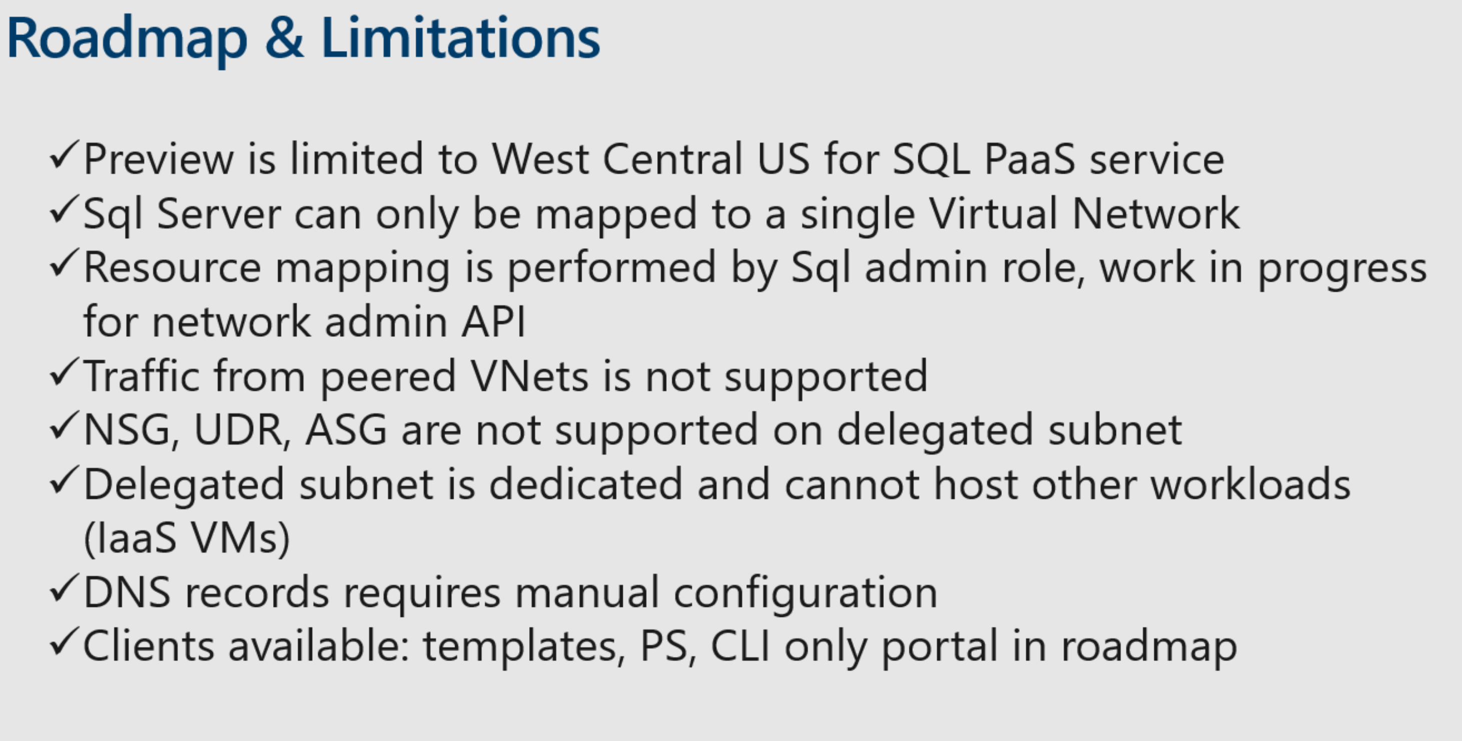
```
Add-AzureRmDelegation -Name 'SqlDelegation' -ServiceName 'Microsoft.Sql/servers' -Subnet $subnet
```

Put an interface Endpoint on the SQLDB logical server

```
$interfaceEndpoint = New-AzureRmSqlServerInterfaceEndpointProfile -ResourceGroupName $serverRGName -ServerName $serverName -InterfaceEndpointProfileName $interfaceEndpointProfileName -VirtualNetworkSubnetId $subnet.ID
```

Verify that the IE has been associated with the server

```
Get-AzureRmSqlServerInterfaceEndpointProfile -ResourceGroupName $serverRGName -ServerName $serverName
```



Roadmap & Limitations

- ✓Preview is limited to West Central US for SQL PaaS service
- ✓Sql Server can only be mapped to a single Virtual Network
- ✓Resource mapping is performed by Sql admin role, work in progress for network admin API
- ✓Traffic from peered VNets is not supported
- ✓NSG, UDR, ASG are not supported on delegated subnet
- ✓Delegated subnet is dedicated and cannot host other workloads (IaaS VMs)
- ✓DNS records requires manual configuration
- ✓Clients available: templates, PS, CLI only portal in roadmap

How good have you found this content?