

Manage and control traffic flow in your Azure deployment with routes

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/control-network-traffic-flow-with-routes/1-introduction>

Introduction

Completed

A virtual network lets you implement a security perimeter around your resources in the cloud. You can control the information that flows in and out of a virtual network. You can also restrict access to allow only the traffic that originates from trusted sources.

Suppose that you're the solution architect for a retail organization. Also suppose that your organization recently suffered a security incident that exposed customer information such as names, addresses, and credit card numbers. Malicious actors infiltrated vulnerabilities in your retailer's network infrastructure, which resulted in the loss of customers' confidential information.

As part of a remediation plan, the security team recommends adding network protections in the form of network virtual appliances. The cloud infrastructure team must ensure traffic gets properly routed through the virtual appliances and gets inspected for malicious activity.

In this module, you'll learn about Azure routing, and you'll create custom routes to control the traffic flow. You'll also learn to redirect the traffic through the network virtual appliance so you can inspect the traffic before it's allowed through.

Learning objectives

In this module, you'll:

- Identify the routing capabilities of an Azure virtual network.
- Configure routing within a virtual network.
- Deploy a basic network virtual appliance.
- Configure routing to send traffic through a network virtual appliance.

Prerequisites

- Knowledge of basic networking concepts, including subnets and IP addressing
- Familiarity with Azure virtual networking

2. Identify routing capabilities of an Azure virtual network

<https://learn.microsoft.com/en-us/training/modules/control-network-traffic-flow-with-routes/2-azure-virtual-network-route>

Identify routing capabilities of an Azure virtual network

Completed

3. Exercise - Create custom routes

<https://learn.microsoft.com/en-us/training/modules/control-network-traffic-flow-with-routes/3-exercise-create-custom-routes>

Exercise - Create custom routes

Completed

As you implement your security strategy, you want to control how network traffic is routed across your Azure infrastructure.

In the following exercise, you use a network virtual appliance (NVA) to help secure and monitor traffic. You want to ensure communication between front-end public servers and internal private servers is always routed through the appliance.

You configure the network so that all traffic flowing from a public subnet to a private subnet will be routed through the NVA. To make this flow happen, you create a custom route for the public subnet to route this traffic to a perimeter-network subnet. Later, you deploy an NVA to the perimeter-network subnet.



Internet

Routing table:

Address prefix	Next hop address	Next hop type
10.0.1.0/24	10.0.2.4	VirtualAppliance

Subnet

Name: **publicsubnet**
Subnet prefix: 10.0.0.0/24

Subnet

Name: **privatesubnet**
Subnet prefix: 10.0.1.0/24

Subnet

Name: **dmzsubnet**
Subnet prefix: 10.0.2.0/24



Virtual network

Name: **vnet**
Address prefix: 10.0.0.0/16

In this exercise, you create the route table, custom route, and subnets. You'll then associate the route table with a subnet.

Note

This exercise is optional. If you want to complete this exercise, you'll need to create an Azure subscription before you begin. If you don't have an Azure account or you don't want to create one at this time, you can read through the instructions so you understand the information that's being presented.

Note

You need to use a resource group to complete the steps in this exercise. You can use a resource group that you already created, or you can create a new resource group specifically for this exercise. If you choose to create a new resource group, that will make it easier to clean up any resources that you create as you complete the exercise. If you don't have an existing resource group or you want to create a new one specifically for this exercise, you can follow the steps in [Use the Azure portal and Azure Resource Manager to manage resource groups](#) to create a resource group by using the Azure portal, or you can follow the steps in [Manage Azure resource groups by using Azure CLI](#) to create a resource group by using the the Azure CLI.

Note

Throughout this exercise, replace **myResourceGroupName** in the examples with the name of an existing resource group, or the name of the resource group that you created for this exercise.

Create a route table and custom route

The first task is to create a new routing table and then add a custom route for all traffic intended for the private subnet.

Note

You might get an error that reads: *This command is implicitly deprecated*. Please ignore this error for this learning module. We're working on it!

1. Open the [Azure Cloud Shell](#), select **Settings**, then select **Go to Classic version**.
2. In Azure Cloud Shell, run the following command to create a route table. Replace **myResourceGroupName** with the name of your resource group.

```
az network route-table create \  
  --name publictable \  
  --resource-group "myResourceGroupName" \  
  --disable-bgp-route-propagation false
```

3. Run the following command in Cloud Shell to create a custom route:

```
az network route-table route create \  
  --route-table-name publictable \  
  --resource-group "myResourceGroupName" \  
  --name productionsubnet \  
  --address-prefix 10.0.1.0/24 \  
  --next-hop-type VirtualAppliance \  
  --next-hop-ip-address 10.0.2.4
```

Create a virtual network and subnets

The next task is to create the **vnet** virtual network and the three subnets you need: **publicsubnet**, **privatesubnet**, and **dmzsubnet**.

1. Run the following command to create the **vnet** virtual network and the **publicsubnet** subnet:

```
az network vnet create \  
  --name vnet \  
  --resource-group "myResourceGroupName" \  
  --address-prefixes 10.0.0.0/16 \  
  --subnet-name publicsubnet
```

```
--subnet-name publicsubnet \  
--subnet-prefixes 10.0.0.0/24
```

2. Run the following command in Cloud Shell to create the **privatesubnet** subnet:

```
az network vnet subnet create \  
  --name privatesubnet \  
  --vnet-name vnet \  
  --resource-group "myResourceGroupName" \  
  --address-prefixes 10.0.1.0/24
```

3. Run the following command to create the **dmzsubnet** subnet:

```
az network vnet subnet create \  
  --name dmzsubnet \  
  --vnet-name vnet \  
  --resource-group "myResourceGroupName" \  
  --address-prefixes 10.0.2.0/24
```

4. You should now have three subnets. Run the following command to show all of the subnets in the **vnet** virtual network:

```
az network vnet subnet list \  
  --resource-group "myResourceGroupName" \  
  --vnet-name vnet \  
  --output table
```

Associate the route table with the public subnet

The final task in this exercise is to associate the route table with the **publicsubnet** subnet.

Run the following command to associate the route table with the public subnet.

```
az network vnet subnet update \  
  --name publicsubnet \  
  --vnet-name vnet \  
  --resource-group "myResourceGroupName" \  
  --route-table publictable
```

4. What is an NVA?

<https://learn.microsoft.com/en-us/training/modules/control-network-traffic-flow-with-routes/4-network-virtual-appliances>

What is an NVA?

Completed

5. Exercise - Create an NVA and virtual machines

<https://learn.microsoft.com/en-us/training/modules/control-network-traffic-flow-with-routes/5-exercise-create-nva-vm>

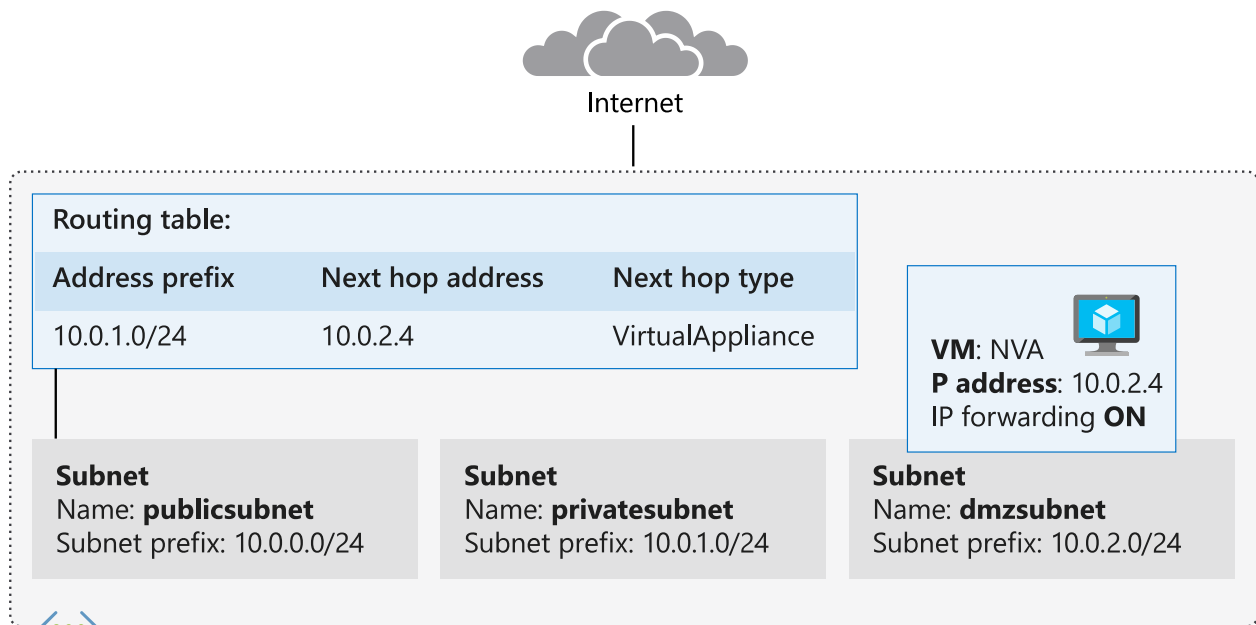
Exercise - Create an NVA and virtual machines

Completed

In the next stage of your security implementation, you'll deploy a network virtual appliance (NVA) to secure and monitor traffic between your front-end public servers and internal private servers.

First, configure the appliance to forward IP traffic. If IP forwarding isn't enabled, traffic that is routed through your appliance will never be received by its intended destination servers.

In this exercise, you deploy the **nva** network appliance to the **dmzsubnet** subnet. Then you enable IP forwarding so that traffic from ***** and traffic that uses the custom route is sent to the **privatesubnet** subnet.



Virtual network

Name: **vnet**

Address prefix: 10.0.0.0/16

In the following steps, you'll deploy an NVA. You'll then update the Azure virtual NIC and the network settings within the appliance to enable IP forwarding.

Note

This exercise is optional. If you want to complete this exercise, you'll need to create an Azure subscription before you begin. If you don't have an Azure account or you don't want to create one at this time, you can read through the instructions so you understand the information that's being presented.

Deploy the network virtual appliance

To build the NVA, deploy an Ubuntu LTS instance.

1. In [Azure Cloud Shell](#), run the following command to deploy the appliance. Replace **<password>** with a suitable password of your choice for the **azureuser** admin account, and **myResourceGroupName** with the name of your resource group.

```
az vm create \
  --resource-group "myResourceGroupName" \
  --name nva \
  --vnet-name vnet \
  --subnet dmzsubnet \
  --image Ubuntu2204 \
  --admin-username azureuser \
  --admin-password <password>
```

Enable IP forwarding for the Azure network interface

In the next steps, you enable IP forwarding for the **nva** network appliance. When traffic flows to the NVA but is meant for another target, the NVA will route that traffic to its correct destination.

1. Run the following command to get the NVA network interface's ID:

```
NICID=$(az vm nic list \
  --resource-group "myResourceGroupName" \
  --vm-name nva \
  --query "[].{id:id}" --output tsv)

echo $NICID
```

2. Run the following command to get the NVA network interface's name:

```
NICNAME=$(az vm nic show \
  --resource-group "myResourceGroupName" \
  --vm-name nva \
  --nic $NICID \
  --query "{name:name}" --output tsv)

echo $NICNAME
```

3. Run the following command to enable IP forwarding for the network interface:

```
az network nic update --name $NICNAME \
  --resource-group "myResourceGroupName" \
  --ip-forwarding true
```

Enable IP forwarding in the appliance

1. Run the following command to save the NVA virtual machine's public IP address to the variable `NVAIP`:

```
NVAIP="$(az vm list-ip-addresses \
  --resource-group "myResourceGroupName" \
  --name nva \
  --query "[].virtualMachine.network.publicIpAddresses[*].ipAddress" \
  --output tsv)"

echo $NVAIP
```

2. Run the following command to enable IP forwarding within the NVA:

```
ssh -t -o StrictHostKeyChecking=no azureuser@$NVAIP 'sudo sysctl -w net.ipv4.
```


When prompted, enter the password you used when you created the virtual machine.

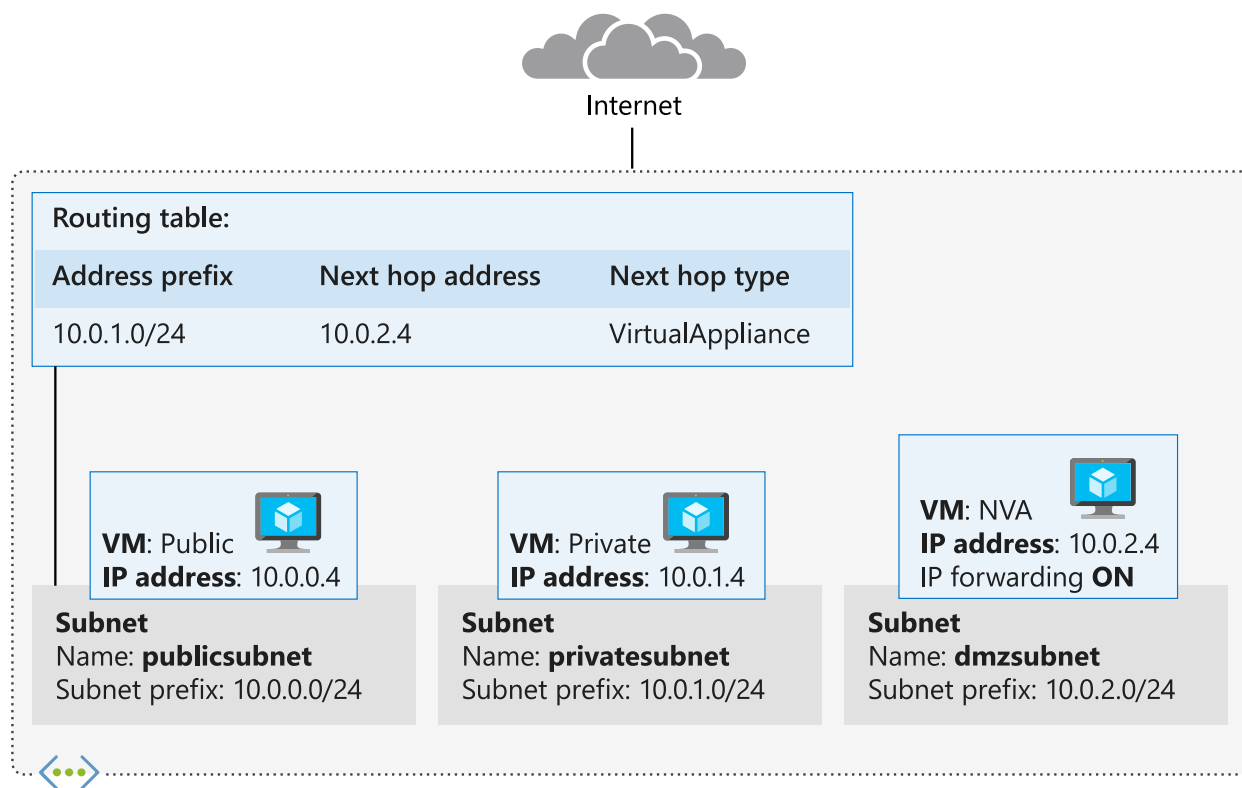
6. Exercise - Route traffic through the NVA

<https://learn.microsoft.com/en-us/training/modules/control-network-traffic-flow-with-routes/6-exercise-route-traffic-through-nva>

Exercise - Route traffic through the NVA

Completed

Now that you've created the network virtual appliance (NVA) and virtual machines (VMs), you'll route the traffic through the NVA.



Virtual network

Name: **vnet**

Address prefix: 10.0.0.0/16

Note

This exercise is optional. If you want to complete this exercise, you'll need to create an Azure subscription before you begin. If you don't have an Azure account or you don't want to create

one at this time, you can read through the instructions so you understand the information that's being presented.

Create public and private virtual machines

The next steps deploy a VM into the public and private subnets.

1. Open the [Azure Cloud Shell](#) editor and create a file named **cloud-init.txt**.

```
code cloud-init.txt
```

2. Add the following configuration information to the file. With this configuration, the `inetutils-traceroute` package is installed when you create a new VM. This package contains the `traceroute` utility that you'll use later in this exercise.

```
#cloud-config
package_upgrade: true
packages:
  - inetutils-traceroute
```

3. Press Ctrl+S to save the file, and then press Ctrl+Q to close the editor.
4. In Cloud Shell, run the following command to create the **public** VM. Replace `<password>` with a suitable password for the **azureuser** account.

```
az vm create \
  --resource-group "myResourceGroupName" \
  --name public \
  --vnet-name vnet \
  --subnet publicsubnet \
  --image Ubuntu2204 \
  --admin-username azureuser \
  --no-wait \
  --custom-data cloud-init.txt \
  --admin-password <password>
```

5. Run the following command to create the **private** VM. Replace `<password>` with a suitable password.

```
az vm create \
  --resource-group "myResourceGroupName" \
  --name private \
  --vnet-name vnet \
  --subnet privatesubnet \
  --image Ubuntu2204 \
  --admin-username azureuser \
  --no-wait \
```

```
--custom-data cloud-init.txt \  
--admin-password <password>
```

6. Run the following Linux `watch` command to check that the VMs are running. The `watch` command periodically runs the `az vm list` command so that you can monitor the progress of the VMs.

```
watch -d -n 5 "az vm list \  
  --resource-group "myResourceGroupName" \  
  --show-details \  
  --query '[*].{Name:name, ProvisioningState:provisioningState, PowerState:  
  --output table"
```

A **ProvisioningState** value of "Succeeded" and a **PowerState** value of "VM running" indicate a successful deployment. When all three VMs are running, you're ready to move on. Press Ctrl-C to stop the command and continue with the exercise.

7. Run the following command to save the public IP address of the **public** VM to a variable named `PUBLICIP`:

```
PUBLICIP="$(az vm list-ip-addresses \  
  --resource-group "myResourceGroupName" \  
  --name public \  
  --query "[]\.virtualMachine.network.publicIpAddresses[*].ipAddress" \  
  --output tsv)"  
  
echo $PUBLICIP
```

8. Run the following command to save the public IP address of the **private** VM to a variable named `PRIVATEIP`:

```
PRIVATEIP="$(az vm list-ip-addresses \  
  --resource-group "myResourceGroupName" \  
  --name private \  
  --query "[]\.virtualMachine.network.publicIpAddresses[*].ipAddress" \  
  --output tsv)"  
  
echo $PRIVATEIP
```

Test traffic routing through the network virtual appliance

The final steps use the Linux `traceroute` utility to show how traffic is routed. You'll use the `ssh` command to run `traceroute` on each VM. The first test shows the route taken by ICMP packets sent from the **public** VM to the **private** VM. The second test shows the route taken by ICMP packets sent from the **private** VM to the **public** VM.

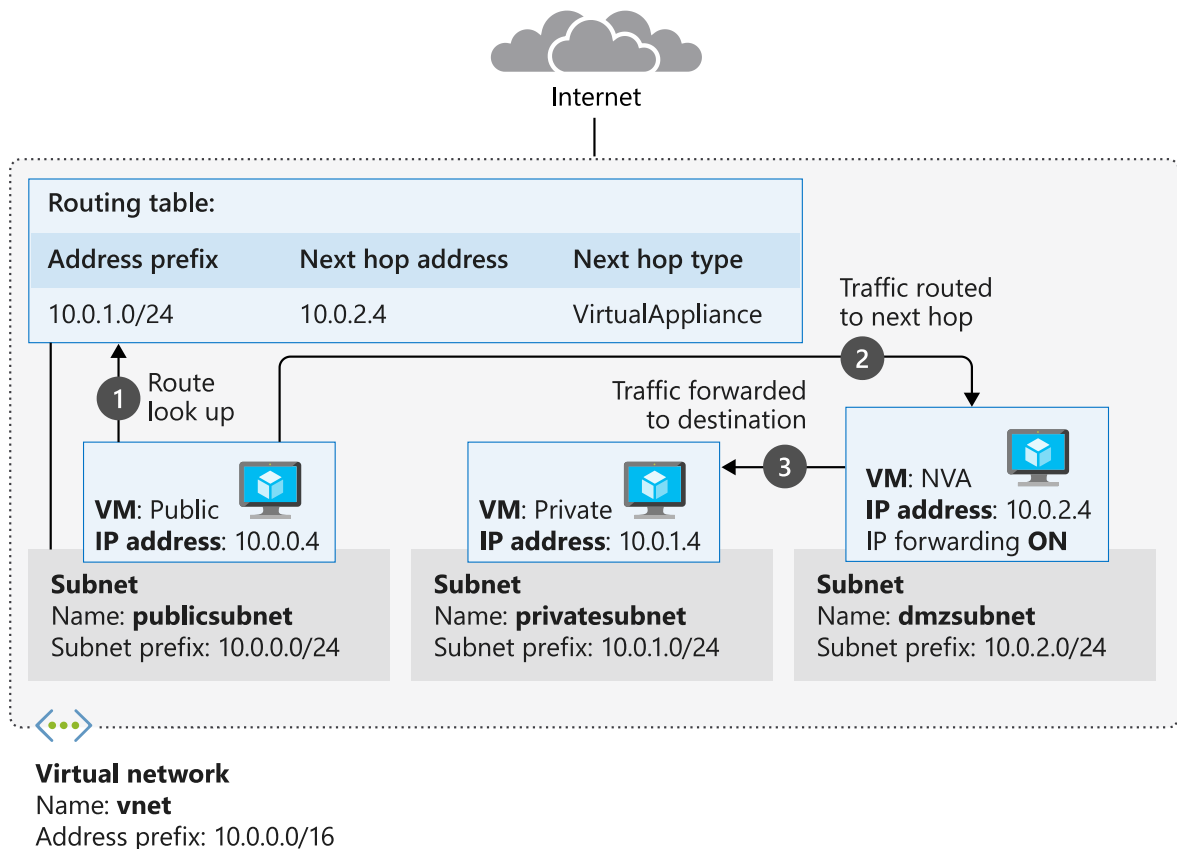
1. Run the following command to trace the route from **public** to **private**. When prompted, enter the password for the **azureuser** account that you specified earlier.

```
ssh -t -o StrictHostKeyChecking=no azureuser@$PUBLICIP 'traceroute private --
```

If you receive the error message `bash: traceroute: command not found`, wait a minute and retry the command. The automated installation of `traceroute` can take a minute or two after VM deployment. After the command succeeds, the output should look similar to the following example:

```
traceroute to private.kzffavtrkpeulburui2lgyxwg.gx.internal.cloudapp.net (10
1  10.0.2.4  0.710ms  0.410ms  0.536ms
2  10.0.1.4  0.966ms  0.981ms  1.268ms
Connection to 52.165.151.216 closed.
```

Notice that the first hop is to 10.0.2.4. This address is the private IP address of **nva**. The second hop is to 10.0.1.4, the address of **private**. In the first exercise, you added this route to the route table and linked the table to the **publicsubnet** subnet. So now all traffic from **public** to **private** is routed through the NVA.



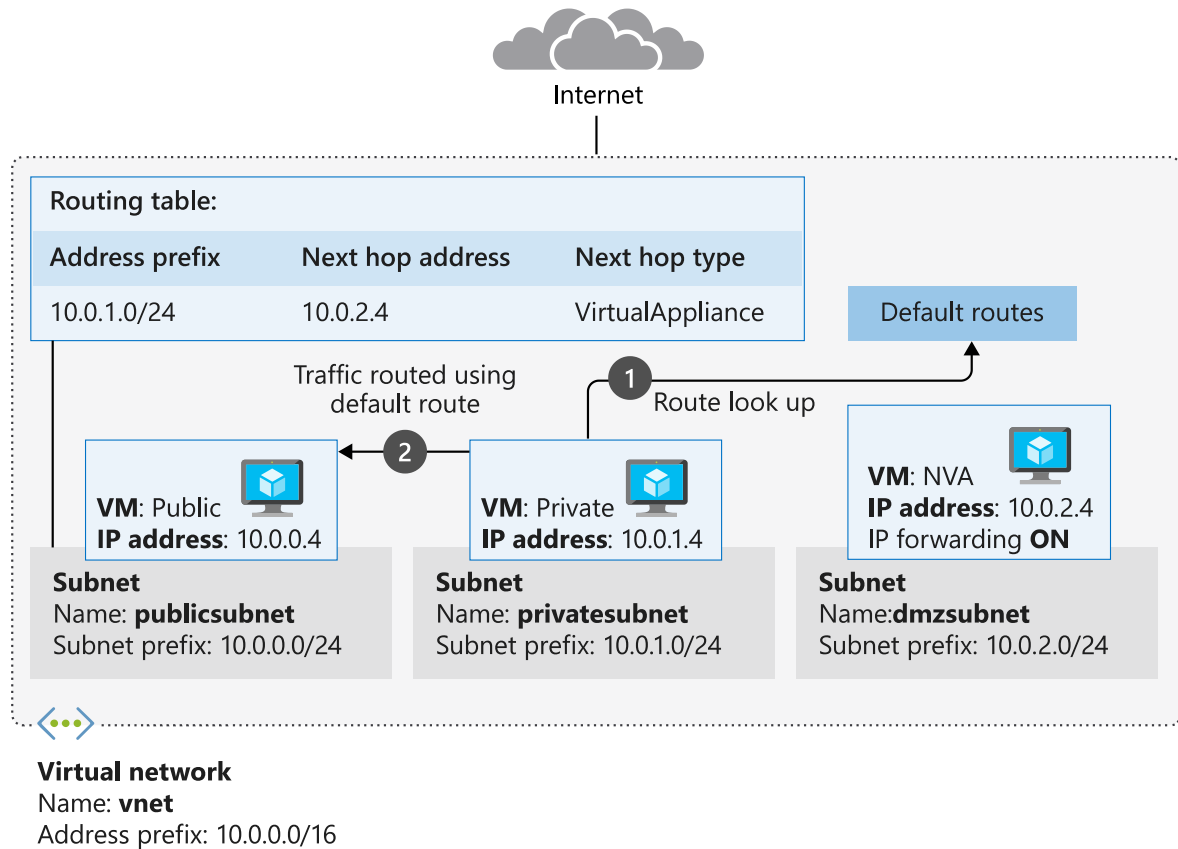
2. Run the following command to trace the route from **private** to **public**. When prompted, enter the password for the **azureuser** account.

```
ssh -t -o StrictHostKeyChecking=no azureuser@$PRIVATEIP 'traceroute public --
```

You should see the traffic go directly to **public** (10.0.0.4) and not through the NVA, as shown in the following command output.

```
tracert to public.kzffavtrkpeulburui2lgywxwg.gx.internal.cloudapp.net (10.1.0.4)
1  10.0.0.4  1.095ms  1.610ms  0.812ms
Connection to 52.173.21.188 closed.
```

The **private** VM is using default routes, and traffic is routed directly between the subnets.



You've now configured routing between subnets to direct traffic from the public internet through the **dmzsubnet** subnet before it reaches the private subnet. In the **dmzsubnet** subnet, you added a VM that acts as an NVA. You can configure this NVA to detect potentially malicious requests and block them before they reach their intended targets.

7. Summary

<https://learn.microsoft.com/en-us/training/modules/control-network-traffic-flow-with-routes/7-summary>

Summary

Completed

In this module, you learned how to customize routes in an Azure virtual network and how to redirect the traffic flow through a network virtual appliance. You also learned how to create your own custom network virtual appliance by deploying an Azure virtual machine.

Important

In the optional exercises for this module, you created resources by using your own Azure subscription. Clean up these resources so that you won't continue to be charged for them.

Learn more

For more information on using routes in your network infrastructure, see the following articles:

- [Virtual network traffic routing](#)
- [Tutorial: Route network traffic with a route table using the Azure portal](#)
- [Deploy highly available network virtual appliances](#)
- [Implement a secure hybrid network](#)