

In this lab, you will learn how to create an Azure VNet route table using the Azure CLI.

## Learning Objectives

In this lab, you will learn how to do the following:

Log into the Azure CLI

Understand Azure VNet route tables

Configure VNet route tables using Azure CLI

Confirm VNet route tables using Azure CLI

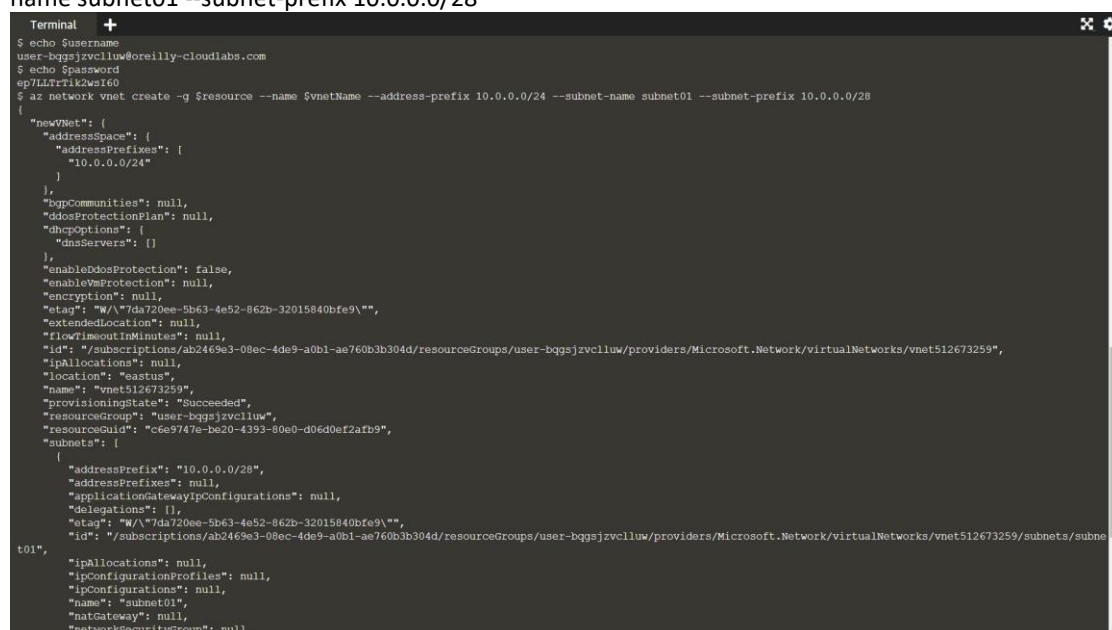
Remove VNet route tables using Azure CLI

Login to azure CLI

Az login -u \$username -p \$password

To create a new Azure Virtual Network and a subnet:

```
az network vnet create --resource-group $resource --name $vnetName --address-prefix 10.0.0.0/24 --subnet-name subnet01 --subnet-prefix 10.0.0.0/28
```



```
Terminal
$ echo $username
user-bqgsjzvc1luw@oreilly-cloudlabs.com
$ echo $password
sp7L7rTik2wsi60
$ az network vnet create -g $resource --name $vnetName --address-prefix 10.0.0.0/24 --subnet-name subnet01 --subnet-prefix 10.0.0.0/28
{
  "newVNet": {
    "addressSpace": {
      "addressPrefixes": [
        "10.0.0.0/24"
      ]
    },
    "bgpCommunities": null,
    "ddosProtectionPlan": null,
    "dhcpOptions": {
      "dnsServers": []
    },
    "enableDdosProtection": false,
    "enableVmProtection": null,
    "encryption": null,
    "etag": "W/7da720ee-5b63-4e52-862b-32015840bfe9",
    "extendedLocation": null,
    "flowTimeoutInMinutes": null,
    "id": "/subscriptions/ab2469e3-08ec-4de9-a0b1-ae760b3b304d/resourceGroups/user-bqgsjzvc1luw/providers/Microsoft.Network/virtualNetworks/vnet512673259",
    "ipAllocations": null,
    "location": "eastus",
    "name": "vnet512673259",
    "provisioningState": "Succeeded",
    "resourceGroup": "user-bqgsjzvc1luw",
    "resourceGuid": "c6e9747e-be20-4393-80e0-d06d0ef2afb9",
    "subnets": [
      {
        "addressPrefix": "10.0.0.0/28",
        "addressPrefixes": null,
        "applicationGatewayIpConfigurations": null,
        "delegations": [],
        "etag": "W/7da720ee-5b63-4e52-862b-32015840bfe9",
        "id": "/subscriptions/ab2469e3-08ec-4de9-a0b1-ae760b3b304d/resourceGroups/user-bqgsjzvc1luw/providers/Microsoft.Network/virtualNetworks/vnet512673259/subnets/subnet01",
        "ipAllocations": null,
        "ipConfigurationProfiles": null,
        "ipConfigurations": null,
        "name": "subnet01",
        "natGateway": null,
        "networkSecurityGroup": null,

```

## Understand Azure Virtual Network Route Tables

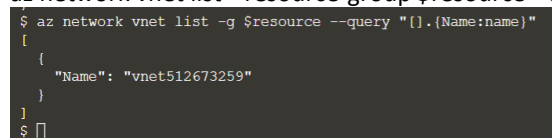
Azure directs Azure VNet subnet traffic to destinations (next hop) using route tables. Azure automatically creates a default route table for each VNet subnet and adds system default routes to the table. These default routes specify where the subnet network traffic should be routed to.

Using route tables, you can override these custom routes by creating user-defined routes. This is helpful in many scenarios; for example, when you need to make sure that all inbound/outbound subnet traffic is directed to a network appliance such as an Azure Firewall instance.

User-defined routes take priority over the default system routes. If no user-defined route is found for the traffic destination, the system routes will be used.

Use the following command to confirm that the Azure VNet is present:

```
az network vnet list --resource-group $resource --query "[].{Name:name}"
```



```
$ az network vnet list -g $resource --query "[].{Name:name}"
[
  {
    "Name": "vnet512673259"
  }
]
$
```

Now, in the next step, let's create a route table that directs all the subnet traffic to the internet.

Azure Virtual Network Route Tables (VRTs) are a mechanism for defining custom routing rules for network traffic within a virtual network (VNet) in Azure. A VRT is a container for a set of route rules that define how network traffic should be forwarded within the VNet. When a VNet is created in Azure, a default route table is also created. However, you can also create custom route tables that can be associated with subnets within the VNet.

When a VRT is associated with a subnet, the route rules defined in the VRT are used to determine how network traffic is routed within the subnet. The route table determines the next hop for each packet based on the destination IP address of the packet and the rules defined in the table. This allows you to define specific routes for network traffic within the VNet, which can be useful in a variety of scenarios, such as:

Routing traffic to a specific network appliance or virtual machine that provides services such as firewalling, load balancing, or VPN connectivity.

Creating custom routing paths for traffic between different subnets within the VNet.

Routing traffic to on-premises networks or other Azure virtual networks.

You can create and manage VRTs using the Azure portal, Azure CLI, Azure PowerShell, or Azure Resource Manager templates. When creating or modifying a VRT, you define a set of route rules that include the destination address prefix, the next hop type (e.g., virtual network gateway, internet, or virtual appliance), and the next hop address. Once the route table is created and associated with a subnet, it is automatically applied to all virtual machines and other resources in that subnet.

Configure Azure Virtual Network Route Tables using Azure CLI

Before creating your custom routes, you need to create an Azure route table resource using this command:

```
az network route-table create --resource-group $resource --name $routeTableName
```

Now let's create a single route in this route table:

```
az network route-table route create --resource-group $resource --route-table-name $routeTableName --name sample_internet_route --next-hop-type Internet --address-prefix 0.0.0.0/0
```

Here are the parameters for this command:

--resource-group: Name of the parent route table resource group.

--route-table-name: Name of the route table in which to create the route.

--name: The new route name.

--next-hop-type: The type of Azure hop the packet should be sent to (accepted values are Internet, None, VirtualAppliance, VirtualNetworkGateway, and VnetLocal).

--address-prefix: The destination CIDR to which the route applies. In our case, we like to use the route for all traffic, so 0.0.0.0/0 is used.

--next-hop-ip-address: The IP address packets should be forwarded to this address. when using the VirtualAppliance hop type. This is not used in our example because we are not using a virtual appliance.

Finally, use the following command to assign this route to your subnet:

```
az network vnet subnet update --resource-group $resource --name subnet01 --vnet-name $vnetName --route-table $routeTableName
```

```

Terminal +
$ az network route-table create -g $resource --name $routeTableName
{
  "disableBgpRoutePropagation": false,
  "etag": "W/\"9da3c8fb-a544-49fc-b2b1-4c564ba4f2d1\"",
  "id": "/subscriptions/ab2469e3-08ec-4de9-a0b1-ae760b3b304d/resourceGroups/user-bqgsjzvc1luw/providers/Microsoft.Network/routeTables/route512673259",
  "location": "eastus",
  "name": "route512673259",
  "provisioningState": "Succeeded",
  "resourceGroup": "user-bqgsjzvc1luw",
  "resourceGuid": "73889904-3859-48d2-bbf4-15bf4754e649",
  "routes": [],
  "type": "Microsoft.Network/routeTables"
}
$ az network route-table route create -g $resource --route-table-name $routeTableName --name sample_internet_route --next-hop-type Internet --address-prefix 0.0.0.0/0
{
  "addressPrefix": "0.0.0.0/0",
  "etag": "W/\"627cf26d-2b82-448e-bf7d-18f5f22cabf4\"",
  "hasBgpOverride": false,
  "id": "/subscriptions/ab2469e3-08ec-4de9-a0b1-ae760b3b304d/resourceGroups/user-bqgsjzvc1luw/providers/Microsoft.Network/routeTables/route512673259/routes/sample_internet_route",
  "name": "sample_internet_route",
  "nextHopType": "Internet",
  "provisioningState": "Succeeded",
  "resourceGroup": "user-bqgsjzvc1luw",
  "type": "Microsoft.Network/routeTables/routes"
}
$ az network vnet subnet update -g $resource --name subnet01 --vnet-name $vnetName --route-table $routeTableName
{
  "addressPrefix": "10.0.0.0/28",
  "addressPrefixes": null,
  "applicationGatewayIpConfigurations": null,
  "delegations": [],
  "etag": "W/\"b5f61e19-9e60-4a07-8c42-f5df09ebabd2\"",
  "id": "/subscriptions/ab2469e3-08ec-4de9-a0b1-ae760b3b304d/resourceGroups/user-bqgsjzvc1luw/providers/Microsoft.Network/virtualNetworks/vnet512673259/subnets/subnet01",
  "ipAllocations": null,
  "ipConfigurationProfiles": null,
  "ipConfigurations": null,
  "name": "subnet01",
  "natGateway": null,
  "networkSecurityGroup": null,
  "privateEndpointNetworkPolicies": "Disabled",
  "privateEndpoints": null,
  "privateLinkServiceNetworkPolicies": "Enabled",
  "provisioningState": "Succeeded",
}

```

In the next step, we will use Azure CLI to confirm that the route exists for this subnet.

Important: The preceding command directs all the subnet traffic with any destination to the internet. We used this route as an example for simplicity, and also because we don't have a virtual appliance in the temporary platform. One of the most common use cases for custom routes is to direct all subnet traffic to a virtual appliance, such as an Azure Firewall. In that case, the following command could be used:

```
az network route-table route create --resource-group $resource --route-table-name $routeTableName -name firewall_route --next-hop-type VirtualAppliance --address-prefix 0.0.0.0/0 --next-hop-ip-address 10.0.100.4
```

In the preceding command, 10.0.100.4 is the IP address of an imaginary Azure Firewall instance.

List VNet Route Tables using Azure CLI

Use the following command to get the subnet details:

```
az network vnet subnet show --resource-group $resource --name subnet01 --vnet-name $vnetName --query "{Name:name, RouteTable: routeTable.id}"
```

```

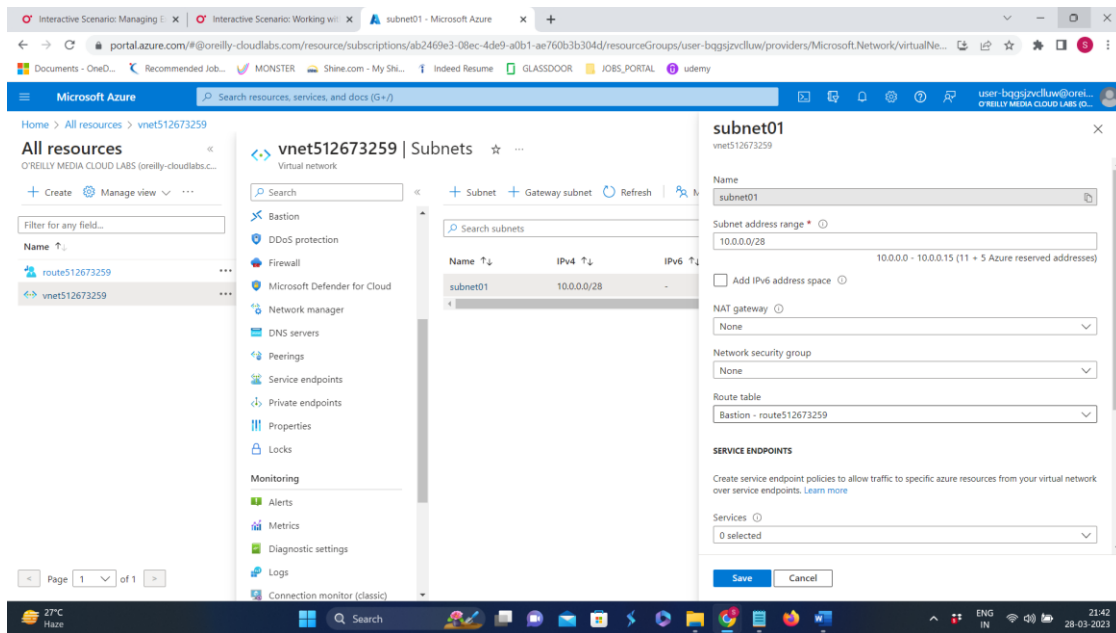
$ az network route-table route create -g $resource --route-table-name $routeTableName --name sample_internet_route --next-hop-type Internet --address-prefix 0.0.0.0/0
$ az network vnet subnet show -g $resource --name subnet01 --vnet-name $vnetName --query "{Name:name, RouteTable: routeTable.id}"
{
  "Name": "subnet01",
  "RouteTable": "/subscriptions/ab2469e3-08ec-4de9-a0b1-ae760b3b304d/resourceGroups/user-bqgsjzvc1luw/providers/Microsoft.Network/routeTables/route512673259"
}
$

```

Check the VNet Route Tables in the Azure Portal

Click on subnet01.

Confirm that the Route table dropdown has your net route table name.



## Delete Azure Virtual Network Route Tables

Use the following command to remove the route table from your subnet:

```
az network vnet subnet update --resource-group $resource --name subnet01 --vnet-name $vnetName --route-table ""
```

```
{
  "addressPrefix": "10.0.0.0/28",
  "addressPrefixes": null,
  "applicationGatewayIpConfigurations": null,
  "delegations": [],
  "etag": "W/\"7b976aab-85b6-40e7-be3f-7ed10acb5fbd\"",
  "id": "/subscriptions/ab2469e3-09ec-4de9-a0b1-ae760b3b304d/resourceGroups/user-bqgsjzvc1lw/providers/Microsoft.Network/virtualNetworks/vnet512673259/subnets/subnet01",
  "ipAllocations": null,
  "ipConfigurationProfiles": null,
  "ipConfigurations": null,
  "name": "subnet01",
  "natGateway": null,
  "networkSecurityGroup": null,
  "privateEndpointNetworkPolicies": "Disabled",
  "privateEndpoints": null,
  "privateLinkServiceNetworkPolicies": "Enabled",
  "provisioningState": "Succeeded",
  "purpose": null,
  "resourceGroup": "user-bqgsjzvc1lw",
  "resourceNavigationLinks": null,
  "routeTable": null,
  "serviceAssociationLinks": null,
  "serviceEndpointPolicies": null,
  "serviceEndpoints": null,
  "type": "Microsoft.Network/virtualNetworks/subnets"
}
```

Now, use the following command to confirm that there is no longer a route table attached to your subnet (null):

```
az network vnet subnet show --resource-group $resource --name subnet01 --vnet-name $vnetName --query "{Name:name, RouteTable: routeTable.id}"
```

Now you can use the following command to delete the route table itself:

```
az network route-table delete --resource-group $resource --name $routeTableName
```

```
{
  "Name": "subnet01",
  "RouteTable": null
}
$ az network route-table delete --resource-group $resource --name $routeTableName
$
```