

# Azure Virtual Network Manager

## What is Azure Network Manager?

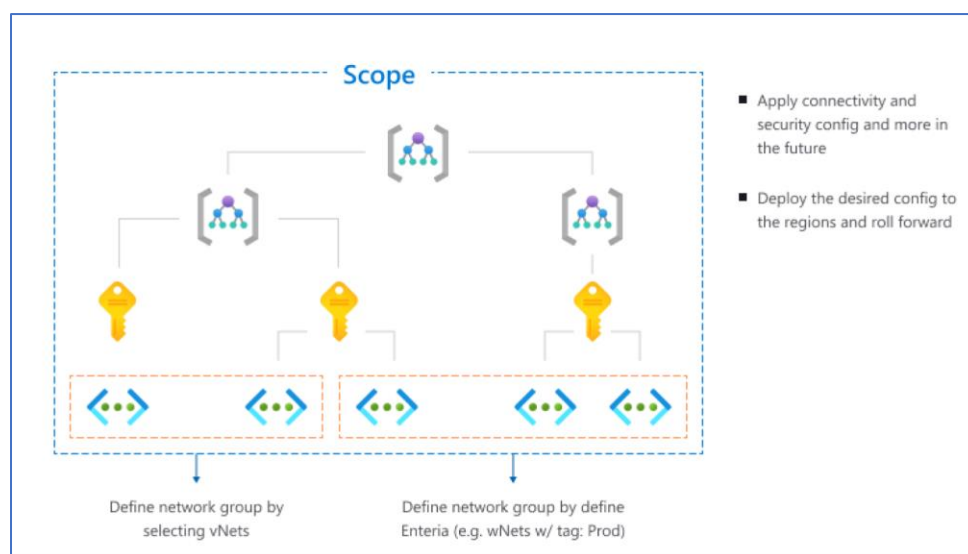
- It is used to Significantly reduce our operational overhead with Azure Virtual Network Manager.
- It is a central management service for your virtual network resources.
- Easily manage your virtual network infrastructure while scaling your cloud-based workloads.
- We can use the centralized solution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

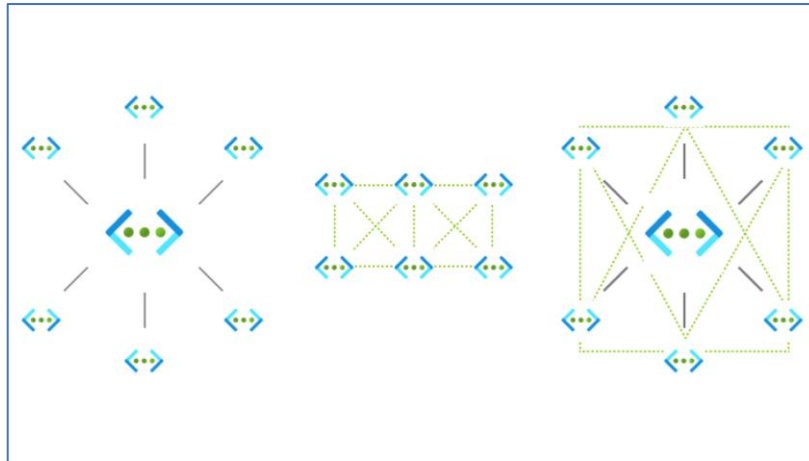
## How to manage virtual network resources across subscriptions?

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.



## Create a new Resource Group:

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

**Create a resource group** ...

Basics Tags Review + create

Resource group - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. [Learn more](#) ⓘ

**Project details**

Subscription \* ⓘ Visual Studio Enterprise Subscription ▼

Resource group \* ⓘ NetworkMgrSubuDemo ✓

**Resource details**

Region \* ⓘ (US) East US ▼

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
}

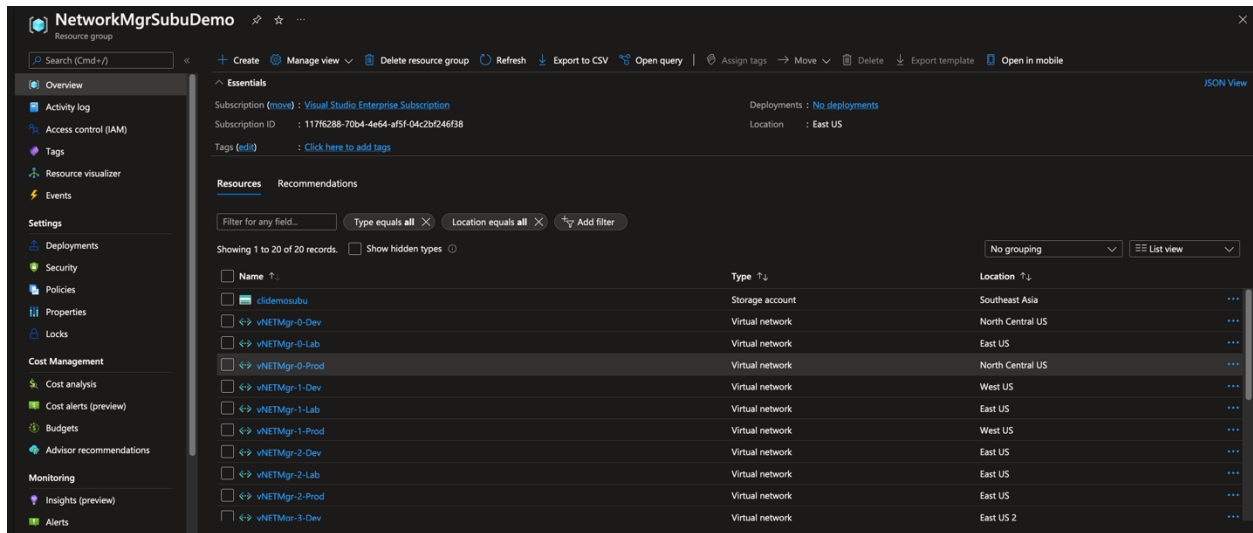
```

The above script is taken from author ( [https://raw.githubusercontent.com/DeanCefola/PowerShell-Scripts/master/Build%20Resources/Build\\_vNET.ps1](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
>> }
```

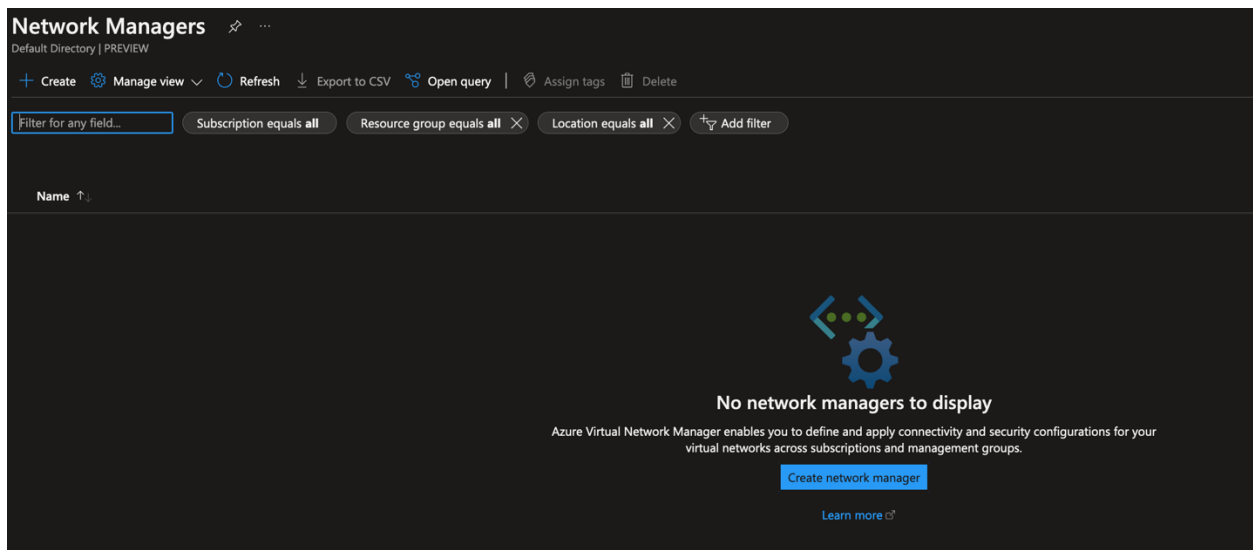
```
Name : vNETMgr-0-Prod
ResourceGroupName : NetworkMgrSubuDemo
Location : northcentralus
Id : /subscriptions/117f6288-70b4-4e64-af5f-04c2bf246f38/resourceGroups/NetworkMgrSubuDemo/providers/Microsoft.Network/virtualNetworks/vNETMgr-0-Prod
Etag : W/"150ff335-439b-45ad-a690-f6591794a4d6"
ResourceGuid : 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": [],
    "PrivateEndpointConnections": [],
    "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.



## Create a network manager

PREVIEW

Basics

Tags

Review + create

Create a network manager to centrally manage virtual networks at scale. [Learn more](#)

### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \*

Visual Studio Enterprise Subscription

Resource group \* ⓘ

NetworkMgrSubuDemo

[Create new](#)

### Instance details

Name \* ⓘ

subunetworkmgrdemo

Region \* ⓘ

(US) East US

Description

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 separate 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	subnetworkmgrdemo
Region	East US
Description	
Enabled features	Connectivity, Security admin
Scopes	Subscriptions: Visual Studio Enterprise Subscription

## Tags


None

## ✓ Your deployment is complete



Deployment name: ANM\_subnetworkmgrdemo\_1661332034452  
Subscription: [Visual Studio Enterprise Subscription](#)  
Resource group: [NetworkMgrSubuDemo](#)

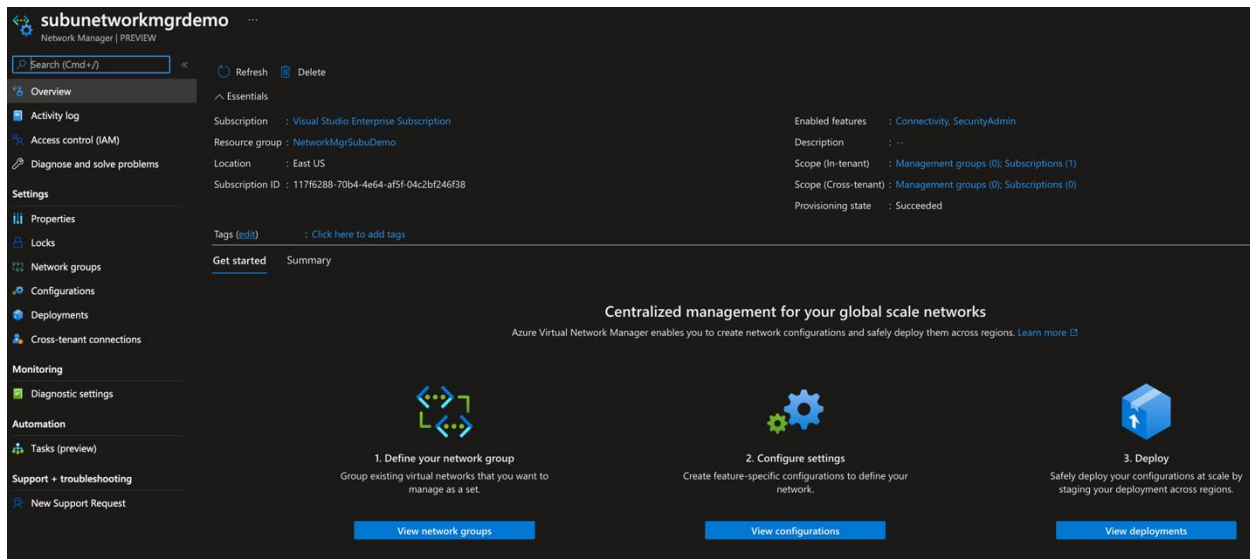
Start time: 8/24/2022, 1:07:19 PM

Correlation ID: 9cc534c6-4228-471f-9b5d-aaba3ce81331 

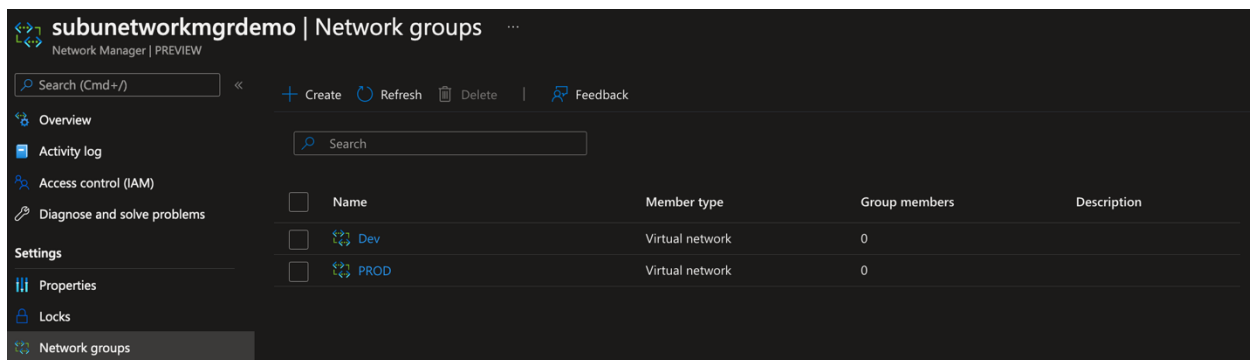
∨ **Deployment details**

∧ **Next steps**

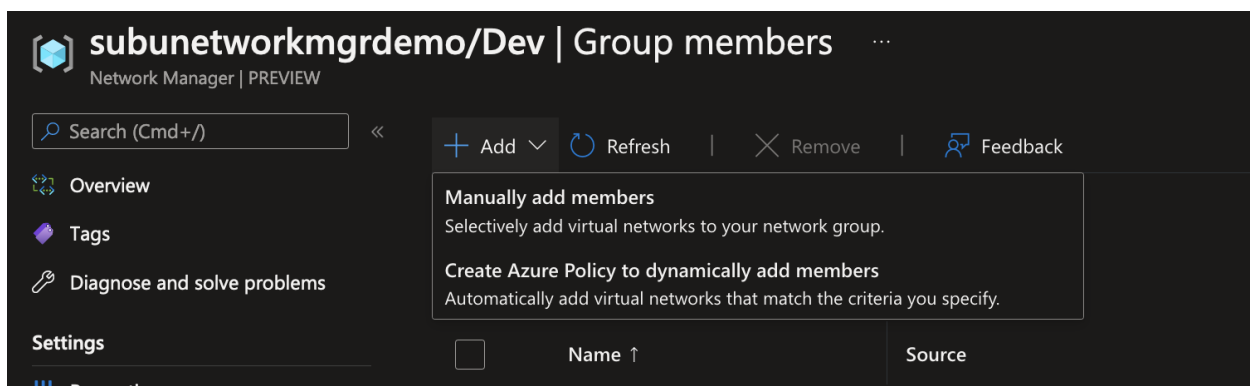
[Go to resource](#)



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.





## Manually add members

PREVIEW

Select virtual networks to add to your network group. [Learn more](#)

×

Tenant : Default Directory

No items selected No Grouping

<input type="checkbox"/>	Name	Subscription	Resource group	Location
<input type="checkbox"/>	↔ DevOps-POC-vnet	117f6288-70b4-4e64-af5f-04c2bf246f38	devops-poc	eastus
<input type="checkbox"/>	↔ vNETMgr-0-Dev	117f6288-70b4-4e64-af5f-04c2bf246f38	networkmgrsubudemo	northcentralus
<input type="checkbox"/>	↔ vNETMgr-1-Dev	117f6288-70b4-4e64-af5f-04c2bf246f38	networkmgrsubudemo	westus
<input type="checkbox"/>	↔ vNETMgr-2-Dev	117f6288-70b4-4e64-af5f-04c2bf246f38	networkmgrsubudemo	eastus
<input type="checkbox"/>	↔ vNETMgr-3-Dev	117f6288-70b4-4e64-af5f-04c2bf246f38	networkmgrsubudemo	eastus2
<input type="checkbox"/>	↔ vNETMgr-4-Dev	117f6288-70b4-4e64-af5f-04c2bf246f38	networkmgrsubudemo	westus2
<input type="checkbox"/>	↔ vNETMgr-5-Dev	117f6288-70b4-4e64-af5f-04c2bf246f38	networkmgrsubudemo	northeurope
<input type="checkbox"/>	↔ vNETMgr-6-Dev	117f6288-70b4-4e64-af5f-04c2bf246f38	networkmgrsubudemo	westeurope
<input type="checkbox"/>	↔ vNETMgr-7-Dev	117f6288-70b4-4e64-af5f-04c2bf246f38	networkmgrsubudemo	francecentral

Then we need to add a new Topology Connectivity.

## Add a connectivity configuration

PREVIEW

Basics **Topology** Review + create

Connectivity configurations define hub-and-spoke or mesh topologies applied to one or more network groups. [Learn more](#)

Name \*

Description

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.

Basics

Topology

Review + create

Topology \* ⓘ

☒ Mesh  
This will create peerings between virtual networks in the network groups you select.

☐ Hub and spoke  
This will create peerings between your specified hub and the virtual networks in the network groups you select.

☐ Enable mesh connectivity across regions ⓘ

Network groups

Virtual networks in the network groups you select will be used to create your desired topology.

+ Add

|

✕ Remove

☐

Name ↑

Group members ↑

No network groups to display

Add network groups to apply this configuration

Add

subnetworkmgrdemo | Configurations

Network Manager | PREVIEW

Search (Cmd+/)

Create

Refresh

Deploy

Delete

Feedback

Overview

Activity log

Access control (IAM)

Diagnose and solve problems

Settings

Properties

Locks

Network groups

Configurations

Search

No items selected

No Grouping

<input type="checkbox"/>	Name	Type	Associated network group
<input type="checkbox"/>	Democonfig	Connectivity - Mesh	1

This is based on the Hub and Spoke model setup.

## Add a connectivity configuration

PREVIEW

Basics **Topology** Review + create

Topology \* ⓘ

☐ Mesh  
This will create peerings between virtual networks in the network groups you select.

☒ Hub and spoke  
This will create peerings between your specified hub and the virtual networks in the network groups you select.

Hub \* ⓘ

↔ vNETMgr-0-Lab  
Select a hub

☐ Delete existing peerings ⓘ

**Spoke network groups**

All virtual networks in network groups that you add are peered to the hub. Direct connectivity creates additional peerings between virtual networks within the same network group and region. When direct connectivity is enabled, you can also enable global mesh to create peerings within the same group across all regions. If your hub has a gateway, spoke network groups can use the hub as gateway.

+ Add | × Remove

<input type="checkbox"/> Name ↑	<input type="checkbox"/> Direct connectivity ↑	<input type="checkbox"/> Hub as gateway ↑	<input type="checkbox"/> Global mesh ↑
<input type="checkbox"/> PROD	<input checked="" type="checkbox"/> Enable connectivity within network group	<input type="checkbox"/> Hub as gateway	<input checked="" type="checkbox"/> Enable mesh connectivity across regions

## subnetworkmgrdemo | Configurations

Network Manager | PREVIEW

Search (Cmd+/) « + Create ▾ Refresh | ⬆ Deploy | 🗑 Delete | 🗨 Feedback

Overview

Activity log Search

Access control (IAM) No items selected No Grouping ▾

Diagnose and solve problems

**Settings**

Properties

Locks

Network groups

Configurations

<input type="checkbox"/>	Name	Type	Associated network group
<input type="checkbox"/>	Democonfig	Connectivity - Mesh	1
<input type="checkbox"/>	subudemohubmodel	Connectivity - Hub and spoke	1

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

Home > ANM\_subnetworkmgrdemo\_1661332034452 | Overview > subnetworkmgrdemo | Configurations > Add a security configuration >

Add a rule collection

PREVIEW

A rule collection is a group of security admin rules applied to the same target network groups. All security admin rules in this rule collection will be applied to the target network groups. [Learn more](#)

Name \*

Target network groups \*

Security admin rules

Traffic will be allowed or denied depending on the action you set for the highest priority match rule. [Learn more](#)

+ Add

ⓘ Delete

<input type="checkbox"/>	Name	Prio... ↑	Direction	Protocol	Source	Source port
No rules to display						
Add rules to your rule collection						
<div>Add</div>						

Add

Cancel

Add a rule

PREVIEW

Traffic will be allowed or denied depending on the action you set for the highest priority match rule. [Learn more](#)

Name \*

Description

Priority \*

Action \*

Direction \*

Protocol \*

Source

Source type ⓘ

Source IP addresses ⓘ

Source port ⓘ

Destination

Destination type ⓘ

Destination IP addresses ⓘ

Add

Cancel

subnetworkmgrdemo | Configurations

Network Manager | PREVIEW

Search (Cmd+⌘)

«

+ Create

↺ Refresh

⬆ Deploy

🗑 Delete

🗨 Feedback

Overview

Activity log

Access control (IAM)

Diagnose and solve problems

Settings

Properties

Locks

Network groups

Configurations

Search

No items selected

No Grouping

<input type="checkbox"/>	Name	Type	Associated network group
<input type="checkbox"/>	Democonfig	Connectivity - Mesh	1
<input type="checkbox"/>	subudemohubmodel	Connectivity - Hub and spoke	1
<input type="checkbox"/>	subudemosec	Security - Admin rules	1

We can see all the Summary in 1 page.

Get started

Summary

Scopes

Name	Type
117f6288-70b4-4e64-af5f-04c2bf246f38	Subscription

Network groups

Name	Member type
Dev	Virtual network
PROD	Virtual network

Configurations

Name	Type
Democonfig	Connectivity - Mesh
subudemohub...	Connectivity - Hub and spoke
subudemosec	Security - Admin rules

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.

Deploy a configuration

PREVIEW

Goal state

Review + deploy

The configurations you deploy represent your overall desired state for virtual networks in your target regions. Azure virtual network manager makes necessary changes to achieve this goal state. [Learn more](#)

### Configurations

Your goal state can include one or more configuration types. For each type you include, the configurations you select will overwrite any existing configurations of the same type. You can remove existing configurations from target regions by deploying a goal state without any configurations selected. [Learn more](#)

☒ Include security admin in your goal state

Security admin configurations

subudemosec

This configuration deployment option will apply all security admin rules EXCEPT 'Deny' rules to the target virtual networks that have service using network intent policies. [Learn more](#)

☒ Include connectivity configurations in your goal state

Connectivity configurations

subudemohubmodel

### Regions

Your goal state will be deployed to the regions you select. [Learn more](#)

Target regions \* 

East US

Previous

Next

Filter items...

☐ Select all

☐ Australia Central

☐ Australia Central 2

☐ Australia East

☐ Australia Southeast

☐ Brazil South

☐ Brazil Southeast

☐ Canada Central

☐ Canada East

☐ Central India

☐ Central US

☐ East Asia

☒ East US

☐ East US 2

☐ France Central

☐ France South

☐ Germany North

☐ Germany West Central

☐ Japan East

☐ Japan West

☐ Korea Central

☐ Korea South

☐ North Central US

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

Deploy a configuration

PREVIEW

Goal state

Review + deploy

The changes to be made in each of your target regions are listed below. You can go back to make any changes if needed. When you deploy this goal state, existing configurations in each region will be overwritten. [Learn more](#)

### East US

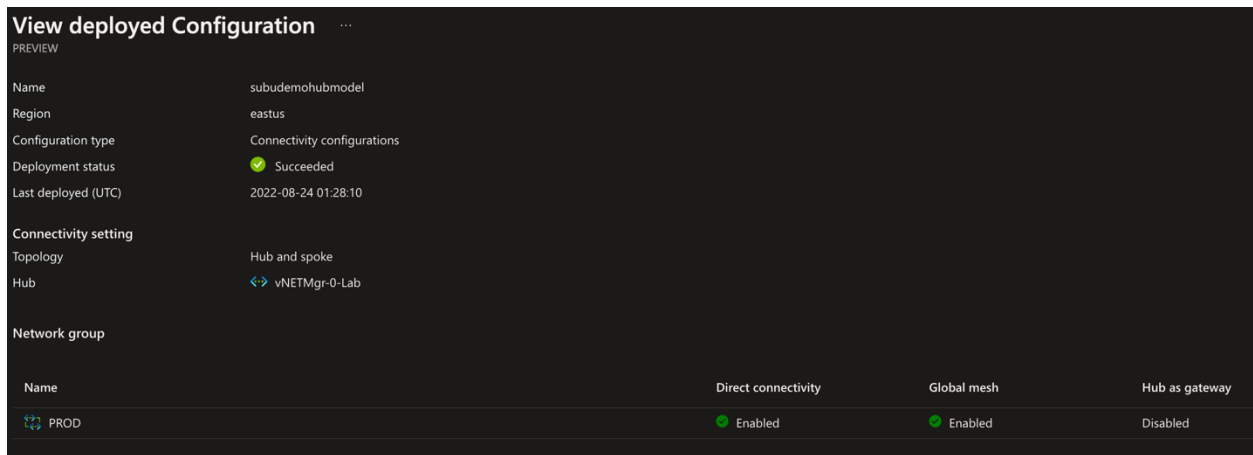
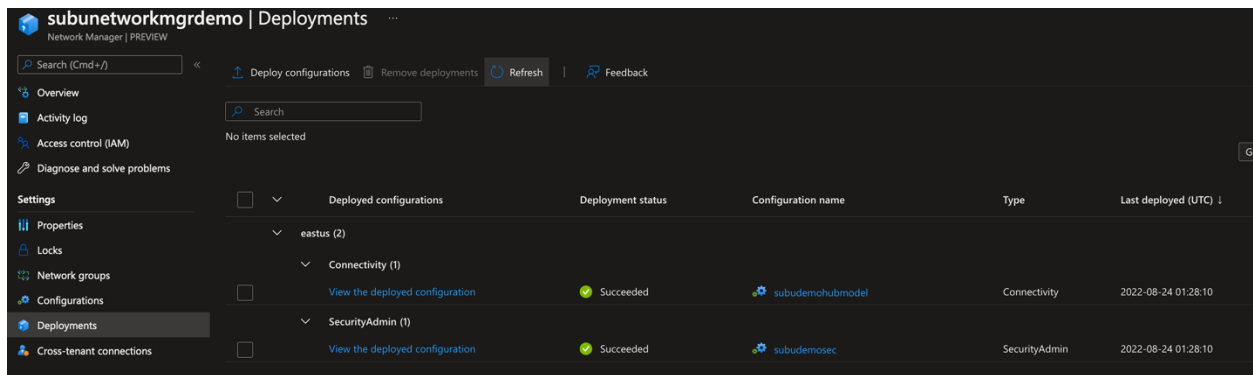
Existing configurations

Name	Type	Last deployed (UTC)
------	------	---------------------

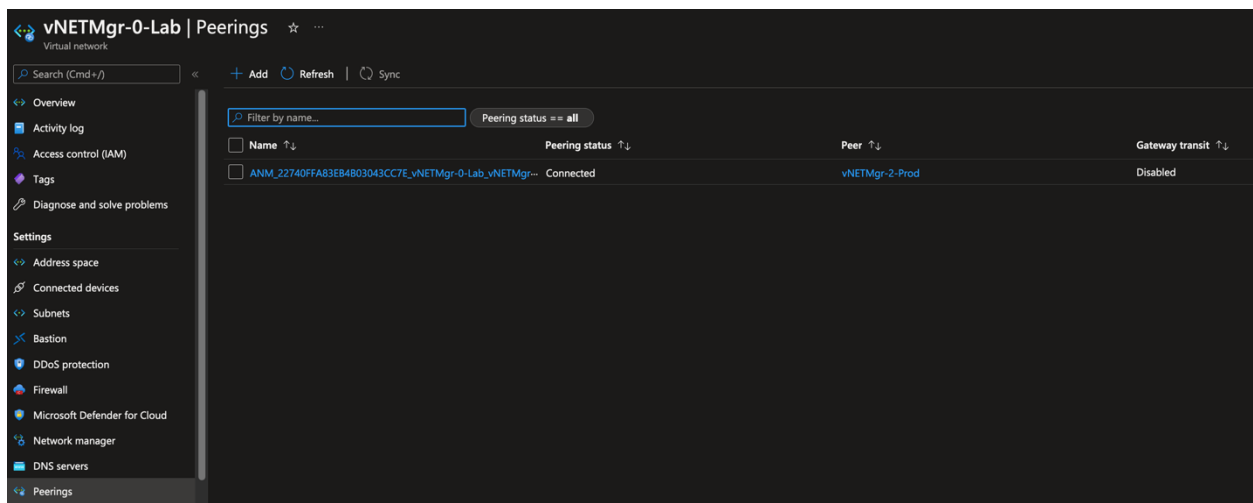
Goal state

Name	Type	Last modified (UTC)	
subudemohubmodel	Connectivity	2022-08-24T09:19:17.7031329Z	<input checked="" type="checkbox"/> Add
subudemosec	SecurityAdmin	2022-08-24T09:23:09.1883474Z	<input checked="" type="checkbox"/> Add

14



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 .