✓ 100 XP ▶

Introduction

3 minutes

Suppose you work as a network architect at a global pharmaceutical company that sequences genes to create formulas for proprietary and commercially confidential medicines. These formulas are used by your manufacturing plants around the world. The company wants to implement Microsoft Azure and transfer part of its gene sequencing functions into cloud-based virtual machines. The gene sequencing results need to be available to several regions around the world. Data processing is currently carried out in on-premises datacenters, with virtual private networking (VPN) used to connect the datacenters in each region.

You've been tasked to assess how Microsoft Azure implements networking and to identify whether it will provide suitable security for the transfer of data. Secure data transfer is required between your on-premises datacenter and Microsoft Azure, and between Microsoft Azure regions. You'll use Azure Virtual Network, Azure VPN Gateway, and Azure ExpressRoute technologies.

Learning objectives

By the end of this module, you will be able to:

- Create an Azure Virtual Network
- Create an Azure VPN Gateway
- Identify the features and benefits of Azure ExpressRoute

Prerequisites

- Experience with creating Azure virtual machines
- Thorough understanding of on-premises networking
- Experience with Azure PowerShell, the Azure CLI, and Remote Desktop

The interactive exercises in this module are **not required** to achieve module completion. Completing the exercises requires the following:

- An Azure subscription
- A local computer running Windows 10 with Azure PowerShell installed

ⓘ **Important**

The exercises in this module require a full Azure subscription. The exercises are optional and are not required to complete this module. Participating in the interactive exercises in this module will result in charges billed to the Azure subscription you use to complete them. Incurred charges can be minimized by cleaning up the resources you create as soon as possible. Cleanup directions are included in the final unit.

Next unit: Explore Azure virtual networking

Continue >

✓ 100 XP

Explore Azure virtual networking

10 minutes

You have an on-premises datacenter that you plan to keep, but you want to use Azure to offload peak traffic using virtual machines (VMs) hosted in Azure. You want to keep your existing IP addressing scheme and network appliances, while ensuring that any data transfer is secure.

What is Azure virtual networking?

Azure virtual networks enable Azure resources, such as virtual machines, web apps, and databases, to communicate with: each other, users on the Internet, and on-premises client computers. You can think of an Azure network as a set of resources that links other Azure resources.

Azure virtual networks provide key networking capabilities:

- Isolation and segmentation
- Internet communications
- Communicate between Azure resources
- Communicate with on-premises resources
- Route network traffic
- Filter network traffic
- Connect virtual networks

Network configurations for virtual machines

Isolation and segmentation

Azure allows you to create multiple isolated virtual networks. When you set up a virtual network, you define a private Internet Protocol (IP) address space, using either public or private IP address ranges. You can then segment that IP address space into subnets, and allocate part of the defined address space to each named subnet.

For name resolution, you can use the name resolution service that's built in to Azure, or you can configure the virtual network to use either an internal or an external Domain Name System (DNS) server.

Internet communications

A VM in Azure can connect out to the Internet by default. You can enable incoming connections from the Internet by defining a public IP address or a public load balancer. For VM management, you can connect via the Azure CLI, Remote Desktop Protocol (RDP), or Secure Shell (SSH).

Communicate between Azure resources

You'll want to enable Azure resources to communicate securely with each other. You can do that in one of two ways:

- **Virtual networks**

  Virtual networks can connect not only VMs, but other Azure resources, such as the App Service Environment, Azure Kubernetes Service, and Azure virtual machine scale sets.

- **Service endpoints**

  You can use service endpoints to connect to other Azure resource types, such as Azure SQL databases and storage accounts. This approach enables you to link multiple Azure resources to virtual networks, thereby improving security and providing optimal routing between resources.

Communicate with on-premises resources

Azure virtual networks enable you to link resources together in your on-premises environment and within your Azure subscription, in effect creating a network that spans both your local and cloud environments. There are three mechanisms for you to achieve this connectivity:

- **Point-to-site Virtual Private Networks**

  This approach is like a Virtual Private Network (VPN) connection that a computer outside your organization makes back into your corporate network, except that it's working in the opposite direction. In this case, the client computer initiates an encrypted VPN connection to Azure, connecting that computer to the Azure virtual network.

- **Site-to-site Virtual Private Networks** A site-to-site VPN links your on-premises VPN device or gateway to the Azure VPN gateway in a virtual network. In effect, the devices in Azure can appear as being on the local network. The connection is encrypted and works over the Internet.

- **Azure ExpressRoute**

  For environments where you need greater bandwidth and even higher levels of security, Azure ExpressRoute is the best approach. Azure ExpressRoute provides dedicated private connectivity to Azure that does not travel over the Internet.

Route network traffic

By default, Azure will route traffic between subnets on any connected virtual networks, on-premises networks, and the Internet. However, you can control routing and override those settings as follows:

- **Route tables**

  A route table allows you to define rules as to how traffic should be directed. You can create custom route tables that control how packets are routed between subnets.

- **Border Gateway Protocol**

  Border Gateway Protocol (BGP) works with Azure VPN gateways or ExpressRoute to propagate on-premises BGP routes to Azure virtual networks.

Filter network traffic

Azure virtual networks enable you to filter traffic between subnets by using the following approaches:

- **Network security groups**

  A network security group is an Azure resource that can contain multiple inbound and outbound security rules. You can define these rules to allow or block traffic, based on factors such as source and destination IP address, port, and protocol.

- **Network virtual appliances**

  A network virtual appliance is a specialized VM that can be compared to a hardened network appliance. A network virtual appliance carries out a particular network function, such as running a firewall or performing WAN optimization.

Connect virtual networks

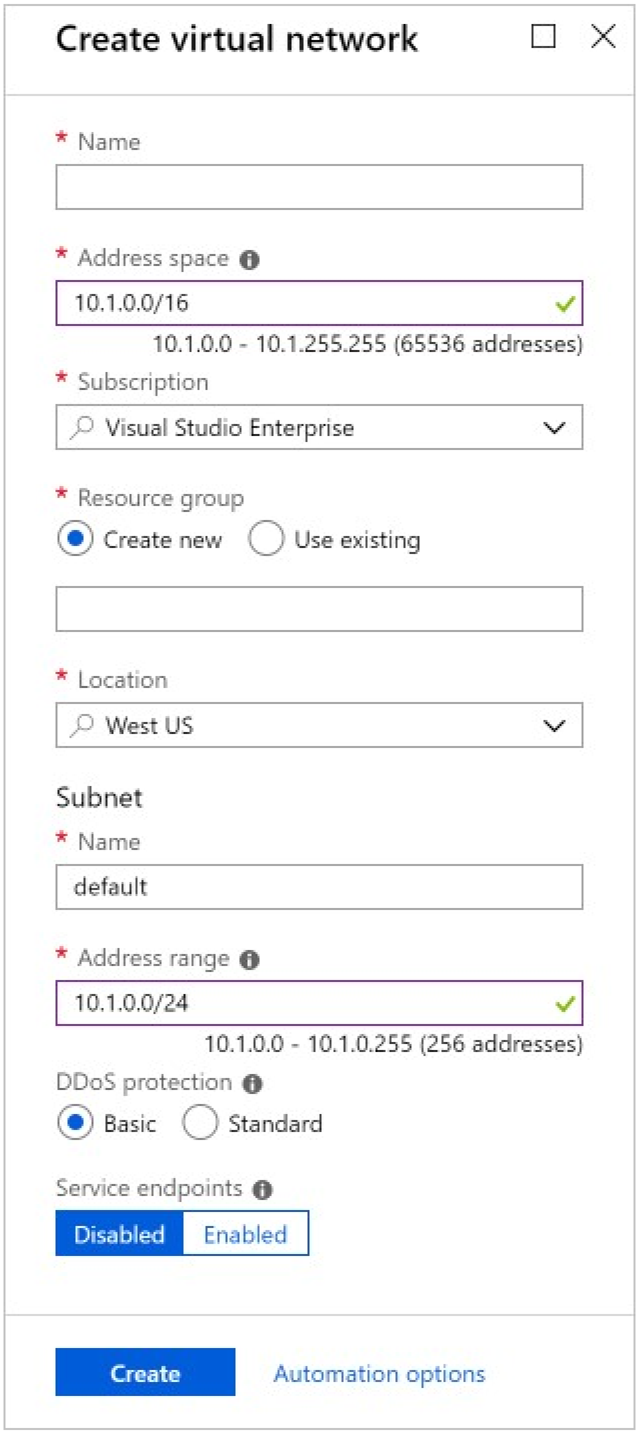
You can link virtual networks together using virtual network *peering*. Peering enables resources in each virtual network to communicate with each other. These virtual networks can be in separate regions, allowing you to create a global interconnected network through Azure.

Azure virtual network settings

You can create and configure Azure virtual networks from the Azure portal, Azure PowerShell on your local computer, or Azure Cloud Shell.

Create a virtual network

When you create an Azure virtual network, you configure a number of basic settings. You'll have the option to configure advanced settings, such as multiple subnets, distributed denial of service (DDoS) protection, and service endpoints.



Create virtual network

* Name

* Address space
10.1.0.0/16 ✓
10.1.0.0 - 10.1.255.255 (65536 addresses)

* Subscription
Visual Studio Enterprise

* Resource group
● Create new ○ Use existing

* Location
West US

Subnet

* Name
default

* Address range
10.1.0.0/24 ✓
10.1.0.0 - 10.1.0.255 (256 addresses)

DDoS protection
● Basic ○ Standard

Service endpoints
Disabled | Enabled

Create | Automation options

You'll configure the following settings for a basic virtual network:

- **Network name**

  The network name must be unique in your subscription, but does not need to be globally unique. Make the name a descriptive one that is easy to remember and identified from other virtual networks.

- **Address space**

  When you set up a virtual network, you define the internal address space in Classless Inter-Domain Routing (CIDR) format. This address space needs to be unique within your subscription and any other networks that you connect to.

  Let's assume, you choose an address space of 10.0.0.0/24 for your first virtual network. The addresses defined in this address space ranges from 10.0.0.1 - 10.0.0.254. You then create a second virtual network and choose an address space of 10.1.0.0./8. The address in this address space ranges from 10.0.0.1 - 10.255.255.254. Some of the address overlap and can't be used for the two virtual networks.

  However, you can use 10.0.0.0/16, with addresses ranging from 10.0.0.1 - 10.0.255.254, and 10.1.0.0/16, with addresses ranging from 10.1.0.1 - 10.1.255.254. You can assign these address spaces to your virtual networks since there's no address overlap.

  > **Note**
  >
  > You can add address spaces after creating the virtual network.

- **Subscription**

  Only applies if you have multiple subscriptions to choose from.

- **Resource group**

  Like any other Azure resource, a virtual network needs to exist in a resource group. You can either select an existing resource group or create a new one.

- **Location**

  Select the location where you want the virtual network to exist.

- **Subnet**

  Within each virtual network address range, you can create one or more subnets that partition the virtual network's address space. Routing between subnets will then depend on the default traffic routes, or you can define custom routes. Alternatively, you can define one subnet that encompasses all the virtual networks' address ranges.

> ⓘ **Note**
>
> Subnet names must begin with a letter or number, end with a letter, number or underscore, and may contain only letters, numbers, underscores, periods, or hyphens.

- **Distributed Denial of Service (DDoS) protection**

  You can select either Basic or Standard DDoS protection. Standard DDoS Protection is a premium service. The Azure DDoS Protection Standard provides for more information on Standard DDoS protection.

- **Service Endpoints**

  Here, you enable service endpoints, and then select from the list which Azure service endpoints you want to enable. Options include Azure Cosmos DB, Azure Service Bus, Azure Key Vault, and so on.

When you have configured these settings, click the **Create** button.

Define additional settings

After creating a virtual network, you can then define further settings. These include:

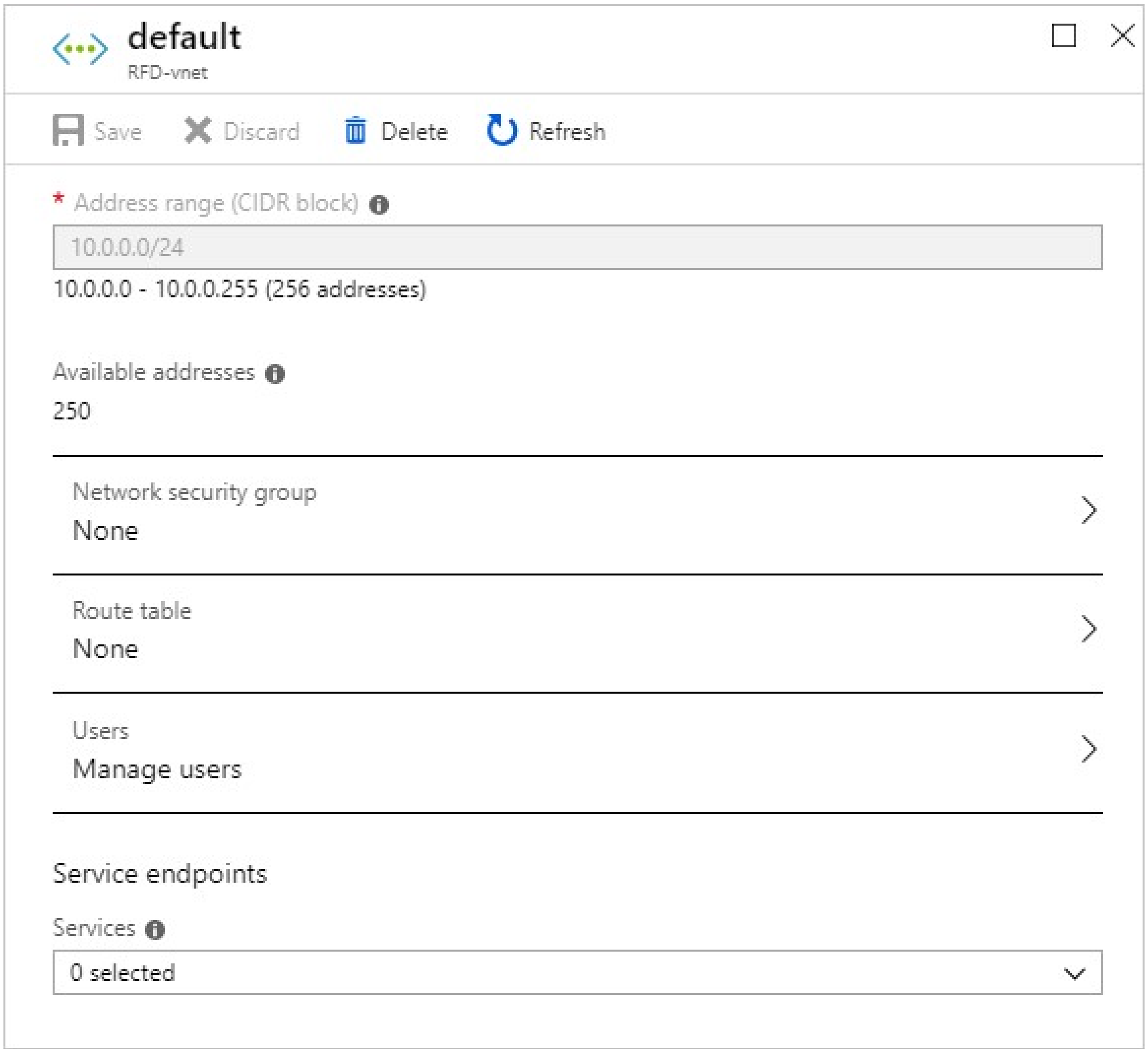
- **Network security group**

  Network security groups have security rules that enable you to filter the type of network traffic that can flow in and out of virtual network subnets and network interfaces. You create the network security group separately, and then associate it with the virtual network.

- **Route table**

> Azure automatically creates a route table for each subnet within an Azure virtual network and adds system default routes to the table. However, you can add custom route tables to modify traffic between virtual networks.

You can also amend the service endpoints.



default
RFD-vnet

* Address range (CIDR block)
10.0.0.0/24
10.0.0.0 - 10.0.0.255 (256 addresses)

Available addresses
250

Network security group
None

Route table
None

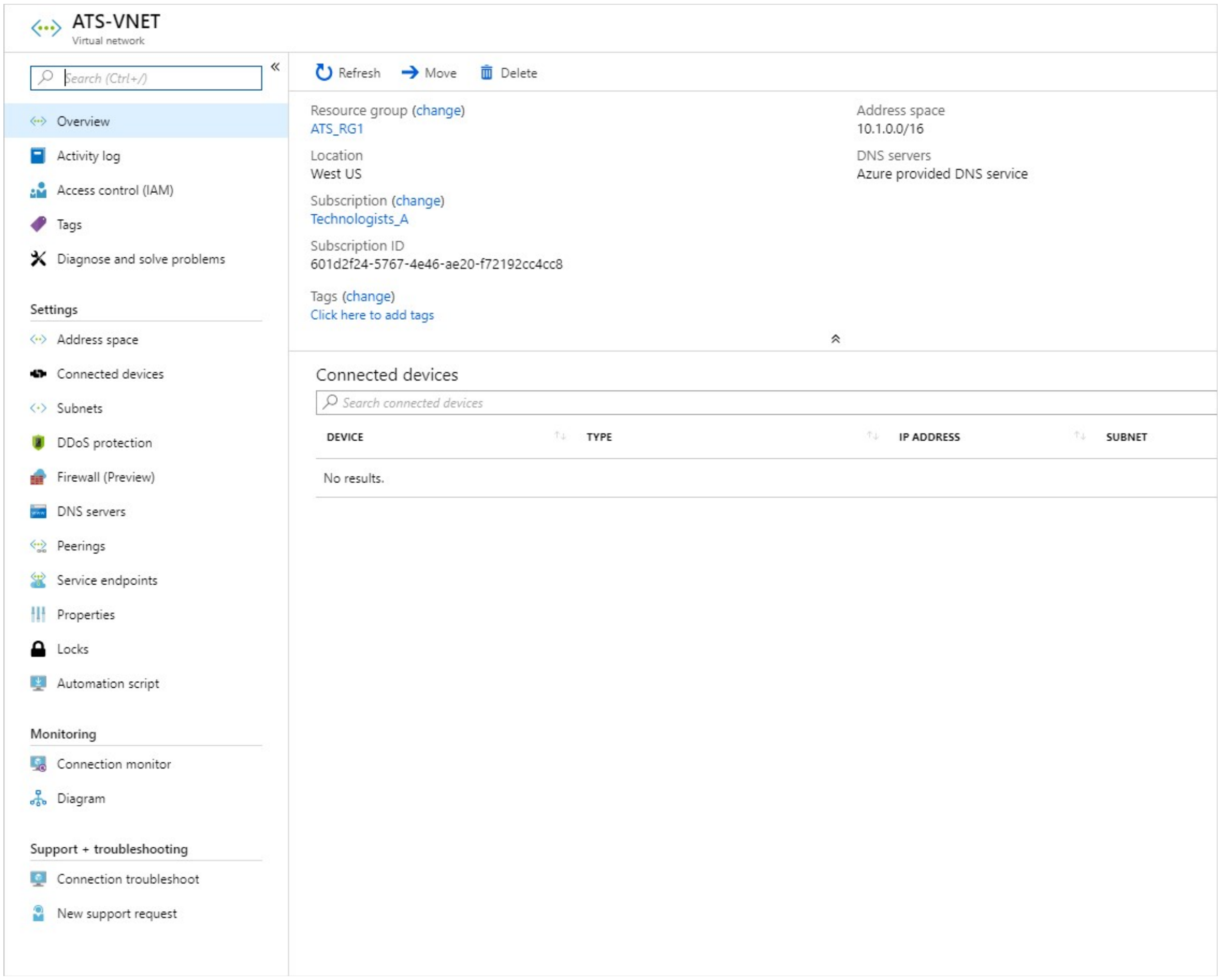
Users
Manage users

Service endpoints

Services
0 selected

Configure virtual networks

When you have created a virtual network, you can change any further settings from the Virtual Networks pane in the Azure portal. Alternatively, you can use PowerShell commands or commands in Cloud Shell to make changes.



Resource group (change)
ATS_RG1

Location
West US

Subscription (change)
Technologists_A

Subscription ID
601d2f24-5767-4e46-ae20-f72192cc4cc8

Tags (change)
Click here to add tags

Address space
10.1.0.0/16

DNS servers
Azure provided DNS service

Connected devices

Search connected devices

DEVICE	TYPE	IP ADDRESS	SUBNET
No results.			

You can then review and change settings in further sub-panes. These settings include:

- Address spaces: You can add further address spaces to the initial definition
- Connected devices: Use the virtual network to connect machines
- Subnets: Add further subnets
- Peerings: Link virtual networks in peering arrangements

You can also monitor and troubleshoot virtual networks, or create an automation script to generate the current virtual network.

Virtual networks are powerful and highly configurable mechanisms for connecting entities in Azure. You can connect Azure resources to one another or to resources you have on-premises. You can isolate, filter and route your network traffic, and Azure allows you to increase security where you feel you need it.

Next unit: Exercise - Create an Azure virtual network

Continue >

✓ 100 XP

Exercise - Create an Azure virtual network

20 minutes

In this exercise, you will create a virtual network in Microsoft Azure. You will then create two virtual machines and use the virtual network to connect the virtual machines and to the Internet.

> **Important**
>
> The exercises in this module require a full Azure subscription. The exercises are optional and are not required to complete this module. Participating in the interactive exercises in this module will result in charges billed to the Azure subscription you use to complete them. Incurred charges can be minimized by cleaning up the resources you create as soon as possible. Cleanup directions are in the final unit.

Sign in to your subscription

Sign in to Azure either by using the PowerShell cmdlet `Connect-AzAccount` on your local machine or by using shell.azure.com/powershell.

Create a resource group

First, create a resource group to contain all of the resources you'll create in this module. Name it `vm-networks` and replace `EastUS` in the following command with the name of the region in which you'd like the group to be created.

PowerShell Copy

```
$Location = "EastUS"
New-AzResourceGroup -Name vm-networks -Location $Location
```

Create a subnet and virtual network

To create a subnet and virtual network, run the following command.

PowerShell Copy

```
 $Subnet= New-AzVirtualNetworkSubnetConfig -Name default -AddressPrefix 10.0.0.0/24
 New-AzVirtualNetwork -Name myVnet -ResourceGroupName vm-networks -Location $Location -AddressPrefix 10.0.0.0/16 -Subnet $Subnet
```

Create two virtual machines

All Azure virtual machines are connected to a virtual network. If you create a virtual machine using Azure PowerShell and don't specify the name of an existing virtual network, Azure PowerShell creates a new virtual network automatically.

Here, we create two virtual machines and specify the virtual network.

1. To create the first virtual machine, run the following command to create a Windows VM with a public IP address that is accessible over port 3389 (Remote Desktop). This creates a Windows 2016 Datacenter VM named `dataProcStage1` that uses the myVnet virtual network.

   PowerShell Copy

   ```
   New-AzVm `
    -ResourceGroupName "vm-networks" `
    -Name "dataProcStage1" `
    -VirtualNetworkName "myVnet" `
    -SubnetName "default" `
    -image "Win2016Datacenter" `
    -Size "Standard_DS2_v2"
   ```

Port 3389 is opened automatically by default when you create a Windows VM in Azure.

2. Enter a user name and password for the VM. Write down user name and password. You need it later to sign in to the server.

3. Run the following command to get the public IP address for your VM so you can use it later. Copy the **IpAddress**.

   PowerShell Copy

   ```
   Get-AzPublicIpAddress -Name dataProcStage1
   ```

4. Create the second VM named `dataProcStage2`.

   PowerShell Copy

   ```
   New-AzVm `
    -ResourceGroupName "vm-networks" `
    -Name "dataProcStage2" `
    -VirtualNetworkName "myVnet" `
    -SubnetName "default" `
    -image "Win2016Datacenter" `
    -Size "Standard_DS2_v2"
   ```

5. Enter a user name and password for the VM. Write down user name and password. You need it later to sign in to the server.

6. Disassociate the public IP address that was created by default for the VM.

   PowerShell Copy

   ```
   $nic = Get-AzNetworkInterface -Name dataProcStage2 -ResourceGroup vm-networks
   $nic.IpConfigurations.publicipaddress.id = $null
   Set-AzNetworkInterface -NetworkInterface $nic
   ```

Connect to dataProcStage1 using Remote Desktop

1. Open Remote Desktop and connect to `dataProcStage1` with the public IP address you noted from the previous steps. If you're using PowerShell locally, use the following command and replace `publicIpAddress` with the VM's public IP address.

   PowerShell Copy

   ```PowerShell
   mstsc /v:publicIpAddress
   ```

2. Sign in to the remote machine with the username and the password you created.

3. In the remote session, open the Windows command prompt and run the following command:

   cmd Copy

   ```cmd
   ping dataProcStage2 -4
   ```

4. In the results, you'll see that all requests to `dataProcStage2` time out. This is because the default Windows Firewall configuration on `dataProcStage2` prevents it from responding to pings.

Connect to dataProcStage2 using Remote Desktop

Configure the Windows Firewall on `dataProcStage2` using a new remote desktop session. You can't access `dataProcStage2` from your desktop because `dataProcStage2` doesn't have a public IP address. You'll use remote desktop from `dataProcStage1` to connect to `dataProcStage2`.

1. In the `dataProcStage1` remote session, open Remote Desktop.

2. Connect to `dataProcStage2` by name. Based on the default network configuration, `dataProcStage1` can resolve the address for `dataProcStage2` using the computer name.

3. Sign in to `dataProcStage2` with the username and the password you created.

4. On `dataProcStage2`, click the Start Menu, type **Firewall**, and press Enter. The **Windows Firewall with Advanced Security** console appears.

5. In the left-hand pane, click **Inbound Rules**.

6. In the right-hand pane, scroll down, and right-click **File and Printer Sharing (Echo Request - ICMPv4-In)**, and then click **Enable Rule**.

7. Switch back to the `dataProcStage1` remote session and run the following command in the command prompt.

   cmd Copy

   ```cmd
   ping dataProcStage2 -4
   ```

8. `dataProcStage2` responds with four replies, demonstrating connectivity between the two VMs.

You successfully created a virtual network, created two VMs that are attached to that virtual network, connected to one of the VMs and shown network connectivity to the other VM within the same virtual network. You can use Azure Virtual Network to connect resources within the Azure network. However, those resources need to be within the same resource group and subscription. Next, we will look at VPN gateways, which enable you to connect virtual network in different resource groups, subscriptions, and even geographical regions.

Next unit: Explore Azure VPN Gateway

Continue >

✓ 100 XP

Explore Azure VPN Gateway

10 minutes

To integrate your on-premises environment with Azure, you need the ability to create an encrypted connection. You can connect over the Internet or over a dedicated link. Here, we'll look at Azure VPN Gateway, which provides an endpoint for incoming connections from on-premises environments.

You have set up an Azure virtual network and need to ensure that any data transfers from Azure to your site and between Azure virtual networks are encrypted. You also need to know how to connect virtual networks between regions and subscriptions.

What is a VPN gateway?

An Azure VPN gateway provides an endpoint for incoming encrypted connections from on-premises locations to Azure over the Internet. It can also send encrypted traffic between Azure virtual networks over Microsoft's dedicated network that links Azure datacenters in different regions. This configuration allows you to link virtual machines and services in different regions securely.

Each virtual network can have only one VPN gateway. All connections to that VPN gateway share the available network bandwidth.

Within each virtual network gateway there are two or more virtual machines (VMs). These VMs have been deployed to a special subnet that you specify, called the *gateway subnet*. They contain routing tables for connections to other networks, along with specific gateway services. These VMs and the gateway subnet are similar to a hardened network device. You don't need to configure these VMs directly and should not deploy any additional resources into the gateway subnet.

Creating a virtual network gateway can take some time to complete, so it's vital that you plan appropriately. When you create a virtual network gateway, the provisioning process generates the gateway VMs and deploys them to the gateway subnet. These VMs will have the settings that you configure on the gateway.

A key setting is the ***gateway type***, which for a VPN gateway will be of type "vpn". Options for VPN gateways include:

- Network-to-network connections over IPsec/IKE VPN tunneling, linking VPN gateways to other VPN gateways.

- Cross-premises IPsec/IKE VPN tunneling, for connecting on-premises networks to Azure through dedicated VPN devices to create site-to-site connections.

- Point-to-site connections over IKEv2 or SSTP, to link client computers to resources in Azure.

Now, let's look at the factors you need to consider for planning your VPN gateway.

Plan a VPN gateway

When you're planning a VPN gateway, there are three architectures to consider:

- Point to site over the Internet
- Site to site over the Internet
- Site to site over a dedicated network, such as Azure ExpressRoute

Planning factors

Factors that you need to cover during your planning process include:

- Throughput - Mbps or Gbps
- Backbone - Internet or private?
- Availability of a public (static) IP address
- VPN device compatibility
- Multiple client connections or a site-to-site link?
- VPN gateway type
- Azure VPN Gateway SKU

The following table summarizes some of these planning issues. The remainder are discussed later.

	Point to site	Site to site	ExpressRoute
Azure supported services	Cloud services and VMs	Cloud services and VMs	All supported services
Typical bandwidth	Depends on VPN Gateway SKU	Up to 1 Gbps with aggregation	From 50 Mbps to 10 Gbps
Protocols supported	SSTP and IPsec	IPsec	Direct connection, VLANs
Routing	RouteBased (dynamic)	PolicyBased (static) and RouteBased	BGP
Connection resiliency	Active-passive	Active-passive or active-active	Active-active
Use case	Testing and prototyping	Dev, test and small-scale production	Enterprise/mission critical

Gateway SKUs

Azure offers the following SKUs for gateway services:

SKU	S2S/network-to-network tunnels	P2S connections	Aggregate throughput benchmark	Use for
Basic	Max 10	Max 128	100 Mbps	Dev/test/POC
VpnGw1	Max 30	Max 128	650 Mbps	Production/critical workloads
VpnGw2	Max 30	Max 128	1 Gbps	Production/critical workloads
VpnGw3	Max 30	Max 128	1.25 Gbps	

SKU	S2S/network-to-network tunnels	P2S connections	Aggregate throughput benchmark	Use for
				Production/critical workloads

> **Note**
>
> It's important that you choose the right SKU. If you have set up your VPN gateway with the wrong one, you'll have to take it down and rebuild the gateway, which can be time consuming.

Workflow

When designing a cloud connectivity strategy using virtual private networking on Azure, you should apply the following workflow:

1. Design your connectivity topology, listing the address spaces for all connecting networks.
2. Create an Azure virtual network.
3. Create a VPN gateway for the virtual network.
4. Create and configure connections to on-premises networks or other virtual networks, as required.
5. If required, create and configure a point-to-site connection for your Azure VPN gateway.

Design considerations

When you design your VPN gateways to connect virtual networks, you must consider the following factors:

- Subnets cannot overlap

It is vital that a subnet in one location does not contain the same address space as in another location.

- IP addresses must be unique

  You cannot have two hosts with the same IP address in different locations, as it will be impossible to route traffic between those two hosts and the network-to-network connection will fail.

- VPN gateways need a gateway subnet called **GatewaySubnet**

  It must have this name for the gateway to work, and it should not contain any other resources.

Create an Azure virtual network

Before you create a VPN gateway, you need to create the Azure virtual network.

Create a VPN gateway

The type of VPN gateway you create will depend on your architecture. Options are:

- RouteBased

  Route-based VPN devices use any-to-any (wildcard) traffic selectors, and let routing/forwarding tables direct traffic to different IPsec tunnels. Route-based connections are typically built on router platforms where each IPsec tunnel is modeled as a network interface or VTI (virtual tunnel interface).

- PolicyBased

  Policy-based VPN devices use the combinations of prefixes from both networks to define how traffic is encrypted/decrypted through IPsec tunnels. A policy-based connection is typically built on firewall devices that perform packet filtering. IPsec tunnel encryption and decryption are added to the packet filtering and processing engine.

Set up a VPN gateway

The steps you need to take will depend on the type of VPN gateway that you are installing. For example, to create a point-to-site VPN gateway by using the Azure portal, you would carry out the following steps:

1. Create a virtual network
2. Add a gateway subnet
3. Specify a DNS server (optional)
4. Create a virtual network gateway
5. Generate certificates
6. Add the client address pool
7. Configure the tunnel type
8. Configure the authentication type
9. Upload the root certificate public certificate data
10. Install an exported client certificate
11. Generate and install the VPN client configuration package
12. Connect to Azure

As there are several configuration paths with Azure VPN gateways, each with multiple options, it is not possible to cover every setup in this course. For more information, see the Additional Resources section.

Configure the gateway

Once your gateway is created, you'll need to configure it. There are several configuration settings you will need to provide, such as the name, location, DNS server, etc. We will go into these in more detail in the exercise.

Azure VPN gateways are a component in Azure virtual networks that enable point-to-site, site-to-site, or network-to-network connections. Azure VPN gateways enable individual client computers to connect to resources in Azure, extend on-premises networks into Azure, or facilitate connections between virtual networks in different regions and subscriptions.

Next unit: Exercise - Create an Azure VPN gateway

Continue >

✓ 100 XP ▶

Exercise - Create an Azure VPN gateway

40 minutes

You want to ensure that you can connect clients or sites within your environment into Azure using encrypted tunnels across the public Internet. In this unit, you'll create a point-to-site VPN gateway, and then connect to that gateway from your client computer. You'll use native Azure certificate authentication connections for security.

You will carry out the following process:

1. Create a RouteBased VPN gateway.
2. Upload the public key for a root certificate for authentication purposes.
3. Generate a client certificate from the root certificate, and then install the client certificate on each client computer that will connect to the virtual network for authentication purposes.
4. Create VPN client configuration files, which contain the necessary information for the client to connect to the virtual network.

Setup

To complete this module, use Azure PowerShell from your local Windows 10 computer.

1. Open a new PowerShell session on your local Windows 10 computer where you have the Azure PowerShell module installed.
2. Sign in to Azure by using the PowerShell cmdlet `Connect-AzAccount`.
3. Set up variables you'll use to create a virtual network. Copy and paste in the following variables into PowerShell.

PowerShell Copy

```
$VNetName  = "VNetData"
$FESubName = "FrontEnd"
$BESubName = "Backend"
$GWSubName = "GatewaySubnet"
$VNetPrefix1 = "192.168.0.0/16"
$VNetPrefix2 = "10.254.0.0/16"
$FESubPrefix = "192.168.1.0/24"
$BESubPrefix = "10.254.1.0/24"
$GWSubPrefix = "192.168.200.0/26"
$VPNClientAddressPool = "172.16.201.0/24"
$ResourceGroup = "VpnGatewayDemo"
$Location = "East US"
$GWName = "VNetDataGW"
$GWIPName = "VNetDataGWPIP"
$GWIPconfName = "gwipconf"
```

Configure a virtual network

1. Run the following command to create a resource group.

   PowerShell Copy

   ```
   New-AzResourceGroup -Name $ResourceGroup -Location $Location
   ```

2. Run the following command to create subnet configurations for the virtual network. These have the name **FrontEnd, BackEnd**, and **GatewaySubnet**. All of these subnets exist within the virtual network prefix.

   PowerShell Copy

   ```
   $fesub = New-AzVirtualNetworkSubnetConfig -Name $FESubName -AddressPrefix $FESubPrefix
   $besub = New-AzVirtualNetworkSubnetConfig -Name $BESubName -AddressPrefix $BESubPrefix
   $gwsub = New-AzVirtualNetworkSubnetConfig -Name $GWSubName -AddressPrefix $GWSubPrefix
   ```

3. Next, run the following command to create the virtual network using the subnet values and a static DNS server.

PowerShell

Copy

```powershell
New-AzVirtualNetwork -Name $VNetName -ResourceGroupName $ResourceGroup
-Location $Location -AddressPrefix $VNetPrefix1,$VNetPrefix2 -Subnet
$fesub, $besub, $gwsub -DnsServer 10.2.1.3
```

4. Now specify the variables for this network that you have just created.

PowerShell

Copy

```powershell
$vnet = Get-AzVirtualNetwork -Name $VNetName -ResourceGroupName $ResourceGroup
$subnet = Get-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet
```

5. Run the following command to request a dynamically assigned public IP address.

PowerShell

Copy

```powershell
$pip = New-AzPublicIpAddress -Name $GWIPName -ResourceGroupName $ResourceGroup -Location $Location -AllocationMethod Dynamic
$ipconf = New-AzVirtualNetworkGatewayIpConfig -Name $GWIPconfName -Subnet $subnet -PublicIpAddress $pip
```

Create the VPN gateway

When creating this VPN gateway:

- GatewayType must be Vpn
- VpnType must be RouteBased

Note that this part of the exercise can take up to 45 minutes to complete.

1. To create the VPN gateway, run the following command and press Enter.

PowerShell

Copy

```powershell
New-AzVirtualNetworkGateway -Name $GWName -ResourceGroupName $ResourceGroup `
-Location $Location -IpConfigurations $ipconf -GatewayType Vpn `
```

```
-VpnType RouteBased -EnableBgp $false -GatewaySku VpnGw1 -VpnClientProtocol "IKEv2"
```

2. Wait for the command output to appear.

Add the VPN client address pool

1. Run the following command to add the VPN client address pool.

```powershell
$Gateway = Get-AzVirtualNetworkGateway -ResourceGroupName $ResourceGroup -Name $GWName
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $Gateway -VpnClientAddressPool $VPNClientAddressPool
```

2. Wait for the command output to appear.

Generate a client certificate

With the network infrastructure created on Azure, we need to create a self-signed client certificate on our local machine. This can be done similarly on most operating systems, but we will cover how to generate your client certificate on Windows 10 using PowerShell with the Azure PowerShell module and the Windows **Certificate Manager** utility.

1. Our first step is to create the self-signed root certificate. Run the following command.

```powershell
$cert = New-SelfSignedCertificate -Type Custom -KeySpec Signature `
-Subject "CN=P2SRootCert" -KeyExportPolicy Exportable `
-HashAlgorithm sha256 -KeyLength 2048 `
-CertStoreLocation "Cert:\CurrentUser\My" -KeyUsageProperty Sign -KeyUsage CertSign
```

2. Next, generate a client certificate signed by your new root certificate.

PowerShell

Copy

```
New-SelfSignedCertificate -Type Custom -DnsName P2SChildCert -KeySpec
Signature `
-Subject "CN=P2SChildCert" -KeyExportPolicy Exportable `
-HashAlgorithm sha256 -KeyLength 2048 `
-CertStoreLocation "Cert:\CurrentUser\My" `
-Signer $cert -TextExtension @("2.5.29.37={text}1.3.6.1.5.5.7.3.2")
```

Export certificate public key

With our certificates generated, we need to export our root certificate's public key.

1. Run `certmgr` from PowerShell to open the Certificate Manager.
2. Go to **Personal** > **Certificates**.
3. Right-click the **P2SRootCert** certificate in the list and select **All tasks** > **Export...**.
4. In the Certificate Export Wizard, click **Next**.
5. Ensure that **No, do not export the private key** is selected, and then click **Next**.
6. On the **Export File Format** page, ensure that **Base-64 encoded X.509 (.CER)** is selected, and then click **Next**.
7. In the **File to Export** page, under **File name**, navigate to a location you'll remember and save the file as **P2SRootCert.cer**, and then click Next.
8. On the **Completing the Certificate Export Wizard** page, click **Finish**.
9. On the **Certificate Export Wizard** message box, click **OK**.

Upload the root certificate public key information

1. In the PowerShell window, run the following command to declare a variable for the certificate name:

PowerShell Copy

```
$P2SRootCertName = "P2SRootCert.cer"
```

2. Replace the `<cert-path>` placeholder with the export location of your root certificate and run the following command:

PowerShell Copy

```
$filePathForCert = "<cert-path>\P2SRootCert.cer"
$cert = new-object System.Security.Cryptography.X509Certificates.X509Certificate2($filePathForCert)
$CertBase64 = [system.convert]::ToBase64String($cert.RawData)
$p2srootcert = New-AzVpnClientRootCertificate -Name $P2SRootCertName -PublicCertData $CertBase64
```

3. With the group name set, upload the certificate to Azure with the following command.

PowerShell Copy

```
Add-AzVpnClientRootCertificate -VpnClientRootCertificateName $P2SRootCertName -VirtualNetworkGatewayname $GWName -ResourceGroupName $ResourceGroup -PublicCertData $CertBase64
```

Azure will now recognize this certificate as a trusted root certificate for our virtual network.

Configure the native VPN client

1. Run the following command to create VPN client configuration files in .ZIP format.

PowerShell Copy

```
$profile = New-AzVpnClientConfiguration -ResourceGroupName $ResourceGroup -Name $GWName -AuthenticationMethod "EapTls"
$profile.VPNProfileSASUrl
```

2. Copy the URL returned in the output from this command and paste it into your browser. Your browser should start downloading a .ZIP file. Extract the archive contents and put them in a suitable location.

   Some browsers will initially attempt to block downloading this ZIP file as a dangerous download. You will need to override this in your browser to be able to extract the archive contents.

3. In the extracted folder, navigate to either the **WindowsAmd64** folder (for 64-bit Windows computers) or the **WindowsX86** folder (for 32-bit computers).

   If you want to configure a VPN on a non-Windows machine, you can use the certificate and settings files from the **Generic** folder.

4. Double-click on the **VpnClientSetup{architecture}.exe** file, with `{architecture}` reflecting your architecture.

5. In the **Windows protected your PC** screen, click **More info**, and then click **Run anyway**.

6. In the **User Account Control** dialog box, click **Yes**.

7. In the **VNetData** dialog box, click **Yes**.

Connect to Azure

1. Press the Windows key, type **Settings** and press Enter.

2. In the **Settings** window, click **Network and Internet**.

3. In the left-hand pane, click **VPN**.

4. In the right-hand pane, click **VNetData**, and then click **Connect**.

5. In the VNetData window, click **Connect**.

6. In the next VNetData window, click **Continue**.

7. In the **User Account Control** message box, click **Yes**.

If these steps do not work, you may need to restart your computer.

Verify your connection

1. In a new Windows command prompt, run `IPCONFIG /ALL`.
2. Copy the IP address under PPP adapter VNetData, or write it down.
3. Confirm that IP address is in the **VPNClientAddressPool range of 172.16.201.0/24**.
4. You have successfully made a connection to the Azure VPN gateway.

You just set up a VPN gateway, allowing you to make an encrypted client connection to a virtual network in Azure. This approach is great with client computers and smaller site-to-site connections.

Next unit: Explore Azure ExpressRoute

Continue >

✓ 100 XP ▶

Explore Azure ExpressRoute

5 minutes

As your company deals with highly sensitive data and has large amounts of information it will store in Azure, there are some concerns about the security and reliability of connections over the public Internet. The company isn't willing to migrate wholesale to Azure unless it can demonstrate higher levels of connectivity, security, and reliability.

Here, we'll go beyond connections that run over the Internet to dedicated lines direct into the Azure datacenters.

Azure ExpressRoute

Microsoft Azure ExpressRoute enables organizations to extend their on-premises networks into the Microsoft Cloud over a private connection implemented by a connectivity provider. This arrangement means that the connectivity to the Azure datacenters doesn't go over the Internet but across a dedicated link. ExpressRoute also facilitates efficient connections with other Microsoft cloud-based services, such as Office 365 and Dynamics 365.

Advantages that ExpressRoute provides include:

- Faster speeds, from 50 Mbps to 10 Gbps, with dynamic bandwidth scaling
- Lower latency
- Greater reliability through built-in peering
- Highly secure

ExpressRoute brings a number of further benefits, such as:

- Connectivity to all supported Azure services
- Global connectivity to all regions (requires premium add-on)

- Dynamic routing over Border Gateway Protocol
- Service-level agreements (SLAs) for connection uptime
- Quality of Service (QoS) for Skype for Business

Additionally, there's the ExpressRoute premium add-on, which offers benefits such as increased route limits, global service connectivity, and increased virtual network links per circuit.

ExpressRoute connectivity models

Connections into ExpressRoute can be through the following mechanisms:

- IP VPN network (any-to-any)
- Virtual cross-connection through an Ethernet exchange
- Point-to-point Ethernet connection

ExpressRoute capabilities and features are all identical across all of the above connectivity models.

What is layer 3 connectivity?

Microsoft uses an industry-standard dynamic routing protocol (BGP) to exchange routes between your on-premises network, your instances in Azure, and Microsoft public addresses. We establish multiple BGP sessions with your network for different traffic profiles.

Any-to-any (IPVPN) networks

IPVPN providers typically provide connectivity between branch offices and your corporate datacenter over managed layer 3 connections. With ExpressRoute, the Azure datacenters appear as if they were another branch office.

Virtual cross-connection through an Ethernet Exchange

If your organization is co-located with a cloud exchange facility, you request cross-connections to the Microsoft Cloud through your provider's Ethernet exchange. These cross-connections to the Microsoft Cloud can operate at either layer 2 or layer 3 managed connections, as in the networking OSI model.

Point-to-point Ethernet connection

Point-to-point Ethernet links can provide layer 2 or managed layer 3 connections between your on-premises datacenters or offices to the Microsoft Cloud.

How ExpressRoute works

Azure ExpressRoute uses a combination of ExpressRoute circuits and routing domains to provide high-bandwidth connectivity to the Microsoft Cloud.

What are ExpressRoute circuits

An ExpressRoute circuit is the logical connection between your on-premises infrastructure and the Microsoft Cloud. A connectivity provider implements that connection, although some organizations use multiple connectivity providers for redundancy reasons. Each circuit has a fixed bandwidth of either 50, 100, 200 Mbps or 500 Mbps, or 1 Gbps or 10 Gbps, and each of those circuits map to a connectivity provider and a peering location. In addition, each ExpressRoute circuit has default quotas and limits.

An ExpressRoute circuit isn't equivalent to a network connection or a network device. Each circuit is defined by a GUID, called a *service* or *s-key*. This s-key provides the connectivity link between Microsoft, your connectivity provider, and your organization - it isn't a cryptographic secret. Each s-key has a one-to-one mapping to an Azure ExpressRoute circuit.

Each circuit can have up to three peerings, which are a pair of BGP sessions that are configured for redundancy. They are:

- Azure private

- Azure public
- Microsoft

Routing domains

ExpressRoute circuits then map to routing domains, with each ExpressRoute circuit having multiple routing domains. These domains are the same as the three peerings listed above. In an active-active configuration, each pair of routers would have each routing domain configured identically, thus providing high availability. The Azure public and Azure private peering names represent the IP addressing schemes.

Azure private peering

Azure private peering connects to Azure compute services such as virtual machines and cloud services that are deployed with a virtual network. As far as security goes, the private peering domain is simply an extension of your on-premises network into Azure. You then enable bidirectional connectivity between that network and any Azure virtual networks, making the Azure VM IP addresses visible within your internal network.

> ⓘ **Note**
>
> You can connect only one virtual network to the private peering domain.

Azure public peering

Azure public peering enables private connections to services that are available on public IP addresses, such as Azure Storage, Azure SQL databases, and Azure web services. With public peering, you can connect to those service public IP addresses without your traffic being routed over the Internet. Connectivity is always from your WAN to Azure, not the other way around. This is also an all-or-nothing approach, as you can't select the services for which you want public peering enabled.

> ⓘ **Note**

> For Azure PaaS services, it's recommended to use Microsoft peering rather than public peering.

Microsoft peering

Microsoft peering supports connections to cloud-based SaaS offerings, such as Office 365 and Dynamics 365. This peering option provides bi-directional connectivity between your company's WAN and Microsoft cloud services.

ExpressRoute health

As with most features in Microsoft Azure, you can monitor ExpressRoute connections to ensure that they are performing satisfactorily. Monitoring includes coverage of the following areas:

- Availability
- Connectivity to virtual networks
- Bandwidth utilization

The key tool for this monitoring activity is Network Performance Monitor, particularly NPM for ExpressRoute.

Azure ExpressRoute is used to create private connections between Azure datacenters and infrastructure on your premises or in a colocation environment. ExpressRoute connections don't go over the public Internet, and they offer more reliability, faster speeds, and lower latencies than typical Internet connections.

Next unit: Knowledge check

Continue >

✓ 200 XP ▶

Knowledge check

3 minutes

Check your knowledge

1. Which of the following protocols provides dynamic routing for Azure ExpressRoute?

○ IPVPN

○ PPTP

○ IPsec

● Border Gateway Protocol (BGP) ✓

Border Gateway Protocol is an industry-standard dynamic routing protocol that can exchange routes between your on-premises network, your instances in Azure, and Microsoft public addresses.

○ S-key

2. True or false: Azure public peering allows you to connect to services with public IP addresses without your traffic being routed over the internet

● True ✓

Azure public peering enables private connections to services that are available on public IP addresses. Some examples of services that support this are: Azure Storage, and Azure SQL databases.

○ False

Next unit: Summary

Continue >

✓ 100 XP ▶

Summary

3 minutes

Azure offers three primary ways to set up virtual networking:

- Azure virtual networks
- Azure VPN gateways
- Azure ExpressRoute

Azure virtual networks can connect resources such as virtual machines and virtual machine scale sets within the same region, enabling them to communicate. Azure virtual networks can also connect to specified Azure service endpoints, such as Azure Storage, databases, and web apps.

Azure VPN gateways can enable communication with on-premises clients or networks over the public Internet, or connect virtual networks in different Azure regions. When you need a highly secure, dedicated route, you can use Azure ExpressRoute. It creates private, high-bandwidth connections to Azure datacenters that achieve the highest levels of reliability and security.

Cleanup

The interactive exercises in this module created two resource groups, `VpnGatewayDemo` and `vm-networks`. Delete these resources groups.

1. Run the following command to delete the resource group vm-networks and the resources it contains.

   PowerShell Copy

   ```powershell
   Remove-AzResourceGroup -Name vm-networks
   ```

2. Run the following command to delete the resource group VpnGatewayDemo and the resources it contains.

   PowerShell

   ```powershell
   Remove-AzResourceGroup -Name VpnGatewayDemo
   ```

It may take several minutes for the resource groups and the resources they contain to be deleted.

Module complete:

Unlock achievement