

In this lab, you will learn how to update an Azure VNet subnet using the Azure CLI.

Login to azure cli

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

In the next step, we will update this subnet using the Azure CLI.

Understanding Azure Virtual Network Subnet Updating

Azure Virtual Network is the foundation of private networks in Microsoft Azure. Many Azure services support deployment to an Azure Virtual Network. This enables the resources to communicate with each other and other networks in a more secure way.

Several properties of an existing VNet subnet can be updated in the Azure portal, or programmatically using tools such as the Azure CLI. For instance, you can protect your subnet by assigning a network security group (NSG) to it. Here are a few properties which can be updated:

--network-security-group: Name of the NSG to be assigned to the subnet

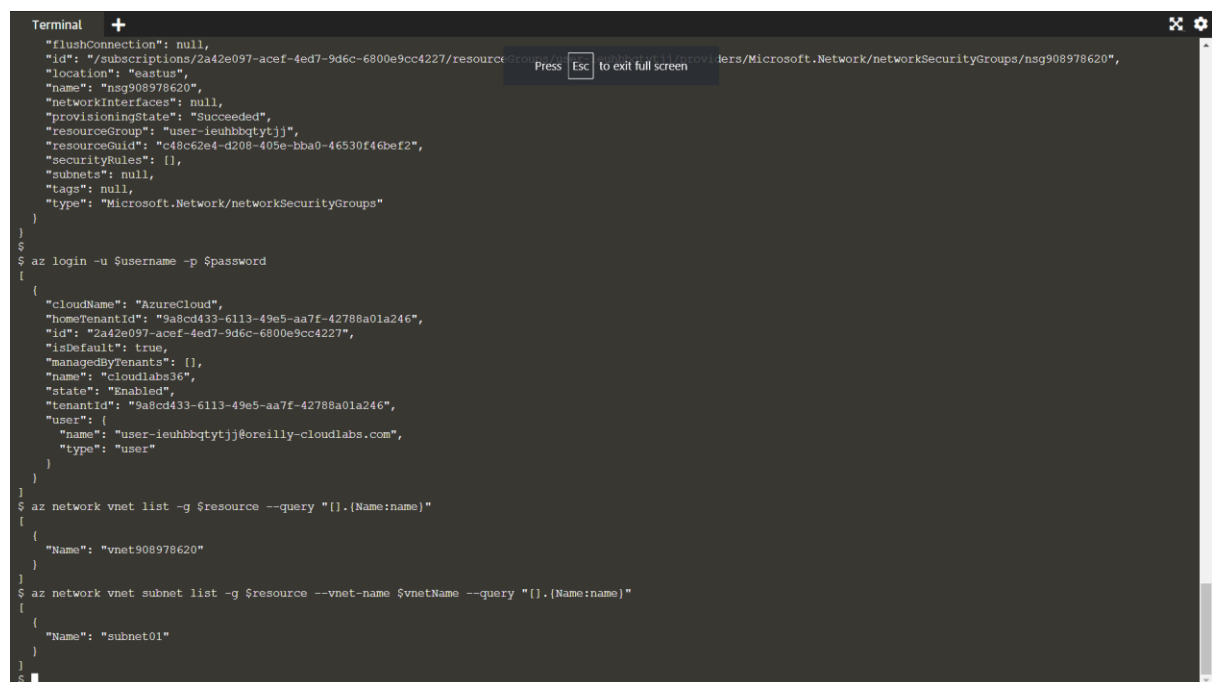
--route-table: The name of the route table resource, to be assigned to the subnet

Before starting, use the following command to confirm that a VNet and its child subnet are successfully created for you:

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

This VNet already contains a subnet, named subnet01. You can confirm this by running the following query:

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

A terminal window with a dark background and light text. It shows the execution of several Azure CLI commands. The first command is 'az login -u \$username -p \$password', which outputs a large JSON object representing the user's profile. The second command is 'az network vnet list -g \$resource --query "[].{Name:name}"', which outputs a JSON array containing the name of the virtual network, 'vnet908978620'. The third command is 'az network vnet subnet list -g \$resource --vnet-name \$vnetName --query "[].{Name:name}"', which outputs a JSON array containing the name of the subnet, 'subnet01'.

```
Terminal +
{"flushConnection": null,
"id": "/subscriptions/2a42e097-acef-4ed7-9d6c-6800e9cc4227/resourceGroups/oreilly-clou...ers/Microsoft.Network/networkSecurityGroups/nsg908978620",
"location": "eastus",
"name": "nsg908978620",
"networkInterfaces": null,
"provisioningState": "Succeeded",
"resourceGroup": "user-ieuhbbqtytjj",
"resourceId": "c48c62e4-d208-409e-bba0-46530f46bef2",
"securityRules": [],
"subnets": null,
"tags": null,
"type": "Microsoft.Network/networkSecurityGroups"
}
$
$ az login -u $username -p $password
{
  "cloudName": "AzureCloud",
  "homeTenantId": "9a8cd433-6113-49e5-aa7f-42788a01a246",
  "id": "2a42e097-acef-4ed7-9d6c-6800e9cc4227",
  "isDefault": true,
  "managedByTenants": [],
  "name": "cloudlabs36",
  "state": "Enabled",
  "tenantId": "9a8cd433-6113-49e5-aa7f-42788a01a246",
  "user": {
    "name": "user-ieuhbbqtytjj@oreilly-cloudlabs.com",
    "type": "user"
  }
}
$ az network vnet list -g $resource --query "[].{Name:name}"
[
  {
    "Name": "vnet908978620"
  }
]
$ az network vnet subnet list -g $resource --vnet-name $vnetName --query "[].{Name:name}"
[
  {
    "Name": "subnet01"
  }
]
$
```

Now let's update subnet01, using the Azure CLI.

Updating an Azure Virtual Network Subnet

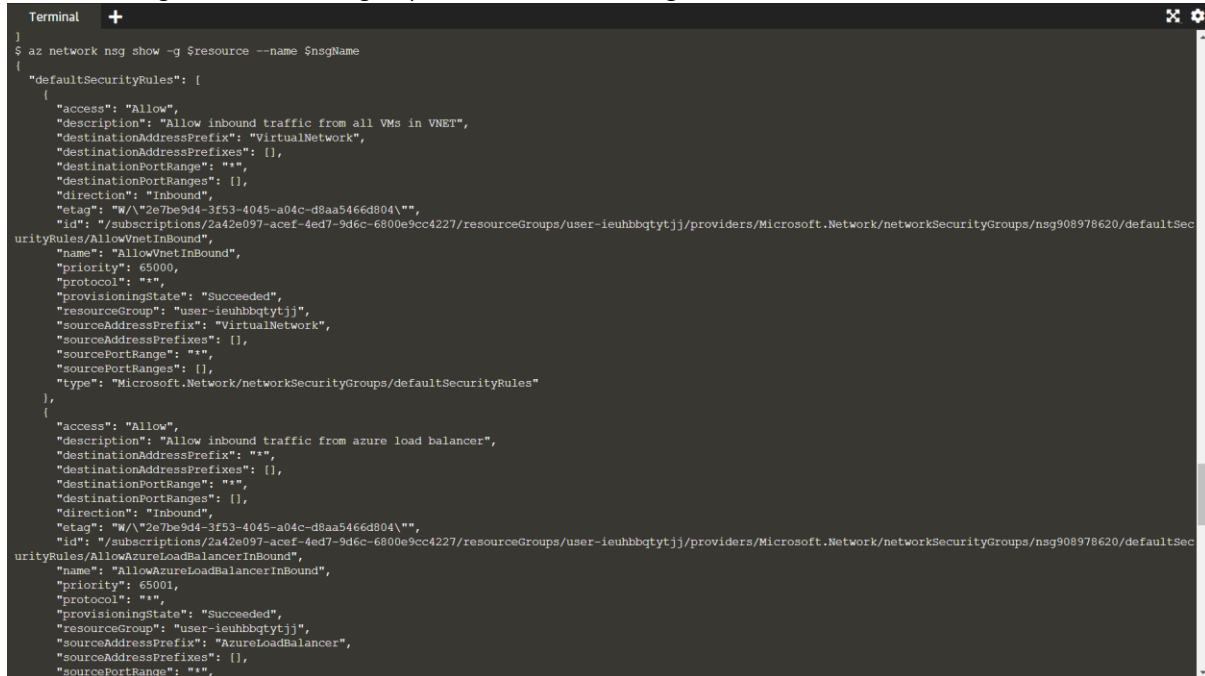
We preconfigured this lab, so a new NSG is already created for you. This NSG blocks all incoming and outgoing traffic to the subnets it's assigned to.

The following command assigns a new NSG to subnet01. NSGs enable you to protect your subnets and resources deployed to them from unwanted traffic. Both incoming and outgoing traffic can be monitored.

Note: An NSG is a standalone resource type. A single NSG can be assigned to multiple subnets, but a subnet can only be assigned to one NSG.

Use the following command to get the details for the existing NSG:

`az network nsg show --resource-group $resource --name $nsgName`



```
Terminal +
$ az network nsg show -g $resource --name $nsgName
{
  "defaultSecurityRules": [
    {
      "access": "Allow",
      "description": "Allow inbound traffic from all VMs in VNET",
      "destinationAddressPrefix": "VirtualNetwork",
      "destinationAddressPrefixes": [],
      "destinationPortRange": "*",
      "destinationPortRanges": [],
      "direction": "Inbound",
      "etag": "W/\"2e7be9d4-3f53-4045-a04c-d8aa5466d804\"",
      "id": "/subscriptions/2a42e097-acef-4ed7-9d6c-6800e9cc4227/resourceGroups/user-ieuhbbqytj/providers/Microsoft.Network/networkSecurityGroups/nsg908978620/defaultSecurityRules/AllowVnetInBound",
      "name": "AllowVnetInBound",
      "priority": 65000,
      "protocol": "*",
      "provisioningState": "Succeeded",
      "resourceGroup": "user-ieuhbbqytj",
      "sourceAddressPrefix": "VirtualNetwork",
      "sourceAddressPrefixes": [],
      "sourcePortRange": "*",
      "sourcePortRanges": [],
      "type": "Microsoft.Network/networkSecurityGroups/defaultSecurityRules"
    },
    {
      "access": "Allow",
      "description": "Allow inbound traffic from azure load balancer",
      "destinationAddressPrefix": "*",
      "destinationAddressPrefixes": [],
      "destinationPortRange": "*",
      "destinationPortRanges": [],
      "direction": "Inbound",
      "etag": "W/\"2e7be9d4-3f53-4045-a04c-d8aa5466d804\"",
      "id": "/subscriptions/2a42e097-acef-4ed7-9d6c-6800e9cc4227/resourceGroups/user-ieuhbbqytj/providers/Microsoft.Network/networkSecurityGroups/nsg908978620/defaultSecurityRules/AllowAzureLoadBalancerInBound",
      "name": "AllowAzureLoadBalancerInBound",
      "priority": 65001,
      "protocol": "*",
      "provisioningState": "Succeeded",
      "resourceGroup": "user-ieuhbbqytj",
      "sourceAddressPrefix": "AzureLoadBalancer",
      "sourceAddressPrefixes": [],
      "sourcePortRange": "*",
      "sourcePortRanges": []
    }
  ]
}
```

Now you can assign the NSG to subnet01 using this command:

`az network vnet subnet update --resource-group $resource --name subnet01 --vnet-name $vnetName --network-security-group $nsgName`

Here are the parameters for this command:

- resource-group: Name of the VNet Resource Group
- name: Name of the existing subnet to update
- network-security-group: NSG name to assign

```
Terminal +
$ az network vnet subnet update -g $resource --name subnet01 --vnet-name $vnetName --network-security-group $nsgName
{
  "addressPrefix": "10.0.0.0/28",
  "addressPrefixes": null,
  "applicationGatewayIpConfigurations": null,
  "delegations": [],
  "etag": "W/\"1356f30a-0566-4910-88d2-62ae0971c203\"",
  "id": "/subscriptions/2a42e097-acef-4ed7-9d6c-6800e9cc4227/resourceGroups/user-ieuhbbqytjy/providers/Microsoft.Network/virtualNetworks/vnet908978620/subnets/subnet01",
  "ipAllocations": null,
  "ipConfigurationProfiles": null,
  "ipConfigurations": null,
  "name": "subnet01",
  "natGateway": null,
  "networkSecurityGroup": {
    "defaultSecurityRules": null,
    "etag": null,
    "flowLogs": null,
    "flushConnection": null,
    "subscriptions/2a42e097-acef-4ed7-9d6c-6800e9cc4227/resourceGroups/user-ieuhbbqytjy/providers/Microsoft.Network/networkSecurityGroups/nsg908978620",
    "location": null,
    "name": null,
    "networkInterfaces": null,
    "provisioningState": null,
    "resourceGroup": "user-ieuhbbqytjy",
    "resourceGuid": null,
    "securityRules": null,
    "subnets": null,
    "tags": null,
    "type": null
  },
  "privateEndpointNetworkPolicies": "Disabled",
  "privateEndpoints": null,
  "privateLinkServiceNetworkPolicies": "Enabled",
  "provisioningState": "Succeeded",
  "purpose": null,
  "resourceGroup": "user-ieuhbbqytjy",
  "resourceNavigationLinks": null,
  "routeTable": null,
  "serviceAssociationLinks": null,
  "serviceEndpointPolicies": null,
  "serviceEndpoints": null,
  "type": "Microsoft.Network/virtualNetworks/subnets"
}
```

Confirm the Subnet is Updated

Use the following command to confirm that the VNet address space is updated:

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

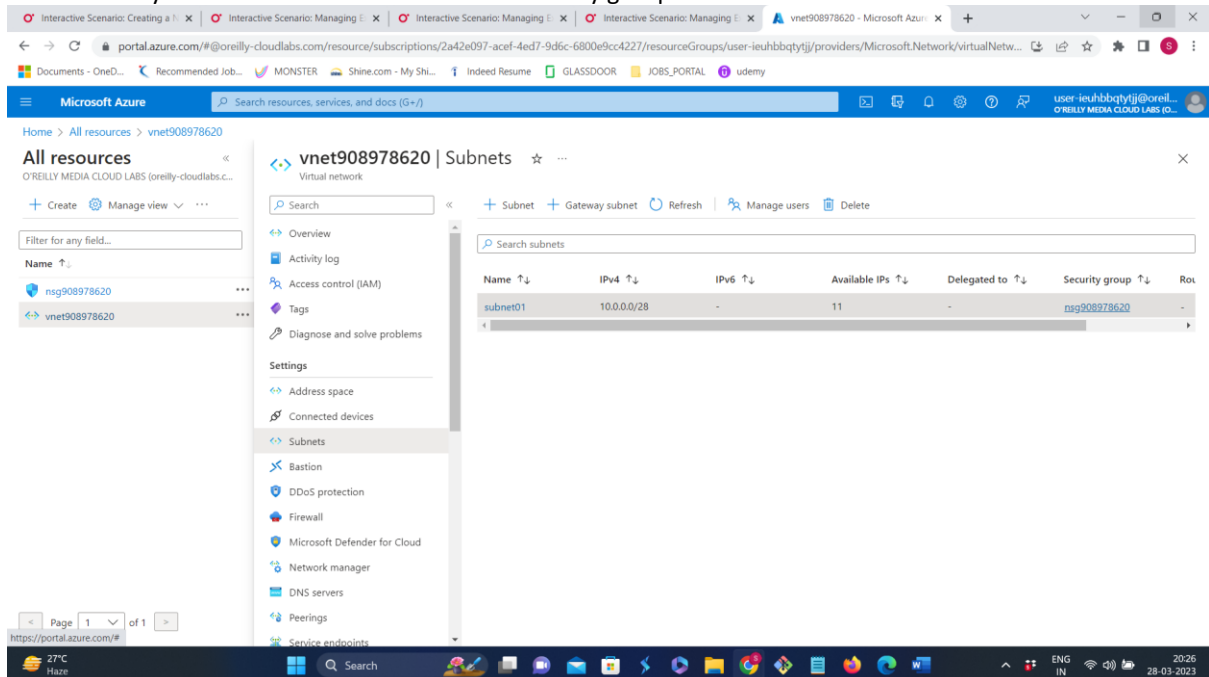
```
$ az network vnet subnet show --vnet-name $vnetName --name subnet01 -g $resource --query "{Name:name, networkSecurityGroup: networkSecurityGroup.id}"
{
  "Name": "subnet01",
  "networkSecurityGroup": "/subscriptions/2a42e097-acef-4ed7-9d6c-6800e9cc4227/resourceGroups/user-ieuhbbqytjy/providers/Microsoft.Network/networkSecurityGroups/nsg908978620"
}
```

Check the Updates in the Azure Portal

Click on your VNet name in the list (you should see only one)

Under Settings, click on Subnets

Confirm that you can see the NSG name in the Security group column



As we can see nsg rule is applied .

Visit the NSGs page

Click on your NSG name

Under Settings, click on Inbound security rules and examine them:

Every NSG has three default inbound rules.

Rule with priority 65000 allows all traffic from other servers within the same Azure VNet

Rule with Priority 65001 allows traffic from Azure Load Balancers

Rule with priority 65500 blocks the rest of the traffic

The screenshot shows the Microsoft Azure portal interface. The left sidebar displays the 'All resources' page with a filter for 'nsg908978620'. The main content area shows the 'Inbound security rules' for the network security group 'nsg908978620'. The rules are listed in a table with columns: Priority, Name, Port, Protocol, Source, and Destination. The rules are:

Priority	Name	Port	Protocol	Source	Destination
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork
65001	AllowAzureLoadBalanc...	Any	Any	AzureLoadBalancer	Any
65500	DenyAllInBound	Any	Any	Any	Any

Under Settings, click on Outbound security rules and examine them:

Every NSG has three default outbound rules.

Rule with priority 65000 allows all traffic to other servers within the same Azure VNet

Rule with Priority 65001 allows traffic to the internet

Rule with priority 65500 blocks the rest of the outgoing traffic

The screenshot shows the Microsoft Azure portal interface. The left sidebar displays the 'All resources' page with a filter for 'nsg908978620'. The main content area shows the 'Outbound security rules' for the network security group 'nsg908978620'. The rules are listed in a table with columns: Priority, Name, Port, Protocol, Source, and Destination. The rules are:

Priority	Name	Port	Protocol	Source	Destination
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork
65001	AllowInternetOutBound	Any	Any	Any	Internet
65500	DenyAllOutBound	Any	Any	Any	Any

Note: As you will see, the default NSG blocks almost all traffic to and from the subnet. You can create rules to allow/deny traffic as your scenario fits.

Delete the Azure Virtual Network and the Subnet

Use the following command to clean up the parent VNet:

```
az network vnet delete --name $vnetName --resource-group $resource
```

Use the following command to list all VNets in your allocated resource group:

```
az network vnet list --resource-group $resource
```

```
CMNaKXE0feHHS850
$ az network vnet delete --name $vnetName -g $resource
$ az network vnet list -g $resource
[]
$
```