Secure your application in a hub and spoke topology!

Agenda

[**Introduction** 2](#_Toc127960052)

[**Scenario 1: Hub and spoke architecture - Dedicated virtual network for private endpoints** 3](#_Toc127960053)

[**Scenario 2: Hub and spoke architecture - Shared virtual network for private endpoints and virtual machines** 3](#_Toc127960054)

[**Scenario 3: Single virtual network** 4](#_Toc127960055)

[**Scenario 4: On-premises traffic to private endpoints** 5](#_Toc127960056)

[**LAB :** 6](#_Toc127960057)

[**Create a VM & networks** 6](#_Toc127960058)

[**Azure Firewall network** 7](#_Toc127960059)

[**Virtual machine network** 8](#_Toc127960060)

[**Private endpoint network** 8](#_Toc127960061)

[**Create virtual machine** 9](#_Toc127960062)

[**Deploy the Firewall** 10](#_Toc127960063)

[**Enable firewall logs** 11](#_Toc127960064)

[**Create Azure SQL database** 11](#_Toc127960065)

[**Create private endpoint** 12](#_Toc127960066)

[**Connect the virtual networks using virtual network peering** 14](#_Toc127960067)

[**Link the virtual networks to the private DNS zone** 14](#_Toc127960068)

[**Configure an application rule with SQL FQDN in Azure Firewall** 15](#_Toc127960069)

[**Route traffic between the virtual machine and private endpoint through Azure Firewall** 16](#_Toc127960070)

[**Connect to the virtual machine from your client computer** 17](#_Toc127960071)

[**Access SQL Server privately from the virtual machine** 17](#_Toc127960072)

[**Validate the traffic in Azure Firewall logs** 18](#_Toc127960073)

[**Clean up resources** 19](#_Toc127960074)

**Introduction**

Azure Private Endpoint is the fundamental building block for Azure Private Link. Private endpoints enable Azure resources deployed in a virtual network to communicate privately with private link resources.

Private endpoints allow resources access to the private link service deployed in a virtual network. Access to the private endpoint through virtual network peering and on-premises network connections extend the connectivity.

**Note**

If you want to secure traffic to private endpoints in Azure Virtual WAN using secured virtual hub, see **[Secure traffic destined to private endpoints in Azure Virtual WAN](https://learn.microsoft.com/en-us/azure/firewall-manager/private-link-inspection-secure-virtual-hub)**.

You may need to inspect or block traffic from clients to the services exposed via private endpoints. Complete this inspection by using [Azure Firewall](https://learn.microsoft.com/en-us/azure/firewall/overview) or a third-party network virtual appliance.

The following limitations apply:

* Network security groups (NSG) are bypassed by traffic coming from private endpoints
* User-defined routes (UDR) are bypassed by traffic coming from private endpoints. User-defined routes can be used to override traffic destined for the private endpoint.
* A single route table can be attached to a subnet
* A route table supports up to 400 routes

Azure Firewall filters traffic using either:

* [FQDN in network rules](https://learn.microsoft.com/en-us/azure/firewall/fqdn-filtering-network-rules) for TCP and UDP protocols
* [FQDN in application rules](https://learn.microsoft.com/en-us/azure/firewall/features#application-fqdn-filtering-rules) for HTTP, HTTPS, and MSSQL.

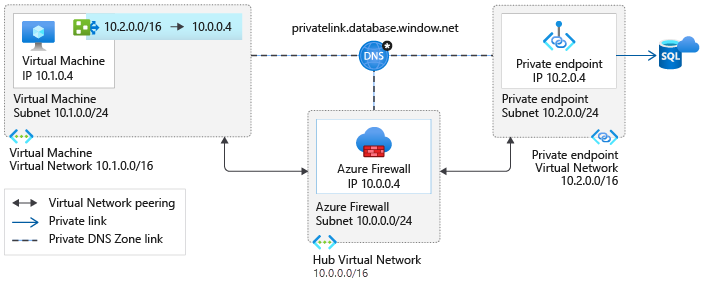
**Important**

The use of application rules over network rules is recommended when inspecting traffic destined to private endpoints in order to maintain flow symmetry. If network rules are used, or an NVA is used instead of Azure Firewall, SNAT must be configured for traffic destined to private endpoints.

**Note**

SQL FQDN filtering is supported in [**proxy-mode**](https://learn.microsoft.com/en-us/azure/azure-sql/database/connectivity-architecture#connection-policy) only (port 1433). **Proxy** mode can result in more latency compared to *redirect*. If you want to continue using redirect mode, which is the default for clients connecting within Azure, you can filter access using FQDN in firewall network rules.

**Scenario 1: Hub and spoke architecture - Dedicated virtual network for private endpoints**

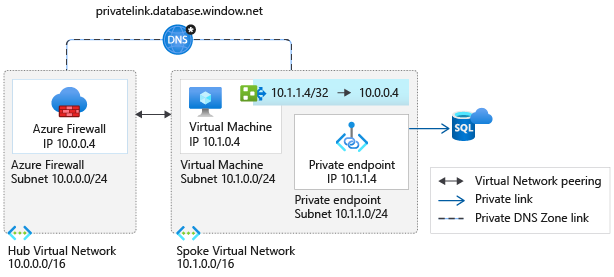


This scenario is the most expandable architecture to connect privately to multiple Azure services using private endpoints. A route pointing to the network address space where the private endpoints are deployed is created. This configuration reduces administrative overhead and prevents running into the limit of 400 routes.

Connections from a client virtual network to the Azure Firewall in a hub virtual network will incur charges if the virtual networks are peered. Connections from Azure Firewall in a hub virtual network to private endpoints in a peered virtual network are not charged.

For more information on charges related to connections with peered virtual networks, see the FAQ section of the [pricing](https://azure.microsoft.com/pricing/details/private-link/) page.

**Scenario 2: Hub and spoke architecture - Shared virtual network for private endpoints and virtual machines**



This scenario is implemented when:

* It's not possible to have a dedicated virtual network for the private endpoints
* When only a few services are exposed in the virtual network using private endpoints

The virtual machines will have /32 system routes pointing to each private endpoint. One route per private endpoint is configured to route traffic through Azure Firewall.

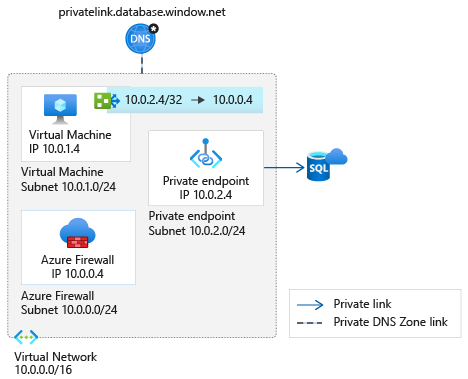
The administrative overhead of maintaining the route table increases as services are exposed in the virtual network. The possibility of hitting the route limit also increases.

Depending on your overall architecture, it's possible to run into the 400 routes limit. It's recommended to use scenario 1 whenever possible.

Connections from a client virtual network to the Azure Firewall in a hub virtual network will incur charges if the virtual networks are peered. Connections from Azure Firewall in a hub virtual network to private endpoints in a peered virtual network are not charged.

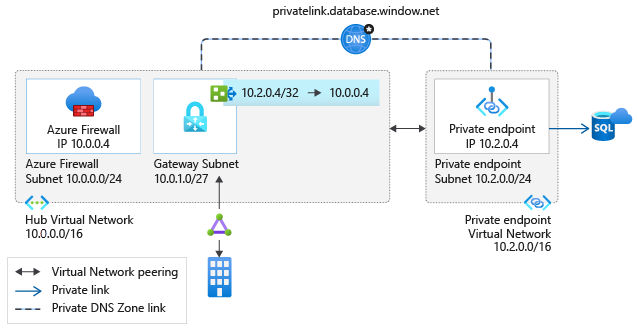
For more information on charges related to connections with peered virtual networks, see the FAQ section of the [pricing](https://azure.microsoft.com/pricing/details/private-link/) page.

**Scenario 3: Single virtual network**



Use this pattern when a migration to a hub and spoke architecture isn't possible. The same considerations as in scenario 2 apply. In this scenario, virtual network peering charges don't apply.

**Scenario 4: On-premises traffic to private endpoints**



This architecture can be implemented if you have configured connectivity with your on-premises network using either:

* [ExpressRoute](https://learn.microsoft.com/en-us/azure/expressroute/expressroute-introduction)
* [Site to Site VPN](https://learn.microsoft.com/en-us/azure/vpn-gateway/tutorial-site-to-site-portal)

If your security requirements require client traffic to services exposed via private endpoints to be routed through a security appliance, deploy this scenario.

The same considerations as in scenario 2 above apply. In this scenario, there aren't virtual network peering charges. For more information about how to configure your DNS servers to allow on-premises workloads to access private endpoints, see [On-Premises workloads using a DNS forwarder](https://learn.microsoft.com/en-us/azure/private-link/private-endpoint-dns#on-premises-workloads-using-a-dns-forwarder).

**LAB :**

In this Lab you will deploy Scenario 1 with a Hub and spoke topology. You’ll create three virtual networks and their corresponding subnets to:

* Contain the Azure Firewall used to restrict communication between the VM and the private endpoint.
* Host the VM that is used to access your private link resource.
* Host the private endpoint.

**Prerequisites:**

* An Azure subscription.
* A Log Analytics workspace.

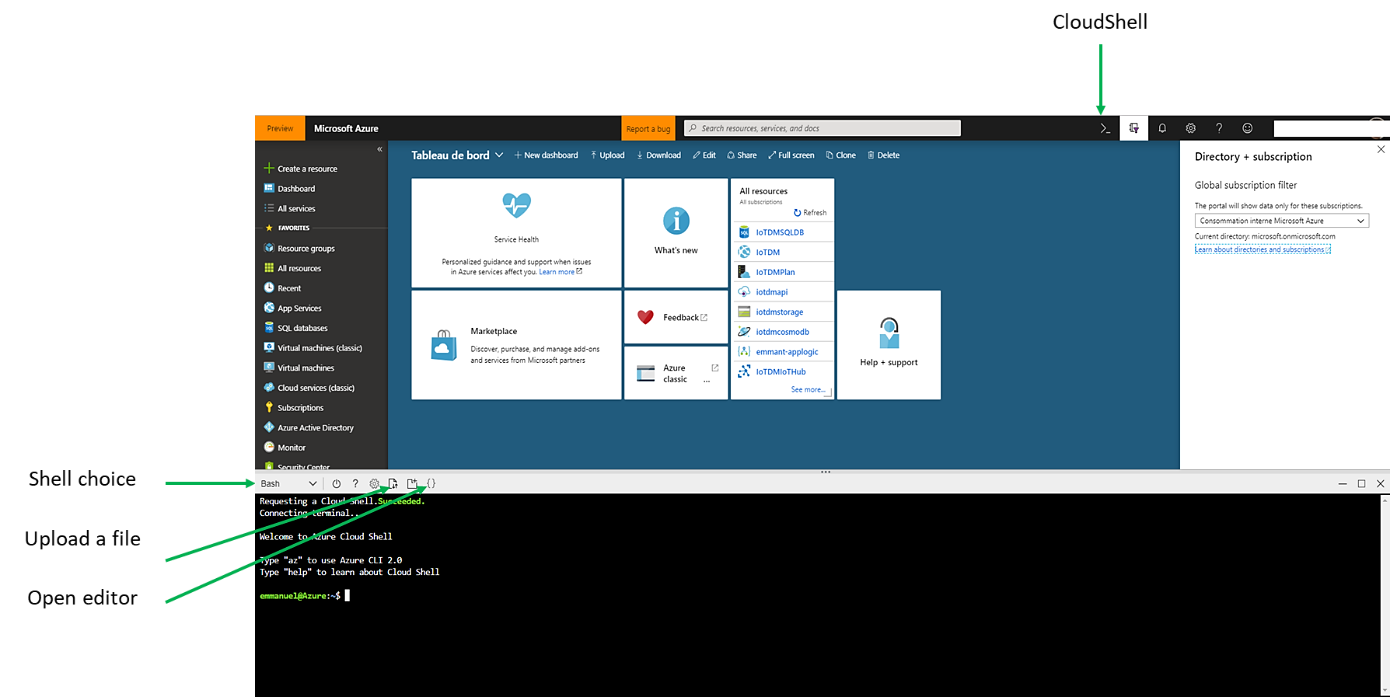
See, [Create a Log Analytics workspace in the Azure portal](https://learn.microsoft.com/en-us/azure/azure-monitor/logs/quick-create-workspace) to create a workspace if you don't have one in your subscription.

**Create a VM & networks**

In this section, you'll create a virtual network and subnet to host the VM used to access your private link resource. An Azure SQL database is used later as the example service.

We will be using an Azure CLI script to deploy the networks and the VM:

1. Download the script here: https://aka.ms/AzureNetworksLab
2. Login to the portal & launch the cloud shell:



1. List your subscriptions :

*az account list –o table*

1. Set the right the subscription if needed:

*az account set ––subscription {id}*

1. Upload the script via Azure portal
2. Move the script under the clouddrive

*mv  1-1-Create-Network.sh  clouddrive/*

1. Go to the clouddrive

*cd clouddrive*

1. Launch the script

*./1-1-Create-Network.sh*

Check the deployment, you should have a resource group **myResourceGroup**, three networks and a VM with following parameters:

**Azure Firewall network**

| **Parameter** | **Value** |
| --- | --- |
| **<virtual-network-name>** | myAzFwVNet |
| **<region-name>** | North Europe |
| **<IPv4-address-space>** | 10.0.0.0/16 |
| **<subnet-name>** | AzureFirewallSubnet |
| **<subnet-address-range>** | 10.0.0.0/24 |

**Virtual machine network**

| **Parameter** | **Value** |
| --- | --- |
| **<virtual-network-name>** | myVMVNet |
| **<region-name>** | North Europe |
| **<IPv4-address-space>** | 10.1.0.0/16 |
| **<subnet-name>** | VMSubnet |
| **<subnet-address-range>** | 10.1.0.0/24 |

**Private endpoint network**

| **Parameter** | **Value** |
| --- | --- |
| **<virtual-network-name>** | myPEVNet |
| **<region-name>** | North Europe |
| **<IPv4-address-space>** | 10.2.0.0/16 |
| **<subnet-name>** | PrivateEndpointSubnet |
| **<subnet-address-range>** | 10.2.0.0/24 |

**Virtual machine**

|  |  |
| --- | --- |
| **Instance details** |  |
| Virtual machine name | **myVM**. |
| Region | Select North Europe. |
| Image | Select **Ubuntu Server 18.04 LTS - Gen1**. |
| Size | Select **Standard\_B2s**. |
|  |  |

**Note**

Azure provides a default outbound access IP for VMs that either aren't assigned a public IP address or are in the back-end pool of an internal basic Azure load balancer. The default outbound access IP mechanism provides an outbound IP address that isn't configurable.

The default outbound access IP is disabled when a public IP address is assigned to the VM, the VM is placed in the back-end pool of a standard load balancer, with or without outbound rules, or if an [**Azure Virtual Network NAT gateway**](https://learn.microsoft.com/en-us/azure/virtual-network/nat-gateway/nat-overview) resource is assigned to the subnet of the VM.

VMs that are created by virtual machine scale sets in flexible orchestration mode don't have default outbound access.

For more information about outbound connections in Azure, see [**Default outbound access in Azure**](https://learn.microsoft.com/en-us/azure/virtual-network/ip-services/default-outbound-access) and [**Use source network address translation (SNAT) for outbound connections**](https://learn.microsoft.com/en-us/azure/load-balancer/load-balancer-outbound-connections).

**Deploy the Firewall**

1. On the Azure portal menu or from the **Home** page, select **Create a resource**.
2. Type **firewall** in the search box and press **Enter**.
3. Select **Firewall** and then select **Create**.
4. On the **Create a Firewall** page, use the following table to configure the firewall:

| **Setting** | **Value** |
| --- | --- |
| **Project details** |  |
| Subscription | Select your subscription. |
| Resource group | Select **myResourceGroup**. |
| **Instance details** |  |
| Name | Enter **myAzureFirewall**. |
| Region | Select North Europe |
| Availability zone  Firewall SKU  Firewall policy | Leave the default **None**.  Premium  Select **Add new** and in Name enter **myFirewall-policy** |
| Choose a virtual network | Select **Use Existing**. |
| Virtual network | Select **myAzFwVNet**. |
| Public IP address | Select **Add new** and in Name enter **myFirewall-ip**. |
| Forced tunneling | Leave the default **Disabled**. |
|  |  |

1. Select **Review + create**. You're taken to the **Review + create** page where Azure validates your configuration.
2. When you see the **Validation passed** message, select **Create**.

**Enable firewall logs**

In this section, you enable the logs on the firewall.

1. In the Azure portal, select **All resources** in the left-hand menu.
2. Select the firewall **myAzureFirewall** in the list of resources.
3. Under **Monitoring** in the firewall settings, select **Diagnostic settings**
4. Select **+ Add diagnostic setting** in the Diagnostic settings.
5. In **Diagnostics setting**, enter or select this information:

| **Setting** | **Value** |
| --- | --- |
| Diagnostic setting name | Enter **myDiagSetting**. |
| Category details |  |
| log | Select **AzureFirewallApplicationRule** and **AzureFirewallNetworkRule**. |
| Destination details | Select **Send to Log Analytics**. |
| Subscription | Select your subscription. |
| Log Analytics workspace | Select your Log Analytics workspace. |

1. Select **Save**.

**Create Azure SQL database**

In this section, you create a private SQL Database.

1. On the upper-left side of the screen in the Azure portal, select **Create a resource** > **Databases** > **SQL Database**.
2. In **Create SQL Database - Basics**, enter or select this information:

| **Setting** | **Value** |
| --- | --- |
| **Project details** |  |
| Subscription | Select your subscription. |
| Resource group | Select **myResourceGroup**. You created this resource group in the previous section. |
| **Database details** |  |
| Database name | Enter **mydatabase**. |
| Server | Select **Create new** and enter the information below. |
| Server name | Enter **mydbserver1**. If this name is taken, enter a unique name. |
| Server admin login | Enter a name of your choosing. |
| Password | Enter a password of your choosing. |
| Confirm Password | Reenter password |
| Location | Select North Europe. |
| Want to use SQL elastic pool | Leave the default **No**. |
| Compute + storage  Backup storage redundancy | Leave the default **General Purpose Gen5, 2 vCores, 32 GB Storage**.  **Locally-redundant backup storage** |
|  |  |

1. In the Networking tab leave the default configuration
2. In the Security tab, leave the default configuration except for the following:

| **Setting** | **Value** |
| --- | --- |
| Enable Microsoft Defender for SQL | Not Now. |

1. Select **Review + create**. You're taken to the **Review + create** page where Azure validates your configuration.
2. When you see the **Validation passed** message, select **Create**.

**Create private endpoint**

In this section, you create a private endpoint for the Azure SQL database in the previous section.

1. In the Azure portal, select **All resources** in the left-hand menu.
2. Select the Azure SQL server **mydbserver1** in the list of services. If you used a different server name, choose that name.
3. In the server **Security** settings, select **Networking** **, Private access** then **Create Private endpoint**
4. In **Create a private endpoint**, enter or select this information in the **Basics** tab:

| **Setting** | **Value** |
| --- | --- |
| **Project details** |  |
| Subscription | Select your subscription. |
| Resource group | Select **myResourceGroup**. |
| **Instance details** |  |
| Name | Enter **SQLPrivateEndpoint**. |
| Region | Select **North Europe.** |

1. In the **Resource** tab, enter or select this information:

| **Setting** | **Value** |
| --- | --- |
| Connection method | Select **Connect to an Azure resource in my directory**. |
| Subscription | Select your subscription. |
| Resource type | Select **Microsoft.Sql/servers**. |
| Resource | Select **mydbserver1** or the name of the server you created in the previous step. |
| Target sub-resource | Select **sqlServer**. |

1. In the **Virtual Network** tab, enter or select this information:

| **Setting** | **Value** |
| --- | --- |
| **Networking** |  |
| Virtual network | Select **myPEVnet**. |
| Subnet | Select **PrivateEndpointSubnet**. |

1. In the **DNS** tab, select this information:

| **Setting** | **Value** |
| --- | --- |
| **Private DNS integration** |  |
| Integrate with private DNS zone | Select **Yes**. |
| Subscription | Select your subscription. |
| Private DNS zones | Leave the default **privatelink.database.windows.net**. |

1. Select the **Review + create** tab or select **Review + create** at the bottom of the page.
2. Select **Create**.
3. After the endpoint is created, select **Firewalls and virtual networks under Security.**
4. In **Firewalls and virtual networks**, select **Yes** next to **Allow Azure services and resources to access this server**.
5. Select **Save**.

**Connect the virtual networks using virtual network peering**

In this section, we'll connect virtual networks **myVMVNet** and **myPEVNet** to **myAzFwVNet** using peering in a hub and spoke topology. There won't be direct connectivity between **myVMVNet** and **myPEVNet**.

1. In the portal's search bar, enter **myAzFwVNet**.
2. Select **Peerings** under **Settings** menu and select **+ Add**.
3. In **Add Peering** enter or select the following information:

| **Setting** | **Value** |
| --- | --- |
| **This virtual network** |  |
| Peering link name | Enter **myAzFwVNet-to-myVMVNet**. |
| Traffic to remote virtual network | Leave the default **Allow**. |
| Traffic forwarded from remote virtual network | Leave the default **Allow**. |
| Virtual network gateway or Route Server | Leave the default **None**. |
| **Remote virtual network** |  |
| Peering link name | Enter **myVMVNet-to-myAzFwVNet**. |
| Virtual network deployment model | Resource manager. |
| Virtual network | Select **myVMVNet.** |
| I know my resource ID | Leave unchecked. |
| Traffic to remote virtual network | Leave the default **Allow**. |
| Traffic forwarded from remote virtual network | Leave the default **Allow**. |
| Virtual network gateway or Route Server | Leave the default **None**. |

1. Select **OK**.
2. Repeat the same steps for the peering with the virtual network **myPEVNet**

**Link the virtual networks to the private DNS zone**

In this section, we'll link virtual networks **myVMVNet** and **myAzFwVNet** to the **privatelink.database.windows.net** private DNS zone. This zone was created when we created the private endpoint.

The link is required for the VM and firewall to resolve the FQDN of database to its private endpoint address. Virtual network **myPEVNet** was automatically linked when the private endpoint was created.

**Note**

If you don't link the VM and firewall virtual networks to the private DNS zone, both the VM and firewall will still be able to resolve the SQL Server FQDN. They will resolve to its public IP address.

1. In the portal's search bar, enter **privatelink.database**.
2. Select **privatelink.database.windows.net** in the search results.
3. Select **Virtual network links** under **Settings**.
4. Select **+ Add**
5. In **Add virtual network link** enter or select the following information:

| **Setting** | **Value** |
| --- | --- |
| Link name | Enter **Link-to-myVMVNet**. |
| **Virtual network details** |  |
| I know the resource ID of virtual network | Leave unchecked. |
| Subscription | Select your subscription. |
| Virtual network | Select **myVMVNet**. |
| **CONFIGURATION** |  |
| Enable auto registration | Leave unchecked. |

1. Select **OK**.
2. Repeat the same steps for **myAzFwVNet** virtual network.

**Configure an application rule with SQL FQDN in Azure Firewall**

In this section, configure an application rule to allow communication between **myVM** and the private endpoint for SQL Server **mydbserver1.database.windows.net**.

This rule allows communication through the firewall that we created in the previous steps.

1. In the portal's search bar, enter **Firewall Policies**.
2. Select **myFirewall-policy**
3. Select the **Application rules** tab.
4. Select **+ Add application rule collection**.
5. In **Add application rule collection** enter or select the following information:

| **Setting** | **Value** |
| --- | --- |
| Name | Enter **SQLPrivateEndpoint**. |
| Priority | Enter **100**. |
| Action | Enter **Allow**. |
| **Rules** |  |
| Name | Enter **SQLPrivateEndpoint**. |
| Source type | Leave the default **IP address**. |
| Source | Enter **10.1.0.0/16**. |
| Destination type | Select **FQDN** |
| Target FQDNs | Enter **mydbserver1.database.windows.net**. |
| Protocol: Port | Enter **mssql:1433**. |
|  |  |
|  |  |

1. Select **Add**.

**Route traffic between the virtual machine and private endpoint through Azure Firewall**

We didn't create a virtual network peering directly between virtual networks **myVMVNet** and **myPEVNet**. The virtual machine **myVM** doesn't have a route to the private endpoint we created.

In this section, we'll create a route table with a custom route. The route sends traffic from the **myVM** subnet to the address space of virtual network **myPEVNet**, through the Azure Firewall.

1. On the Azure portal menu or from the **Home** page, select **Create a resource**.
2. Type **route table** in the search box and press **Enter**.
3. Select **Route table** and then select **Create**.
4. On the **Create Route table** page, use the following table to configure the route table:

| **Setting** | **Value** |
| --- | --- |
| **Project details** |  |
| Subscription | Select your subscription. |
| Resource group | Select **myResourceGroup**. |
| **Instance details** |  |
| Region | Select North Europe. |
| Name | Enter **VMsubnet-to-AzureFirewall**. |
| Propagate gateway routes | Select **No**. |

1. Select **Review + create**. You're taken to the **Review + create** page where Azure validates your configuration.
2. When you see the **Validation passed** message, select **Create**.
3. Once the deployment completes select **Go to resource**.
4. Select **Routes** under **Settings**.
5. Select **+ Add**.
6. On the **Add route** page, enter, or select this information:

| **Setting** | **Value** |
| --- | --- |
| Route name | Enter **myVMsubnet-to-privateendpoint**. |
| Address prefix | Enter **10.2.0.0/16**. |
| Next hop type | Select **Virtual appliance**. |
| Next hop address | Enter **10.0.0.4**. |

1. Select **OK**.
2. Select **Subnets** under **Settings**.
3. Select **+ Associate**.
4. On the **Associate subnet** page, enter or select this information:

| **Setting** | **Value** |
| --- | --- |
| Virtual network | Select **myVMVNet**. |
| Subnet | Select **VMSubnet**. |

1. Select **OK**.

**Connect to the virtual machine from your client computer**

Connect to the VM **myVm** from the internet as follows:

1. In the portal's search bar, enter **myVm-ip**.
2. Select **myVM-ip** in the search results.
3. Copy or write down the value under **IP address**.
4. If you're using Windows 10, run the following command using PowerShell. For other Windows client versions, use an SSH client like [Putty](https://www.putty.org/):

* Replace **username** with the admin username you entered during VM creation.
* Replace **IPaddress** with the IP address from the previous step.

*ssh username@IPaddress*

1. Enter the password you defined when creating myVm

**Access SQL Server privately from the virtual machine**

In this section, you'll connect privately to the SQL Database using the private endpoint.

1. Enter **nslookup mydbserver1.database.windows.net**

You'll receive a message similar to below:

*Server: 127.0.0.53*

*Address: 127.0.0.53#53*

*Non-authoritative answer:*

*mydbserver1.database.windows.net canonical name = mydbserve1r.privatelink.database.windows.net.*

*Name: mydbserver.privatelink.database.windows.net*

*Address: 10.2.0.4*

1. Install [SQL Server command-line tools](https://learn.microsoft.com/en-us/sql/linux/quickstart-install-connect-ubuntu#tools):

Use the following steps to install the **mssql-tools** on Ubuntu. If **curl** isn't installed, you can run this code:

*sudo apt-get update*

*sudo apt install curl*

1. Import the public repository GPG keys.

*curl https://packages.microsoft.com/keys/microsoft.asc | sudo tee /etc/apt/trusted.gpg.d/microsoft.asc*

1. Register the Ubuntu repository.

*curl https://packages.microsoft.com/config/ubuntu/20.04/prod.list | sudo tee /etc/apt/sources.list.d/msprod.list*

1. Update the sources list and run the installation command with the unixODBC developer package. For more information, see [Install the Microsoft ODBC driver for SQL Server (Linux)](https://learn.microsoft.com/en-us/sql/connect/odbc/linux-mac/installing-the-microsoft-odbc-driver-for-sql-server?view=sql-server-ver16)

*sudo apt-get update*

*sudo apt-get install mssql-tools unixodbc-dev*

For convenience, add /opt/mssql-tools/bin/ to your PATH environment variable, to make **sqlcmd** or **bcp** accessible from the bash shell. For non-interactive sessions, modify the PATH environment variable in your ~/.bashrc file with the following command:

*echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bashrc*

*source ~/.bashrc*

1. Run the following command to connect to the SQL Server. Use the server admin and password you defined when you created the SQL Server in the previous steps.

* Replace **<ServerAdmin>** and **<YourPassword>**  with the admin username and the admin password you entered during the SQL server creation.

*sqlcmd -S mydbserver1.database.windows.net -U '<ServerAdmin>' -P '<YourPassword>'*

1. A SQL command prompt will be displayed on successful login. Enter exit to exit the **sqlcmd** tool.
2. Close the connection to **myVM** by entering exit.

**Validate the traffic in Azure Firewall logs**

1. In the Azure portal, select **All Resources** and select your Log Analytics workspace.
2. Select **Logs** under **General** in the Log Analytics workspace page.
3. Select the blue **Get Started** button.
4. In the **Example queries** window, select **Firewalls** under **All Queries**.
5. Select the **Run** button under **Application rule log data**.
6. In the log query output, verify **mydbserver1.database.windows.net** is listed under **FQDN** and **SQLPrivateEndpoint** is listed under **RuleCollection**. Example:

Une image contenant texte

Description générée automatiquement

1. Or deploy the workbook, go to [Azure Monitor Workbook for Azure Firewall](https://github.com/Azure/Azure-Network-Security/tree/master/Azure%20Firewall/Workbook%20-%20Azure%20Firewall%20Monitor%20Workbook) and following the instructions on the page

**Clean up resources**

When you're done using the resources, delete the resource group and all of the resources it contains:

1. Enter **myResourceGroup** in the **Search** box at the top of the portal and select **myResourceGroup** from the search results.
2. Select **Delete resource group**.
3. Enter **myResourceGroup** for **TYPE THE RESOURCE GROUP NAME** and select **Delete**.