Azure Virtual Network Manager

What is Azure Network Manager?

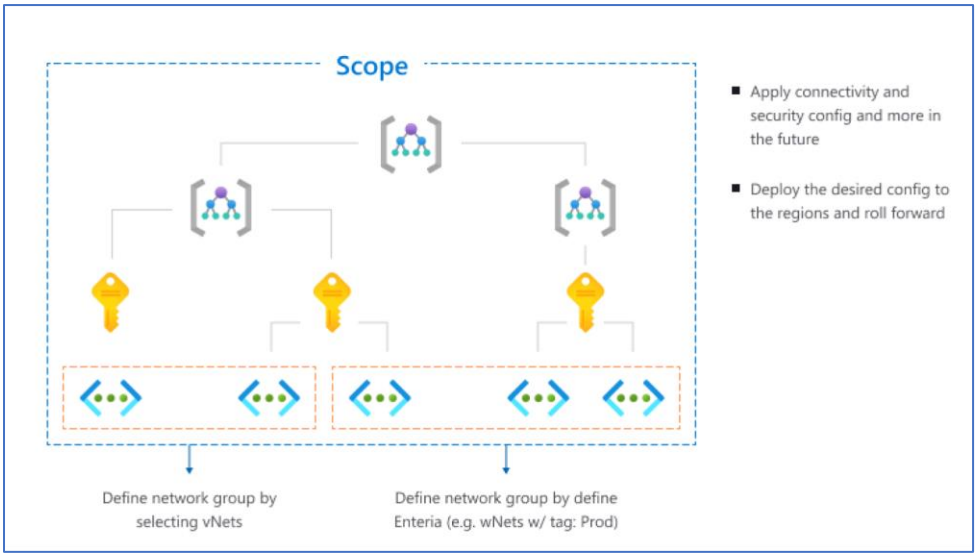
- It is used to Significantly reduce our operational overhead with Azure Virtual Network Manager.
- It is a central management service for your virtual network resources.
- Easily manage your virtual network infrastructure while scaling your cloud-based workloads.
- We can use the centralized solution to create and manage complex network topologies and network security rules globally across subscriptions.

Four Main Advantages of VNM:

- Global management of virtual network resources across regions and subscriptions
- Automated management of complex virtual network topologies such as hub and spoke and mesh
- Organization-level security rule enforcement at scale
- Simple deployment of configurations to test in specific regions

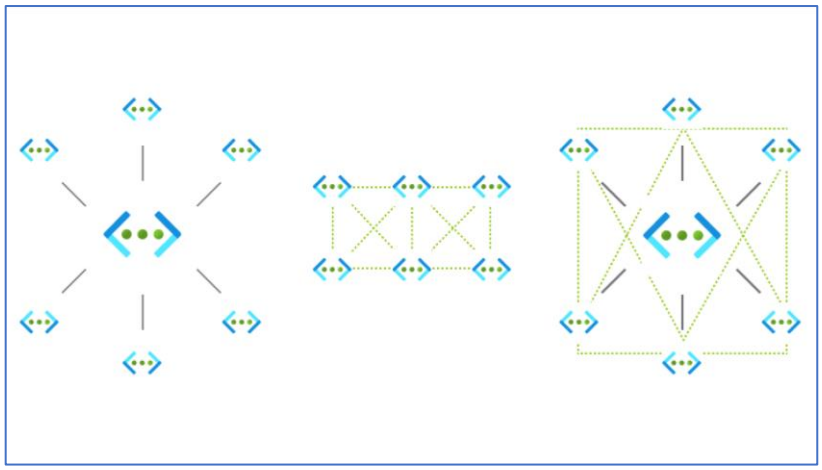
How to manage virtual network resources across subscriptions?

Apply security and connectivity configurations for all your virtual networks across regions and subscriptions. Manage the configurations for your entire environment from a single pane of glass.



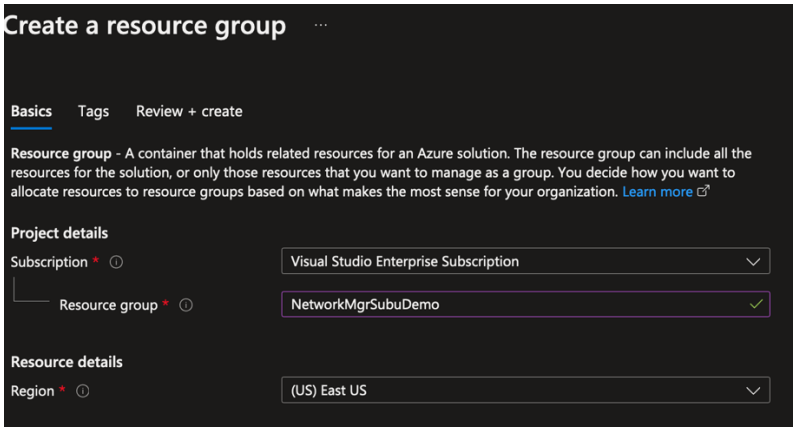
How are the Topologies applied to Vnet?

Create complex virtual network topologies such as hub and spoke and mesh in just a few clicks. Azure Virtual Network Manager automatically responds to the changes you've made and maintains the virtual network topology.



<u>Create a new Resource Group:</u>

Login to Azure portal (https://portal.azure.com/) and then create a new RG



Now we need to create multiple Vnet to manage it in a single umbrella so we need to run the script to create all at the same time.

Now am opening the Azure Cloud Shell to run the below shell.

```
##################
#    Check RG    #
##################
```

```
$RGName = 'NetworkMgrSubuDemo'
If (!(Get-AzResourceGroup -name $RGName -ErrorAction SilentlyContinue)) {
    Write-Host -ForegroundColor Red -BackgroundColor Black "Creating Resource Group"
    New-AzResourceGroup -Name $RGName -Location 'eastus'
}
else {
    Write-Host -ForegroundColor Cyan -BackgroundColor Black "Resource Group $RGName already exists"
}


####################
#    Input Array   #
####################
$vNETs = @(
    @{Name="vNETMgr-0-Prod";Location='northcentralus';AddressPrefix='192.168.0.0/24'}
    @{Name="vNETMgr-0-Dev";Location='northcentralus';AddressPrefix='192.168.1.0/24'}
    @{Name="vNETMgr-1-Prod";Location='westus';AddressPrefix='192.168.2.0/24'}
    @{Name="vNETMgr-1-Dev";Location='westus';AddressPrefix='192.168.3.0/24'}
    @{Name="vNETMgr-2-Prod";Location='eastus';AddressPrefix='192.168.4.0/24'}
    @{Name="vNETMgr-2-Dev";Location='eastus';AddressPrefix='192.168.5.0/24'}
    @{Name="vNETMgr-3-Prod";Location='eastus2';AddressPrefix='192.168.6.0/24'}
    @{Name="vNETMgr-3-Dev";Location='eastus2';AddressPrefix='192.168.7.0/24'}
    @{Name="vNETMgr-4-Prod";Location='westus2';AddressPrefix='192.168.8.0/24'}
    @{Name="vNETMgr-4-Dev";Location='westus2';AddressPrefix='192.168.9.0/24'}
    @{Name="vNETMgr-5-Prod";Location='NorthEurope';AddressPrefix='192.168.10.0/24'}
    @{Name="vNETMgr-5-Dev";Location='NorthEurope';AddressPrefix='192.168.11.0/24'}
    @{Name="vNETMgr-6-Prod";Location='WestEurope';AddressPrefix='192.168.12.0/24'}
    @{Name="vNETMgr-6-Dev";Location='WestEurope';AddressPrefix='192.168.13.0/24'}
    @{Name="vNETMgr-7-Prod";Location='franceCentral';AddressPrefix='192.168.14.0/24'}
    @{Name="vNETMgr-7-Dev";Location='franceCentral';AddressPrefix='192.168.15.0/24'}
    @{Name="vNETMgr-0-Lab";Location='eastus';AddressPrefix='172.18.0.0/24'}
    @{Name="vNETMgr-1-Lab";Location='eastus';AddressPrefix='172.18.1.0/24'}
    @{Name="vNETMgr-2-Lab";Location='eastus';AddressPrefix='172.18.2.0/24'}
    @{Name="vNETMgr-3-Lab";Location='eastus';AddressPrefix='172.18.3.0/24'}
)


######################
#    Build vNETs    #
######################
foreach ($vNET in $vNETs) {
    $Subnet = New-AzVirtualNetworkSubnetConfig -Name Subnet -AddressPrefix $vNET.AddressPrefix
    New-AzVirtualNetwork `
        -Name $vnet.Name `
        -ResourceGroupName $RGName `
        -Location $vNET.Location `
        -AddressPrefix $vNET.AddressPrefix `
        -Subnet $Subnet
```

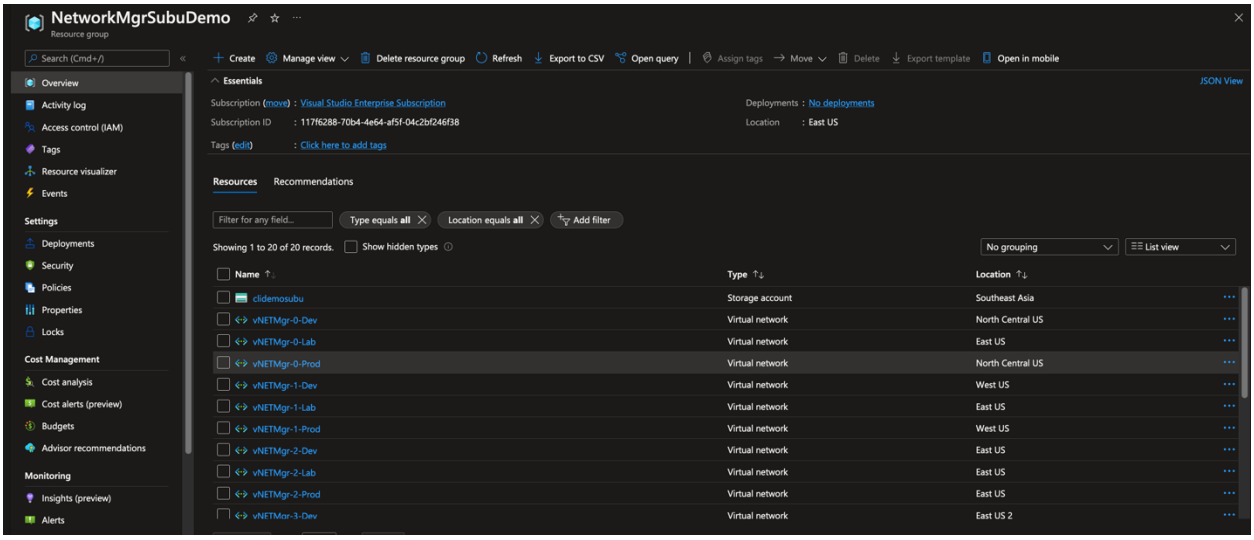
```
}
```

The above script is taken from author (https://raw.githubusercontent.com/DeanCefola/PowerShell-Scripts/master/Build%20Resources/Build_vNET.ps1)

```
PowerShell
PS /home/f24b46f7-f757-4bc9-a93d-9a0047fe> $vNETs = @(
>>      @{Name="vNETMgr-0-Prod";Location='northcentralus';AddressPrefix='192.168.0.0/24'}
>>      @{Name="vNETMgr-0-Dev";Location='northcentralus';AddressPrefix='192.168.1.0/24'}
>>      @{Name="vNETMgr-1-Prod";Location='westus';AddressPrefix='192.168.2.0/24'}
>>      @{Name="vNETMgr-1-Dev";Location='westus';AddressPrefix='192.168.3.0/24'}
>>      @{Name="vNETMgr-2-Prod";Location='eastus';AddressPrefix='192.168.4.0/24'}
>>      @{Name="vNETMgr-2-Dev";Location='eastus';AddressPrefix='192.168.5.0/24'}
>>      @{Name="vNETMgr-3-Prod";Location='eastus2';AddressPrefix='192.168.6.0/24'}
>>      @{Name="vNETMgr-3-Dev";Location='eastus2';AddressPrefix='192.168.7.0/24'}
>>      @{Name="vNETMgr-4-Prod";Location='westus2';AddressPrefix='192.168.8.0/24'}
>>      @{Name="vNETMgr-4-Dev";Location='westus2';AddressPrefix='192.168.9.0/24'}
>>      @{Name="vNETMgr-5-Prod";Location='NorthEurope';AddressPrefix='192.168.10.0/24'}
>>      @{Name="vNETMgr-5-Dev";Location='NorthEurope';AddressPrefix='192.168.11.0/24'}
>>      @{Name="vNETMgr-6-Prod";Location='WestEurope';AddressPrefix='192.168.12.0/24'}
>>      @{Name="vNETMgr-6-Dev";Location='WestEurope';AddressPrefix='192.168.13.0/24'}
>>      @{Name="vNETMgr-7-Prod";Location='franceCentral';AddressPrefix='192.168.14.0/24'}
>>      @{Name="vNETMgr-7-Dev";Location='franceCentral';AddressPrefix='192.168.15.0/24'}
>>      @{Name="vNETMgr-0-Lab";Location='eastus';AddressPrefix='172.18.0.0/24'}
>>      @{Name="vNETMgr-1-Lab";Location='eastus';AddressPrefix='172.18.1.0/24'}
>>      @{Name="vNETMgr-2-Lab";Location='eastus';AddressPrefix='172.18.2.0/24'}
>>      @{Name="vNETMgr-3-Lab";Location='eastus';AddressPrefix='172.18.3.0/24'}
>> )
PS /home/f24b46f7-f757-4bc9-a93d-9a0047fe>
PS /home/f24b46f7-f757-4bc9-a93d-9a0047fe>
PS /home/f24b46f7-f757-4bc9-a93d-9a0047fe> ####################
PS /home/f24b46f7-f757-4bc9-a93d-9a0047fe> #    Build vNETs    #
PS /home/f24b46f7-f757-4bc9-a93d-9a0047fe> ####################
PS /home/f24b46f7-f757-4bc9-a93d-9a0047fe> foreach ($vNET in $vNETs) {
>>      $Subnet = New-AzVirtualNetworkSubnetConfig -Name Subnet -AddressPrefix $vNET.AddressPrefix
>>      New-AzVirtualNetwork `
>>          -Name $vnet.Name `
>>          -ResourceGroupName $RGName `
>>          -Location $vNET.Location `
>>          -AddressPrefix $vNET.AddressPrefix `
>>          -Subnet $Subnet
>> }
```

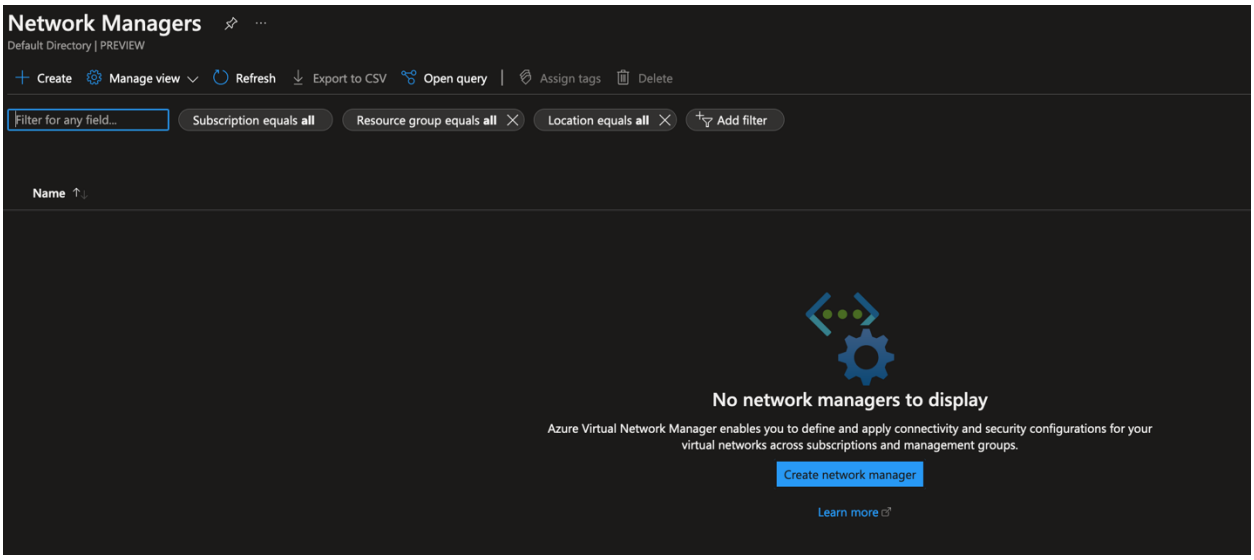
```
Name                   : vNETMgr-0-Prod
ResourceGroupName      : NetworkMgrSubuDemo
Location               : northcentralus
Id                     : /subscriptions/117f6288-70b4-4e64-af5f-04c2bf246f38/resourceGroups/NetworkMgrSubuDemo/providers/Microsoft.Network/virtualNetworks/vNETMgr-0-Prod
Etag                   : W/"150ff335-439b-45ad-a690-f6591794a4d6"
ResourceGuid           : b5af47a0-5835-4ef2-b3c1-6efea3a43030
ProvisioningState      : Succeeded
Tags                   :
AddressSpace           : {
                           "AddressPrefixes": [
                             "192.168.0.0/24"
                           ]
                         }
DhcpOptions            : {}
FlowTimeoutInMinutes   : null
Subnets                : [
                           {
                             "Delegations": [],
                             "Name": "Subnet",
                             "Etag": "W/\"150ff335-439b-45ad-a690-f6591794a4d6\"",
                             "Id":
                         "/subscriptions/117f6288-70b4-4e64-af5f-04c2bf246f38/resourceGroups/NetworkMgrSubuDemo/providers/Microsoft.Network/virtualNetworks/vNETMgr-0-Prod/subnets/Subnet",
                             "AddressPrefix": [
                               "192.168.0.0/24"
                             ],
                             "IpConfigurations": [],
                             "ServiceAssociationLinks": [],
                             "ResourceNavigationLinks": [],
                             "ServiceEndpoints": [],
                             "ServiceEndpointPolicies": [],
                             "PrivateEndpoints": []
                             "ProvisioningState": "Succeeded",
                             "PrivateEndpointNetworkPolicies": "Disabled",
                             "PrivateLinkServiceNetworkPolicies": "Enabled",
                             "IpAllocations": []
                           }
                         ]
VirtualNetworkPeerings : []
EnableDdosProtection   : false
DdosProtectionPlan     : null
```

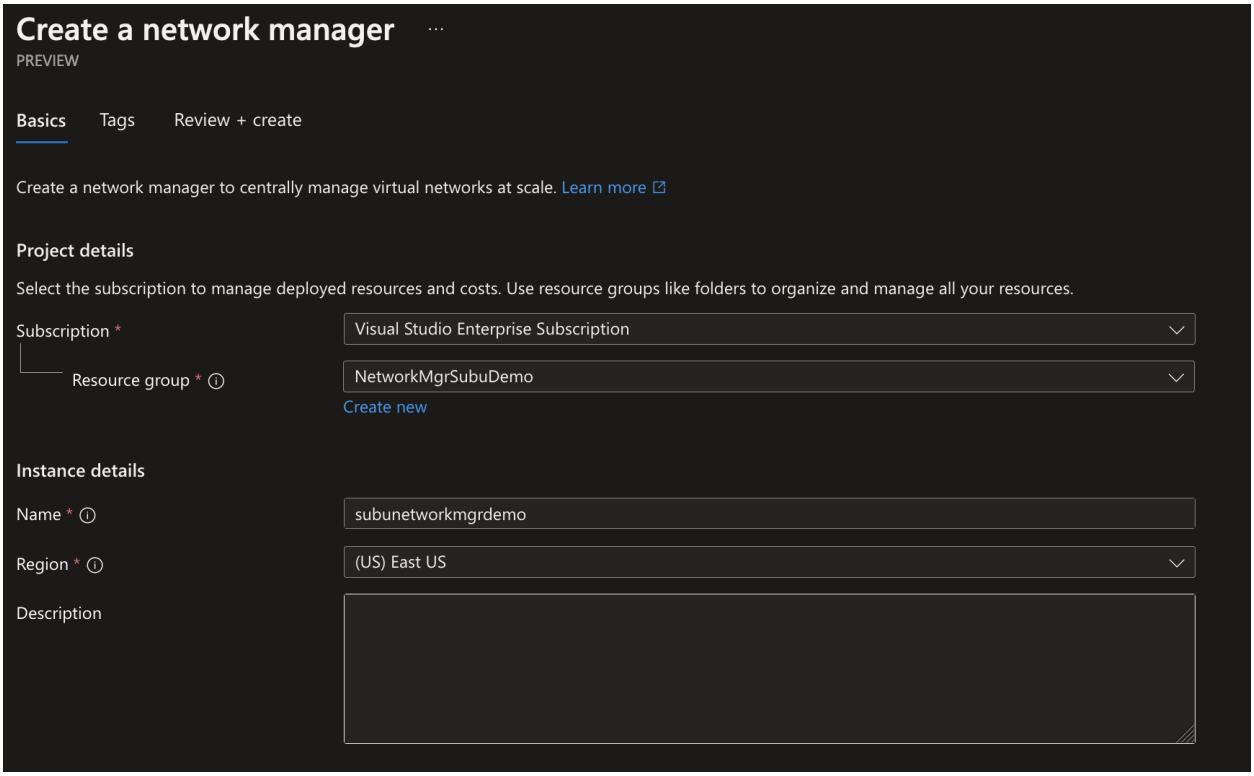
Once after we run the script then all the VNET will be created accordingly and then we can then manage all in 1 place.



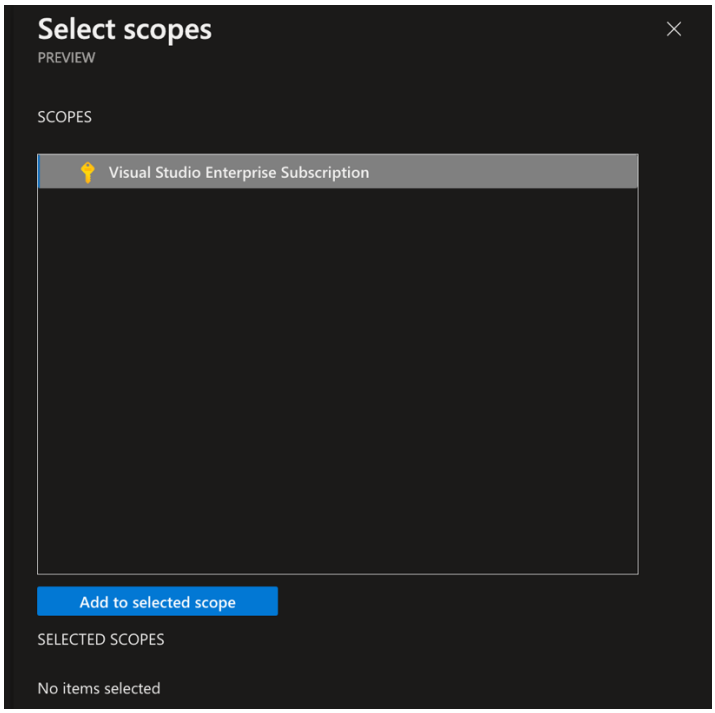
Now all our VNET is ready, so let's create the Network Manager.

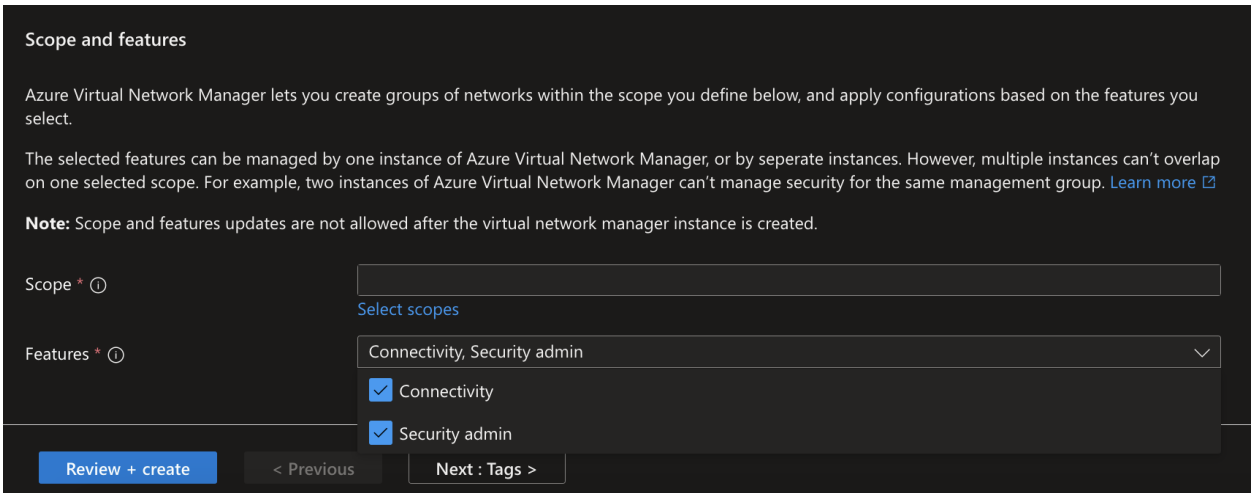
Login to Azure and then type the Network Manager and then use the RG that we created before so that we can see how our all VNET is kept in place.

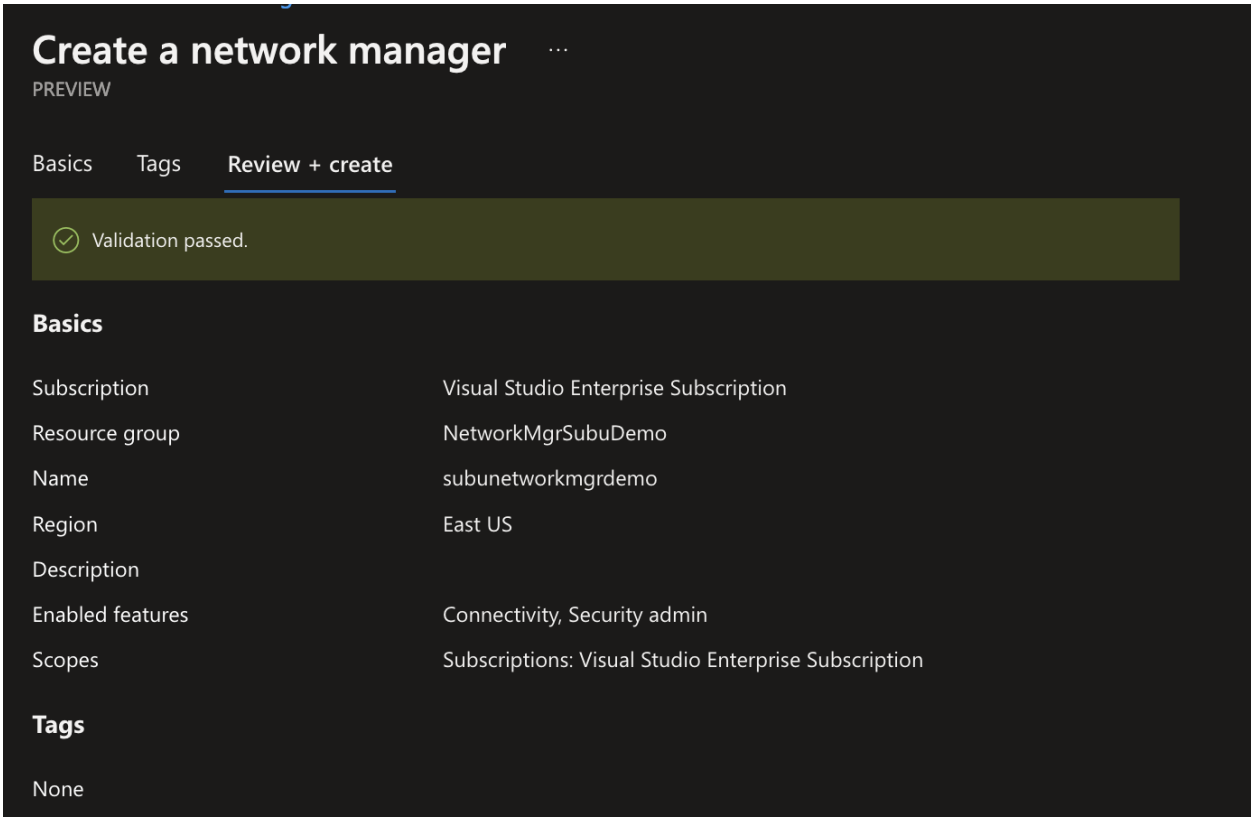


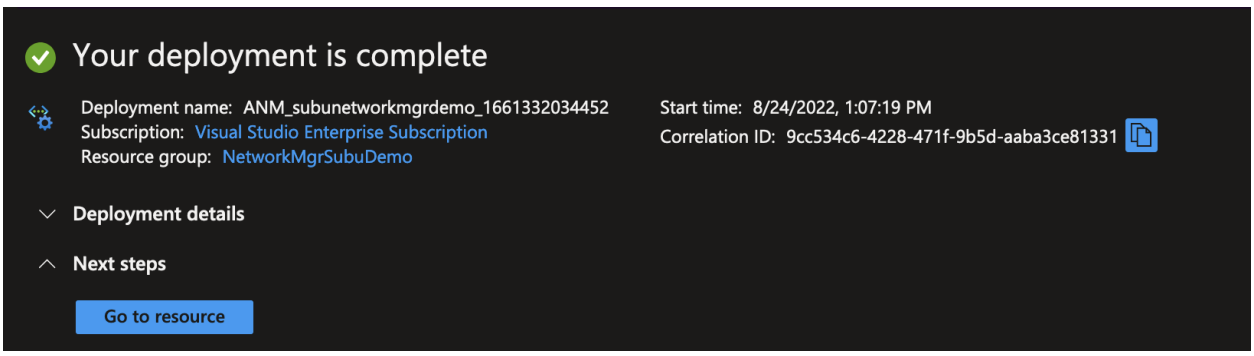


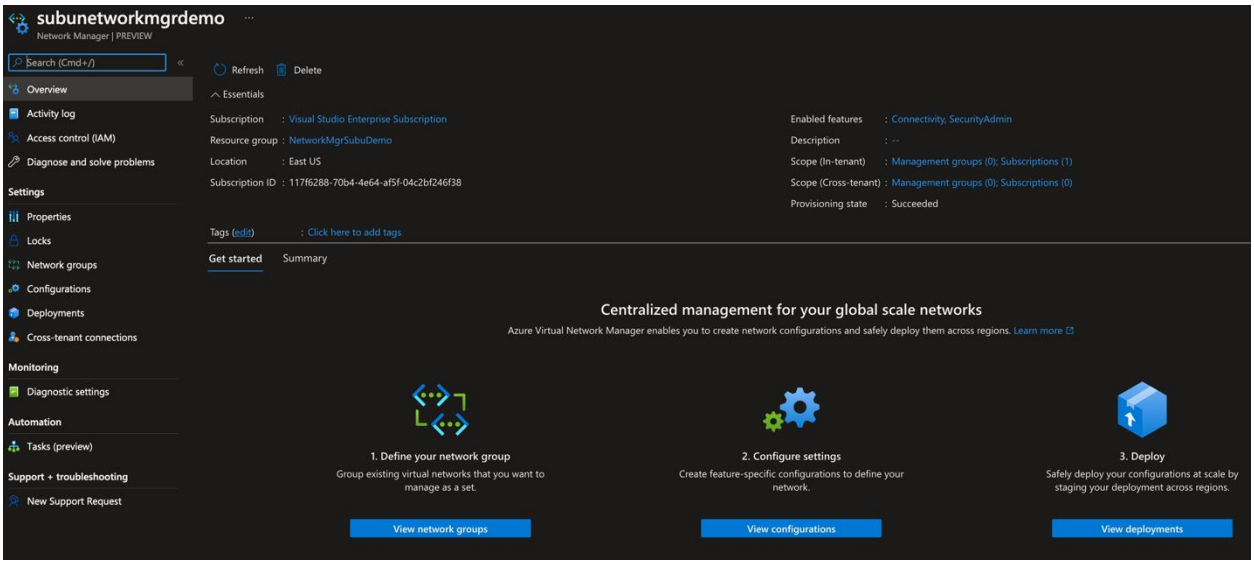
Choose the scope accordingly based on the Subscription or Mgmt Group side.



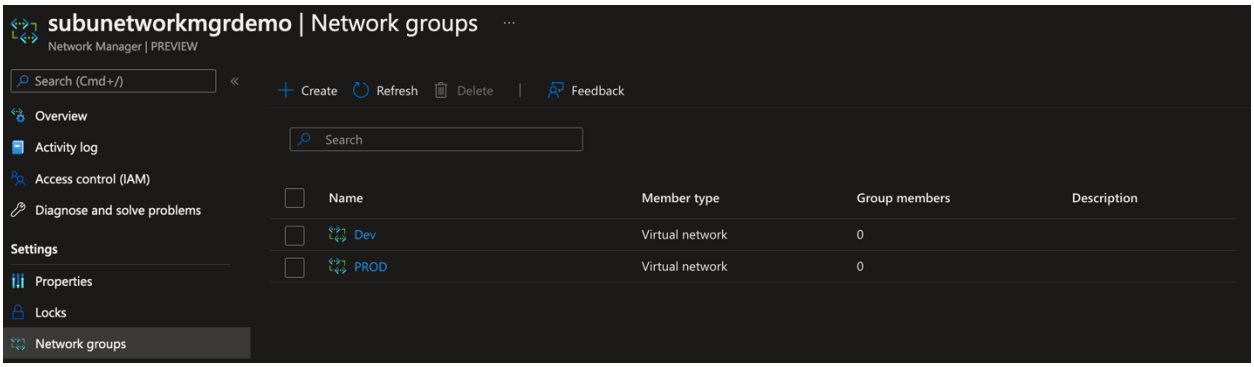

Scope and features
Azure Virtual Network Manager lets you create groups of networks within the scope you define below, and apply configurations based on the features you select.
The selected features can be managed by one instance of Azure Virtual Network Manager, or by seperate instances. However, multiple instances can't overlap on one selected scope. For example, two instances of Azure Virtual Network Manager can't manage security for the same management group. Learn more
Note: Scope and features updates are not allowed after the virtual network manager instance is created.
Scope *
Select scopes
Features *
Connectivity, Security admin
Connectivity
Security admin
Review + create
< Previous
Next : Tags >


Create a network manager
PREVIEW
Basics
Tags
Review + create
Validation passed.
Basics
Subscription
Visual Studio Enterprise Subscription
Resource group
NetworkMgrSubuDemo
Name
subunetworkmgrdemo
Region
East US
Description
Enabled features
Connectivity, Security admin
Scopes
Subscriptions: Visual Studio Enterprise Subscription
Tags
None

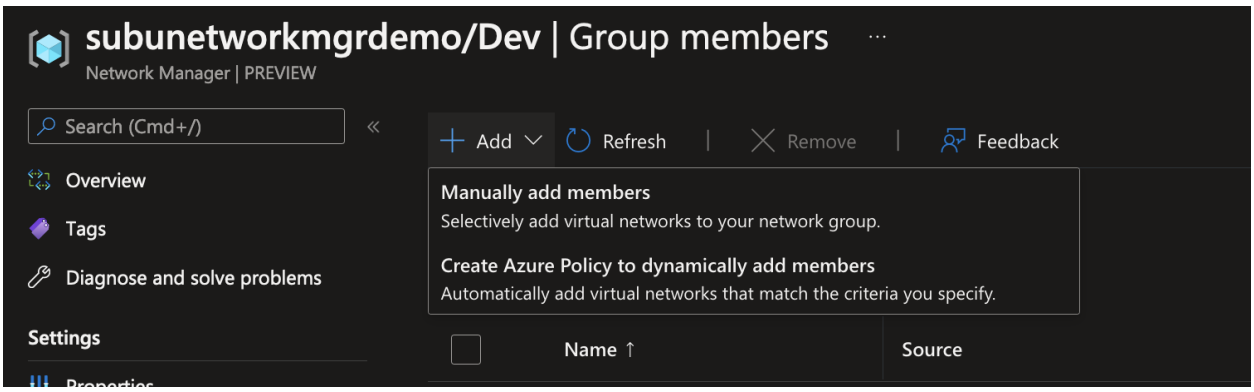

Your deployment is complete
Deployment name: ANM_subunetworkmgrdemo_1661332034452
Subscription: Visual Studio Enterprise Subscription
Resource group: NetworkMgrSubuDemo
Start time: 8/24/2022, 1:07:19 PM
Correlation ID: 9cc534c6-4228-471f-9b5d-aaba3ce81331
Deployment details
Next steps
Go to resource

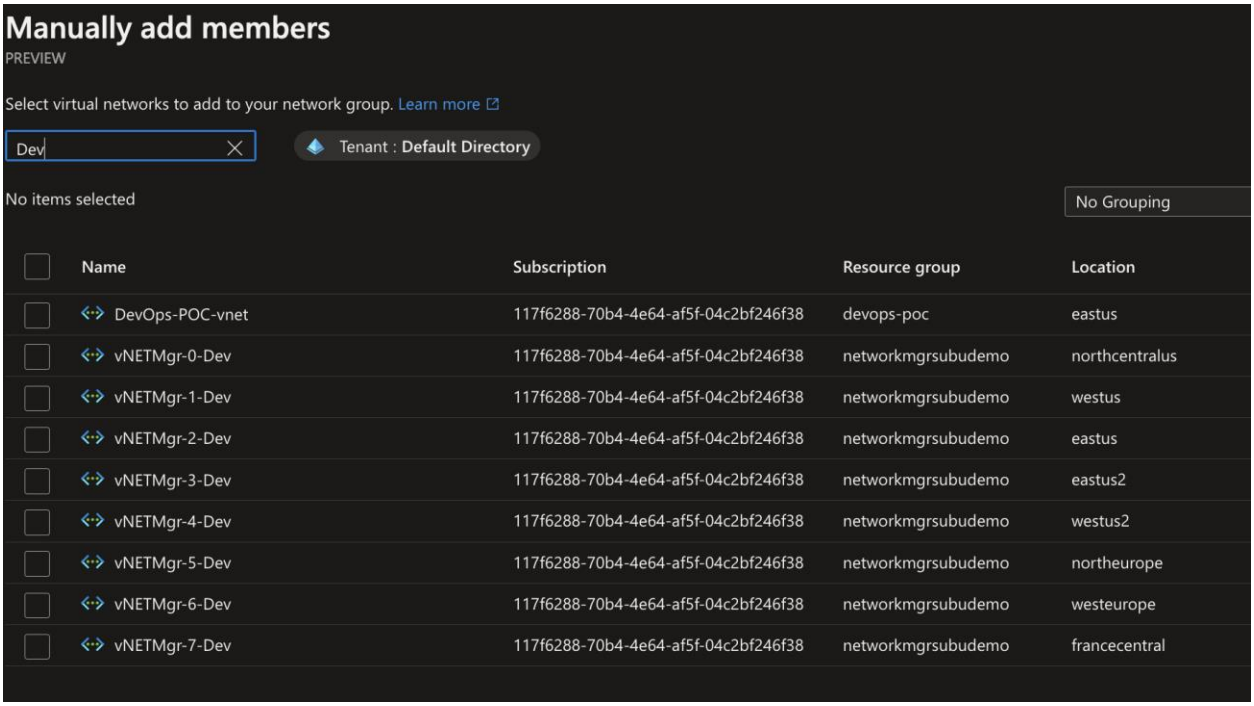


Once the Network Manager is ready, we can get to see that 3 settings such as network group and configure settings.

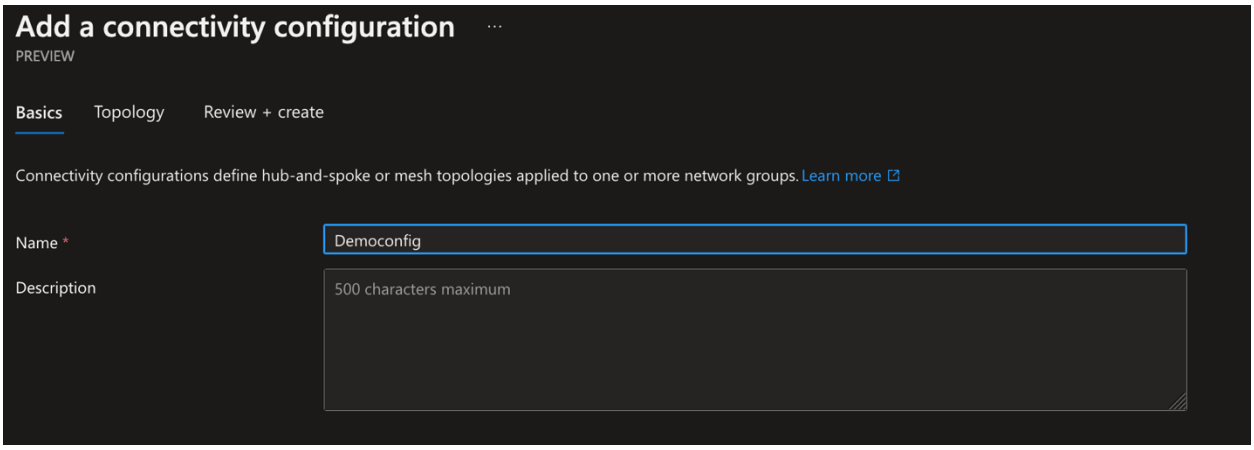


Then we need to enter the group and add the VNET respectively.





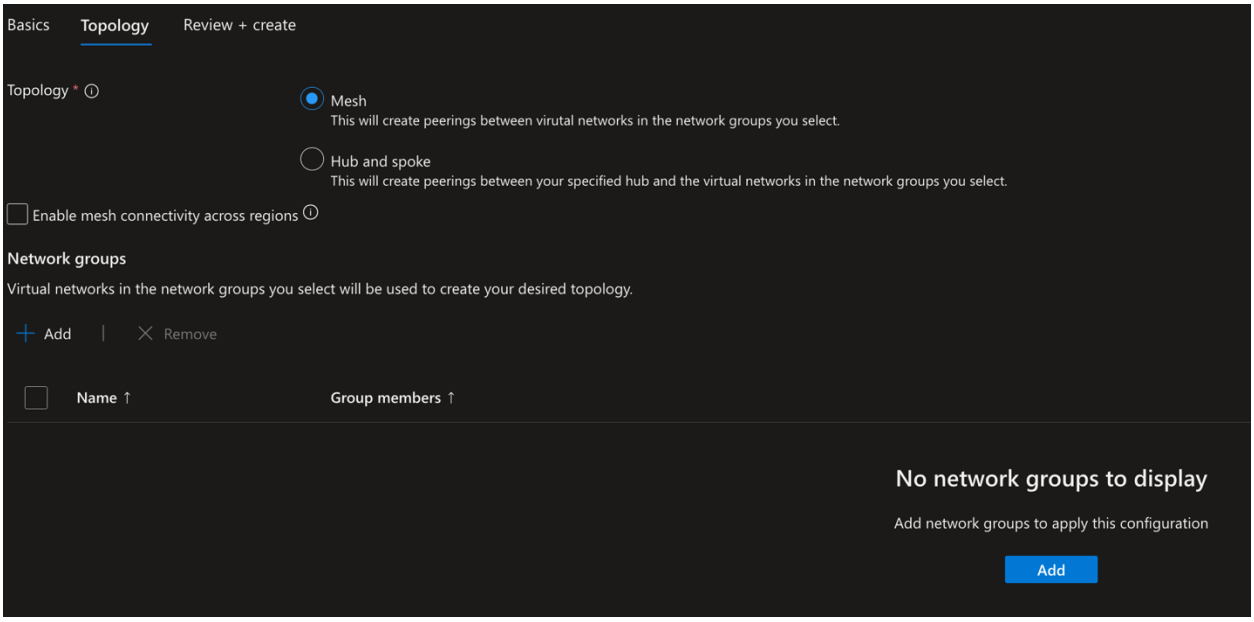
Then we need to add a new Topology Connectivity.

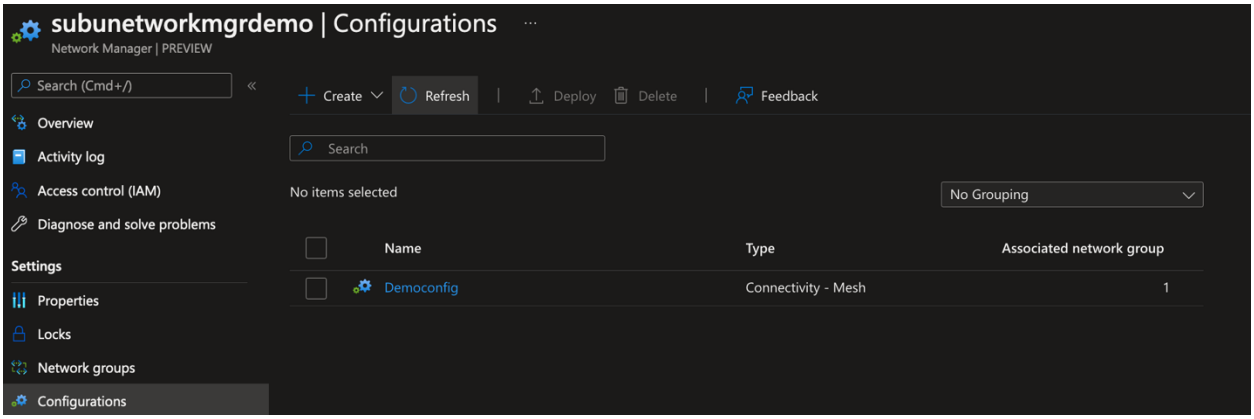


We can see there are 2 types of Topologies such as below.

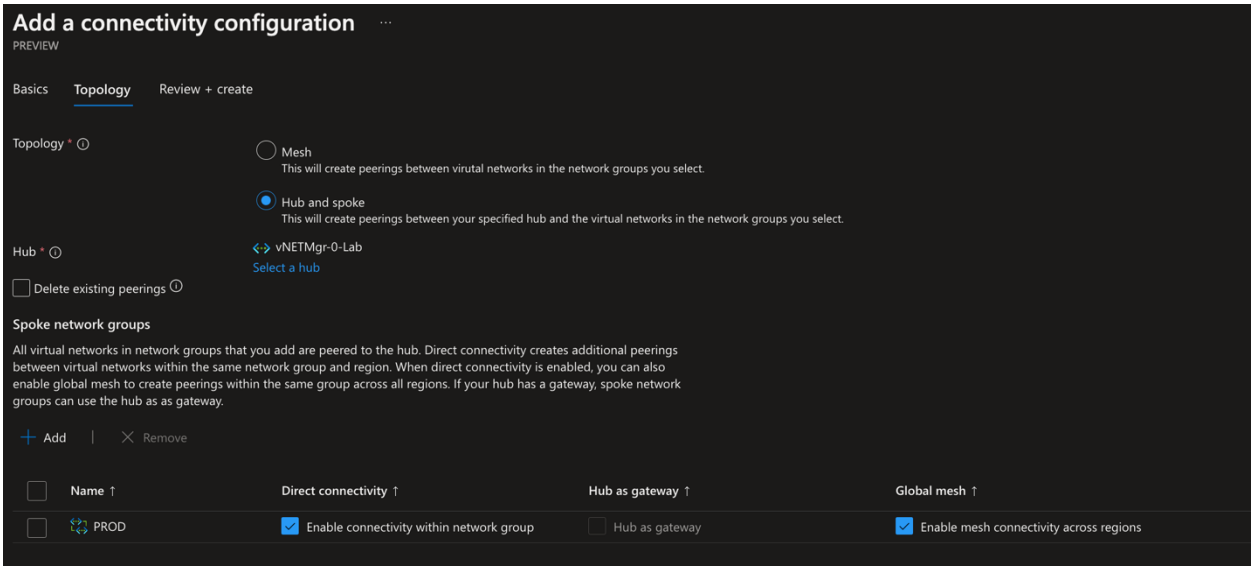
1. Mesh Based model
2. Hub and Spoke model

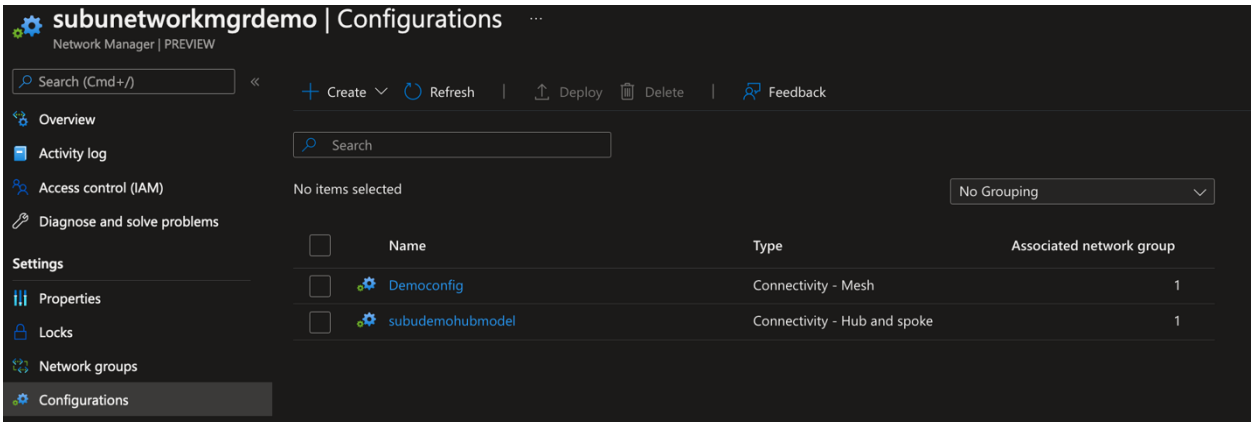
This is based on the Mesh model setup.



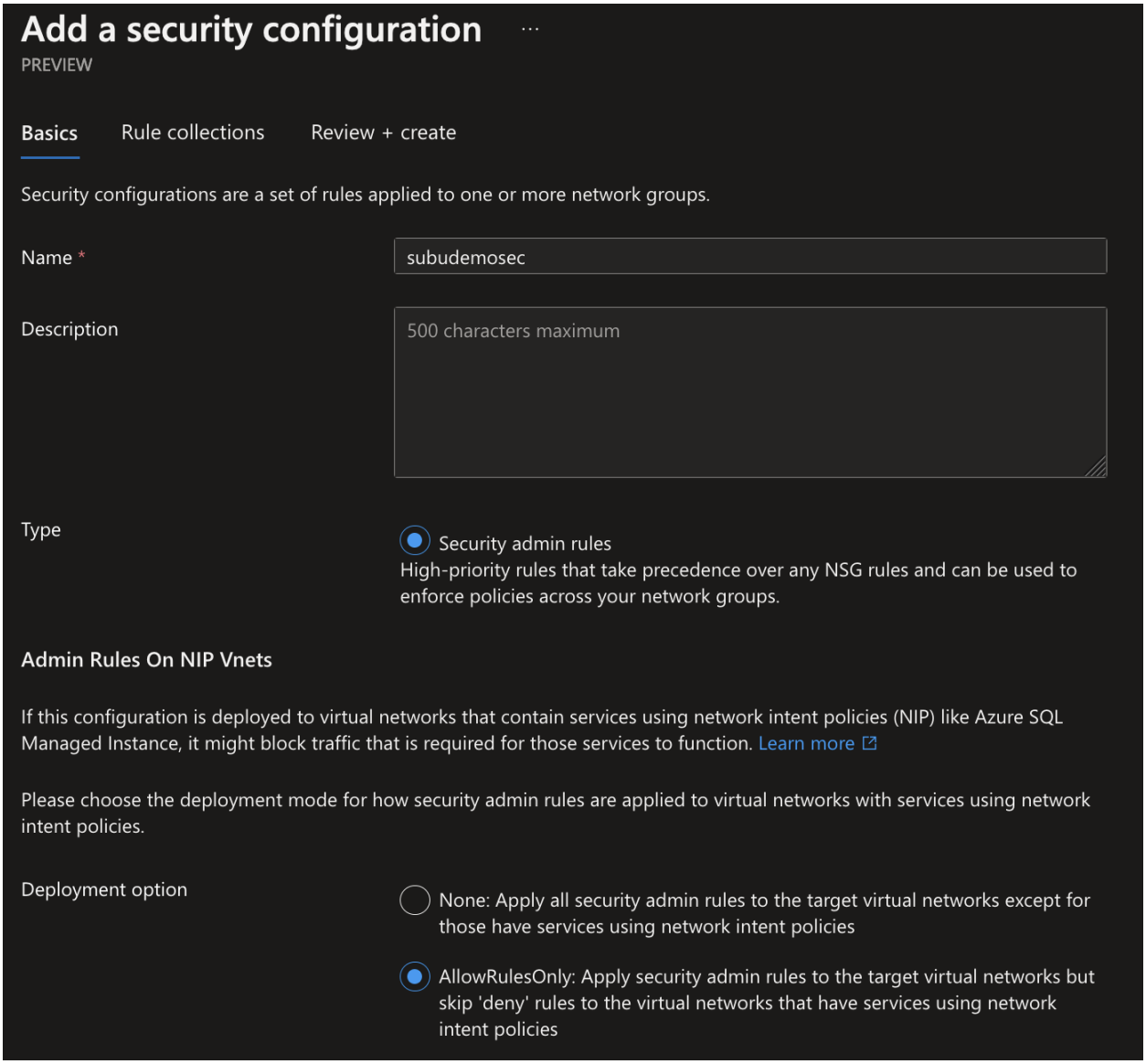


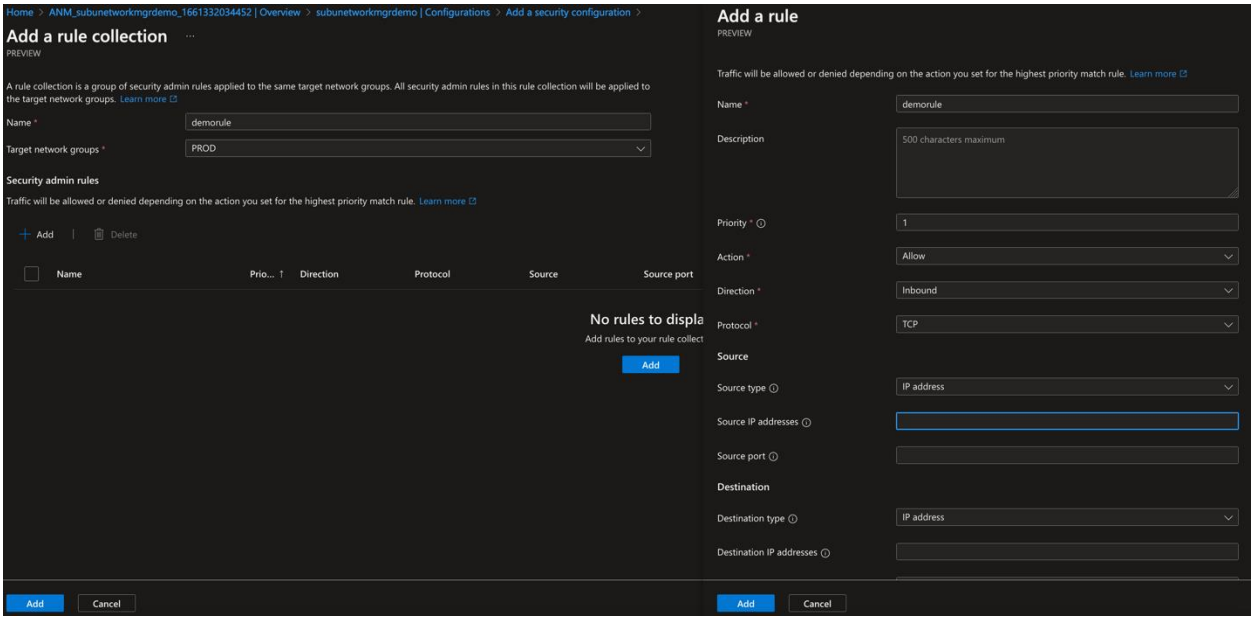
This is based on the Hub and Spoke model setup.

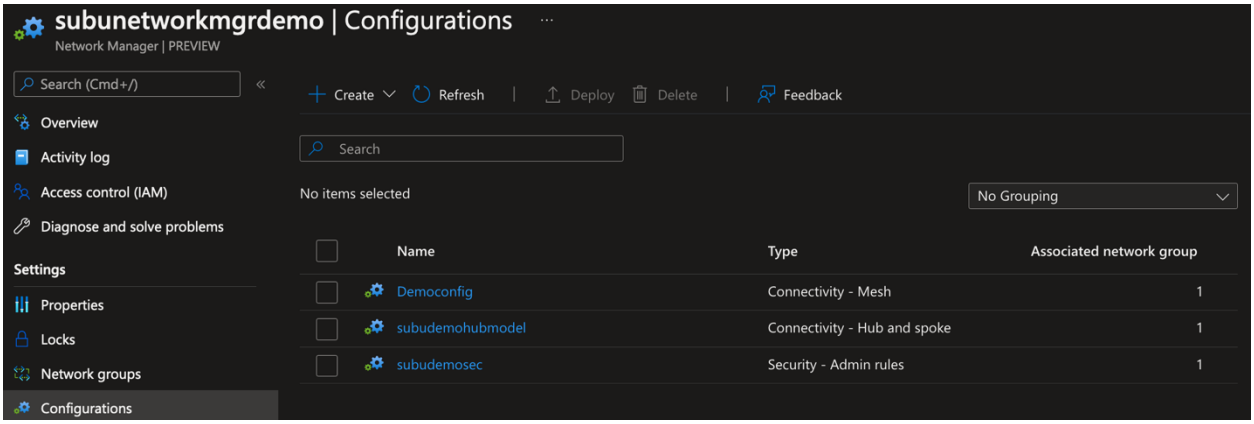




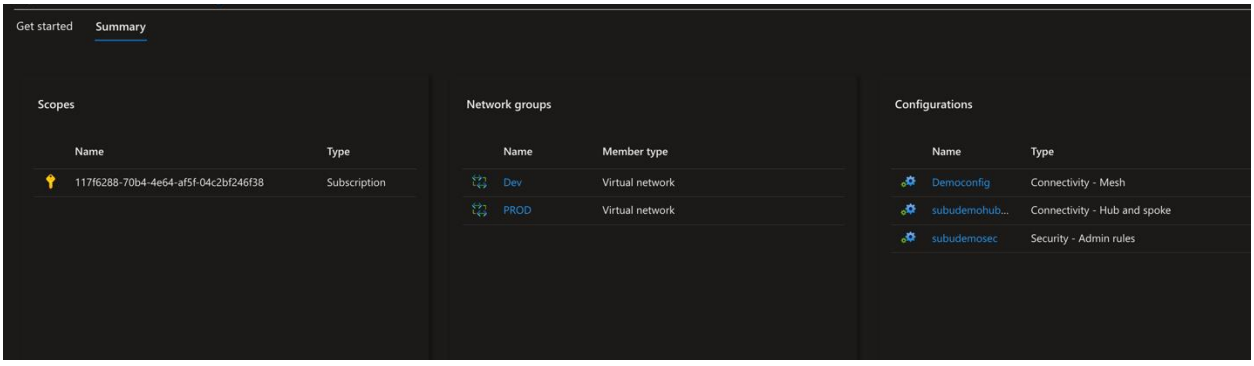
Next, we can setup the security configurations in this place.


Add a security configuration
PREVIEW
Basics
Rule collections
Review + create
Security configurations are a set of rules applied to one or more network groups.
Name *
subudemosec
Description
500 characters maximum
Type
Security admin rules
High-priority rules that take precedence over any NSG rules and can be used to enforce policies across your network groups.
Admin Rules On NIP Vnets
If this configuration is deployed to virtual networks that contain services using network intent policies (NIP) like Azure SQL Managed Instance, it might block traffic that is required for those services to function. Learn more
Please choose the deployment mode for how security admin rules are applied to virtual networks with services using network intent policies.
Deployment option
None: Apply all security admin rules to the target virtual networks except for those have services using network intent policies
AllowRulesOnly: Apply security admin rules to the target virtual networks but skip 'deny' rules to the virtual networks that have services using network intent policies



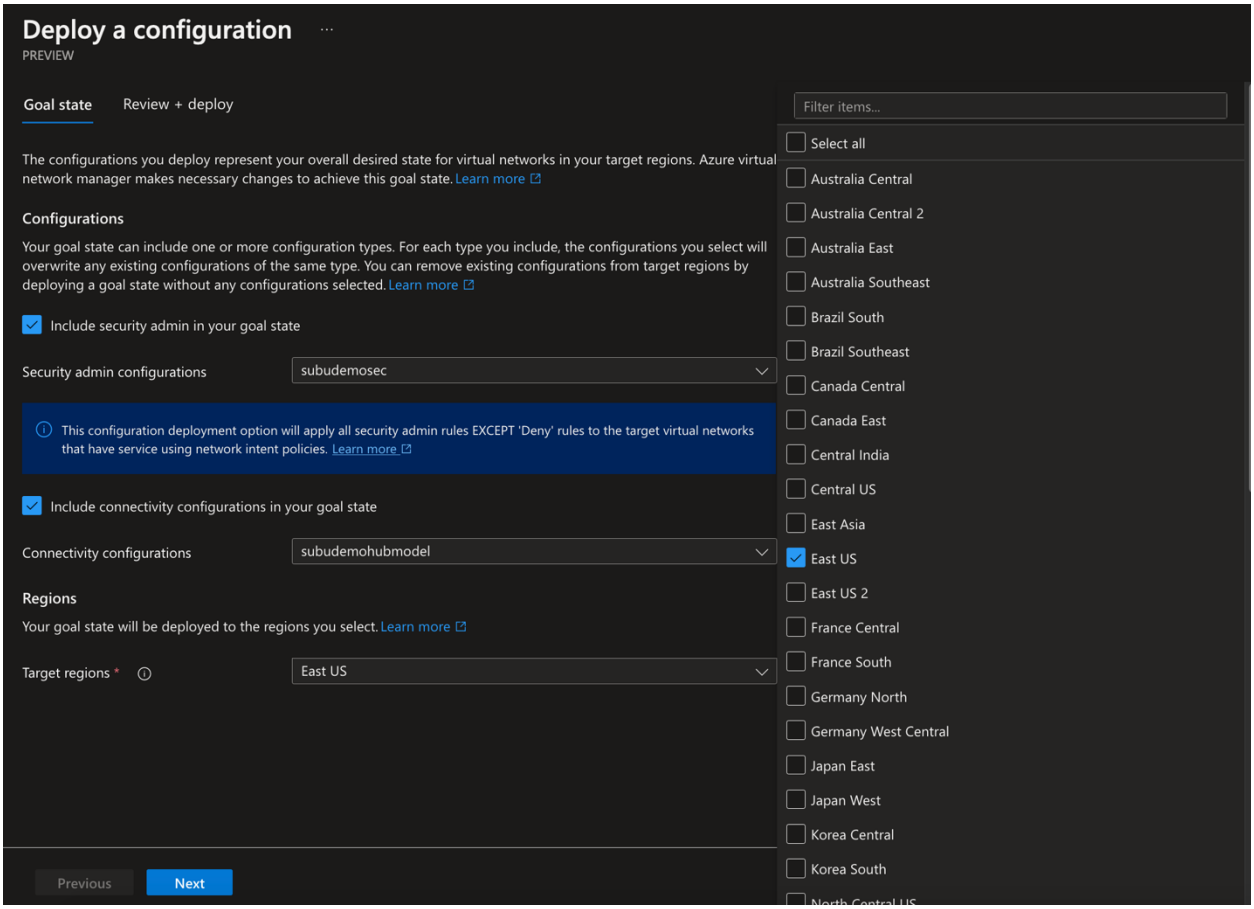


We can see all the Summary in 1 page.

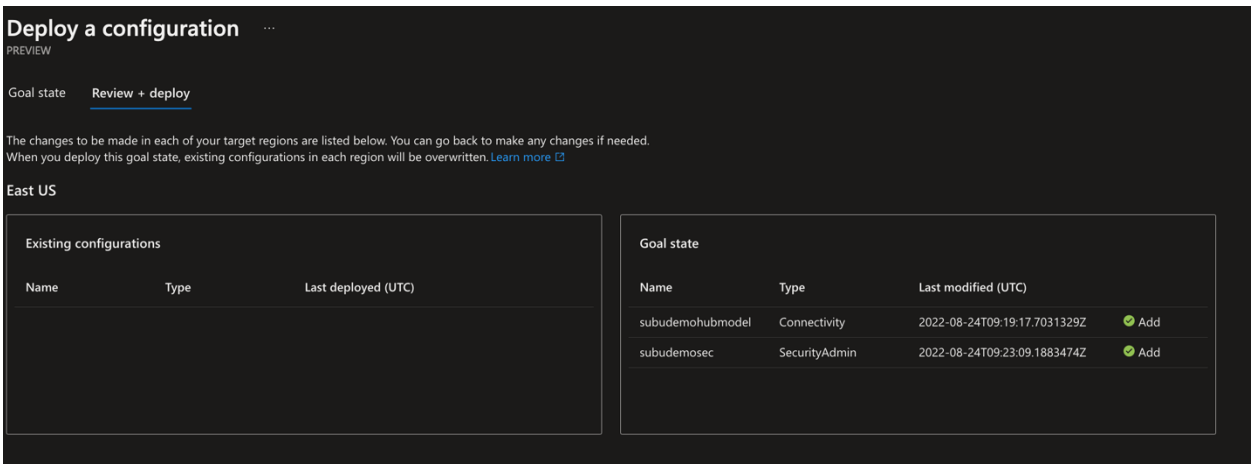


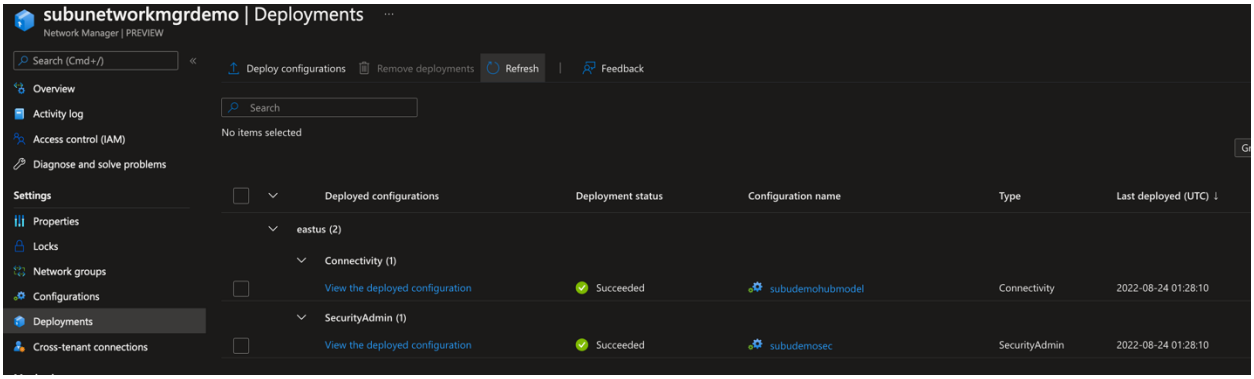
We can even do the deployments using this single panel.

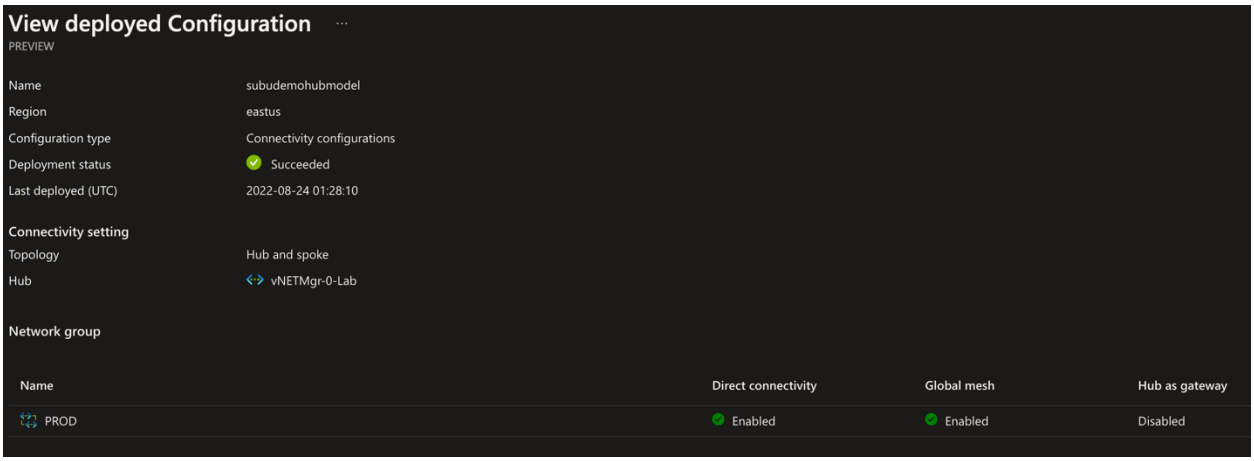
Deploy configurations that represent your goal state for your target regions. You can deploy connectivity configurations, security configurations, or both. The configurations you deploy will be applied to virtual networks in associated network groups within the regions you select.



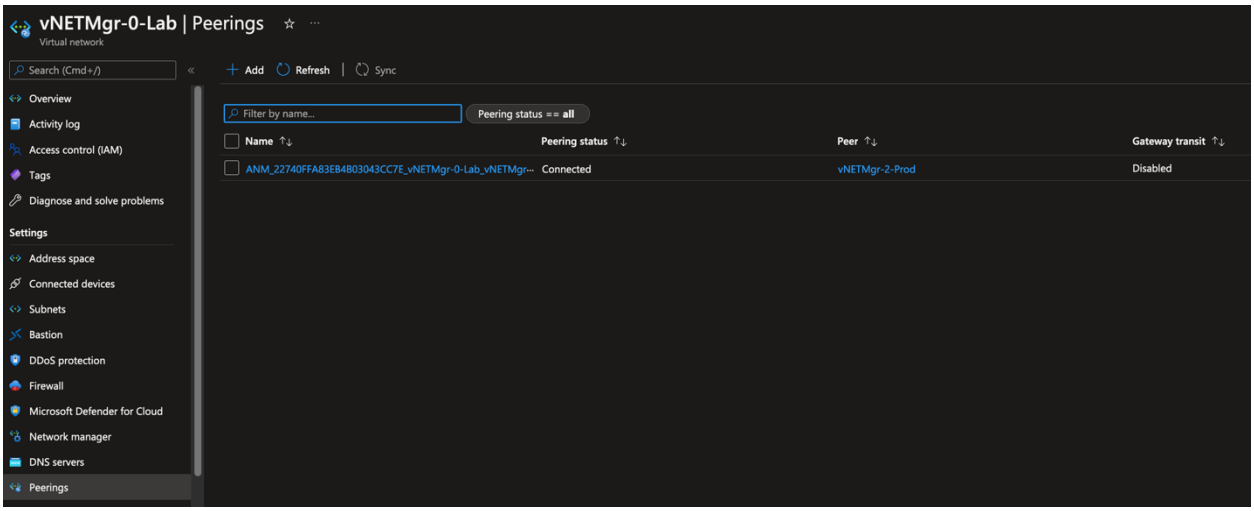
We can change the regions and deploy only to 1 or 2 specific regions if needed from 1 place.







We can see the deployment if its deploy properly by going into the VNET and then see the peering if it happened properly or not.



So basically with this Network Manager we can use this to do 80 % of the work in the 20% of efforts .