

# Multi-Region NSG Association Script - Usage Guide

## Overview

This PowerShell script automatically associates subnets with region-specific NSGs across multiple Azure regions. It intelligently skips subnets that already have NSGs associated and excludes system subnets.

## Configuration

### 1. Update NSG Variables

Edit the `$RegionalNSGs` hashtable in the script to match your environment:

powershell

```
$RegionalNSGs = @{
    "East US" = @{
        ResourceGroupName = "rg-security-eastus"
        NSGName = "nsg-eastus-default"
    }
    "West US" = @{
        ResourceGroupName = "rg-security-westus"
        NSGName = "nsg-westus-default"
    }
    "North Europe" = @{
        ResourceGroupName = "rg-security-northeurope"
        NSGName = "nsg-northeurope-default"
    }
    "Southeast Asia" = @{
        ResourceGroupName = "rg-security-southeastasia"
        NSGName = "nsg-southeastasia-default"
    }
}
```

### 2. Customize Excluded Subnets (Optional)

By default, these system subnets are excluded:

- `GatewaySubnet`
- `AzureFirewallSubnet`
- `AzureBastionSubnet`
- `RouteServerSubnet`

You can add more exclusions using the `-ExcludeSubnets` parameter.

## Usage Examples

### 1. Basic Execution

```
powershell

# Connect to Azure first
Connect-AzAccount

# Run the script
.\Associate-RegionalNSGs.ps1 -SubscriptionId "12345678-1234-1234-1234-123456789012"
```

### 2. Preview Mode (Recommended First Run)

```
powershell

# See what would be changed without making actual changes
.\Associate-RegionalNSGs.ps1 -SubscriptionId "12345678-1234-1234-1234-123456789012" -WhatIf
```

### 3. Force Mode (No Confirmation Prompts)

```
powershell

# Skip all confirmation prompts
.\Associate-RegionalNSGs.ps1 -SubscriptionId "12345678-1234-1234-1234-123456789012" -Force
```

### 4. Custom Excluded Subnets

```
powershell

# Exclude additional subnets
.\Associate-RegionalNSGs.ps1 -SubscriptionId "12345678-1234-1234-1234-123456789012" -ExcludeSubnets @("GatewaySubnet", "AzureFirewallSubnet", "AzureBastionSubnet", "Microsoft.ManagedNetwork")
```

### 5. Complete Example with All Options

```
powershell

.\Associate-RegionalNSGs.ps1 `
    -SubscriptionId "12345678-1234-1234-1234-123456789012" `
    -ExcludeSubnets @("GatewaySubnet", "AzureFirewallSubnet", "AzureBastionSubnet", "Microsoft.ManagedNetwork") `
    -Force
```

## Sample Output

=== Multi-Region NSG Association Script ===

Started at: 12/10/2024 2:30:15 PM

Configured NSGs for regions:

East US -> nsg-eastus-default

West US -> nsg-westus-default

North Europe -> nsg-northeurope-default

Southeast Asia -> nsg-southeastasia-default

Connected to Azure as: admin@company.com

Subscription context set to: Production Subscription

Validating NSGs...

✓ NSG 'nsg-eastus-default' found in region 'East US'

✓ NSG 'nsg-westus-default' found in region 'West US'

✓ NSG 'nsg-northeurope-default' found in region 'North Europe'

✓ NSG 'nsg-southeastasia-default' found in region 'Southeast Asia'

Discovering virtual networks in subscription...

Found 8 virtual networks

=== Processing Region: East US ===

VNets in region: 3

Using NSG: nsg-eastus-default

--- Processing VNet: vnet-prod-eastus (RG: rg-network-eastus) ---

Processing subnet: subnet-web (10.0.1.0/24)

✓ Successfully associated subnet 'subnet-web' with NSG 'nsg-eastus-default'

Processing subnet: subnet-app (10.0.2.0/24)

◦ Subnet already associated with target NSG 'nsg-eastus-default'

Processing subnet: GatewaySubnet (10.0.255.0/27)

◦ Excluded subnet (system/reserved subnet)

=== SUMMARY ===

Total VNets processed: 8

Total subnets found: 24

Subnets processed: 18

Successful associations: 15

Failed associations: 0

Already associated: 3

Excluded subnets: 6

Skipped subnets: 0

=== REGIONAL BREAKDOWN ===

East US:

VNets: 3 | Subnets: 9

Processed: 6 | Success: 5 | Failed: 0 | Skipped: 1

West US:

VNets: 2 | Subnets: 6

Processed: 4 | Success: 4 | Failed: 0 | Skipped: 0

...

Results exported to: nsg-association-results-20241210-143045.csv

## Script Logic Flow

### 1. Validation Phase

- Check Azure connection
- Set subscription context
- Validate all configured NSGs exist

### 2. Discovery Phase

- Get all VNets in subscription
- Group VNets by region

### 3. Processing Phase (Per Region)

- Get regional NSG
- Iterate through VNets in region
- Process each subnet:
  - Skip if excluded
  - Skip if already has target NSG
  - Associate with regional NSG

### 4. Reporting Phase

- Display summary statistics
- Export results to CSV
- Show regional breakdown

## Safety Features

### Built-in Protections

- **WhatIf Mode:** Preview changes without applying
- **Validation:** Checks all resources exist before processing

- **Smart Skipping:** Avoids unnecessary changes
- **Exclusion Lists:** Protects system subnets
- **Confirmation Prompts:** Ask before each change (unless -Force)

## ✓ Error Handling

- Graceful failure handling per subnet
- Continues processing even if some operations fail
- Detailed error reporting
- Rollback-friendly (can be re-run safely)

## 📈 Advanced Scenarios

### 1. Multi-Subscription Processing

powershell

```
$subscriptions = @("sub1-id", "sub2-id", "sub3-id")

foreach ($subId in $subscriptions) {
    Write-Host "Processing subscription: $subId" -ForegroundColor Cyan
    .\Associate-RegionalNSGs.ps1 -SubscriptionId $subId -Force
}
```

### 2. Integration with Azure DevOps

yaml

```
# Azure Pipeline task
- task: AzurePowerShell@5
  displayName: 'Associate Regional NSGs'
  inputs:
    azureSubscription: $(azureSubscription)
    ScriptType: 'FilePath'
    ScriptPath: 'scripts/Associate-RegionalNSGs.ps1'
    ScriptArguments: '-SubscriptionId $(subscriptionId) -Force'
    azurePowerShellVersion: 'LatestVersion'
```

### 3. Scheduled Automation

```
powershell
```

```
# Azure Automation Runbook
```

```
param(  
    [string]$SubscriptionId  
)
```

```
# Import required modules
```

```
Import-Module Az.Network
```

```
# Connect using managed identity
```

```
Connect-AzAccount -Identity
```

```
# Run the association script
```

```
.\Associate-RegionalNSGs.ps1 -SubscriptionId $SubscriptionId -Force
```

## Troubleshooting

### Common Issues

#### 1. NSG Not Found

```
x NSG 'nsg-eastus-default' not found in region 'East US'
```

**Solution:** Verify NSG names and resource groups in the script variables

#### 2. Access Denied

```
Failed to get NSG: The client does not have authorization
```

**Solution:** Ensure you have Network Contributor role or sufficient permissions

#### 3. Region Mismatch

```
No NSG configured for region 'Canada Central'
```

**Solution:** Add the missing region to the `$RegionalNSGs` hashtable

### Debug Commands

powershell

*# Check available regions with VNets*

```
Get-AzVirtualNetwork | Group-Object Location | Select-Object Name, Count
```

*# Verify NSG exists*

```
Get-AzNetworkSecurityGroup -ResourceGroupName "rg-security-eastus" -Name "nsg-eastus-de
```

*# Check current subnet associations*

```
$vnet = Get-AzVirtualNetwork -ResourceGroupName "rg-network" -Name "vnet-prod"
```

```
$vnet.Subnets | Select-Object Name, @{N='NSG';E={$_.NetworkSecurityGroup.Id -split '/'
```



## Best Practices

1. **Always run with -WhatIf first** to preview changes
2. **Use consistent naming conventions** for NSGs across regions
3. **Test in non-production** environments first
4. **Review exclusion lists** for your specific environment
5. **Monitor results** using the exported CSV files
6. **Document regional NSG configurations** for compliance

This script provides a robust, enterprise-ready solution for managing NSG associations across multiple Azure regions! 🚀