Case Study

Requirements

- A low cost solution based on demand of dynamic business conditions.
- As the business expands across EastUS and EA, they would like to have their DataCenter virtualised using cloud computing.
- Critical Data should be made available in case of disaster

Business Requirements

East Asia

- 2 web servers with 99.95% high availability
- These web services has to be utilised with proper balance with client affinity with Public IP
- Selected web servers should be reachable via RDP from internet
- A jump port should accessible from internet to upload contents to web servers.
- Protect web server traffic restricted to allowed based on ip addresses which will be updated as warranted
- Enable backup for WebServers
- Have alert generated in case of 80% above cpu usage

East US

- EastUS server (Server11) should be accessible from internet via public IP
- Establish secure Connection to SEA-EUS Azure sites
- All servers should be reachable with internal ip addresses

Storage Requirements

- EUS based resources should provide data resiliency in case of azure datacentre failure.
- The storage should be accessible by applications with secure access. provide access urls and keys.
- Sales manager should access his resource from windows explorer.
- SEA data resources must provide high resiliency in case of even multiple azure data center failures

Azure Resource management

- Create Vmadmin user who can manage all VM in the subscription
- Create Backup_admin user who can manage backup only in EUS servers in EURG

Implementation Flow

- Created 2 web servers along with resource group, vnet, subnet, load balancer, back end pool using terraform script in East Asia

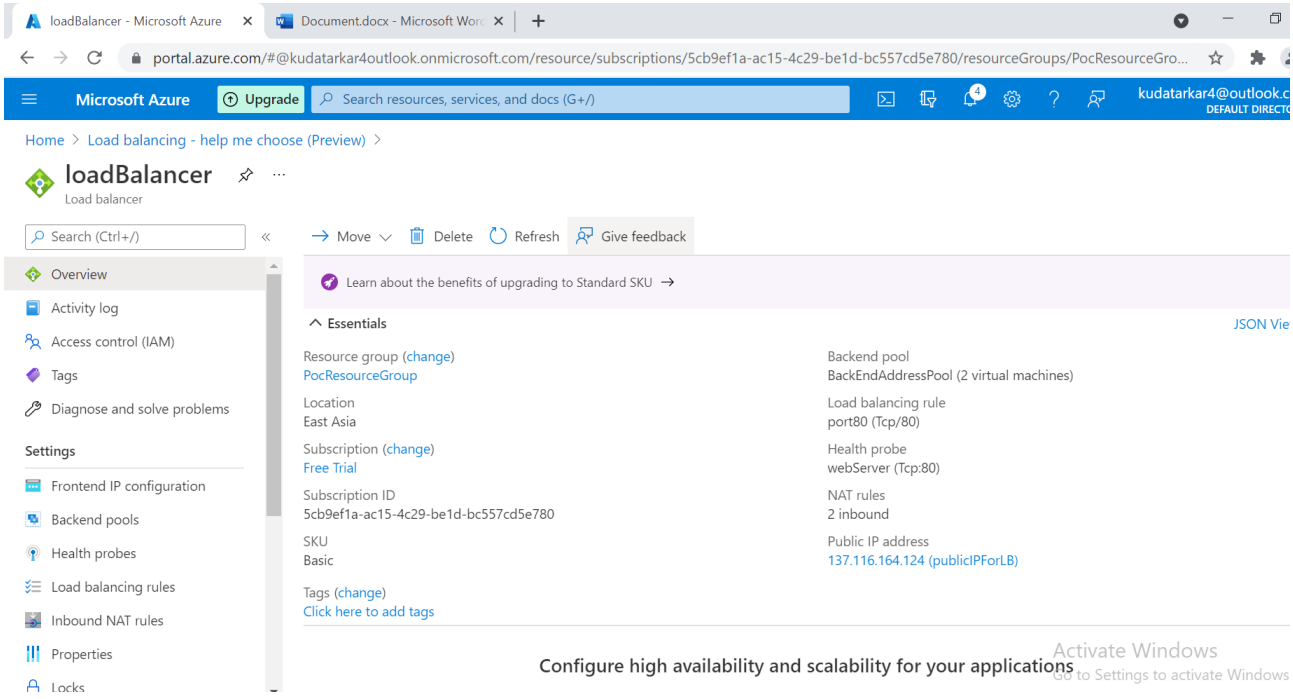
main.tf scriptVM.ps1

D: > POC > main.tf

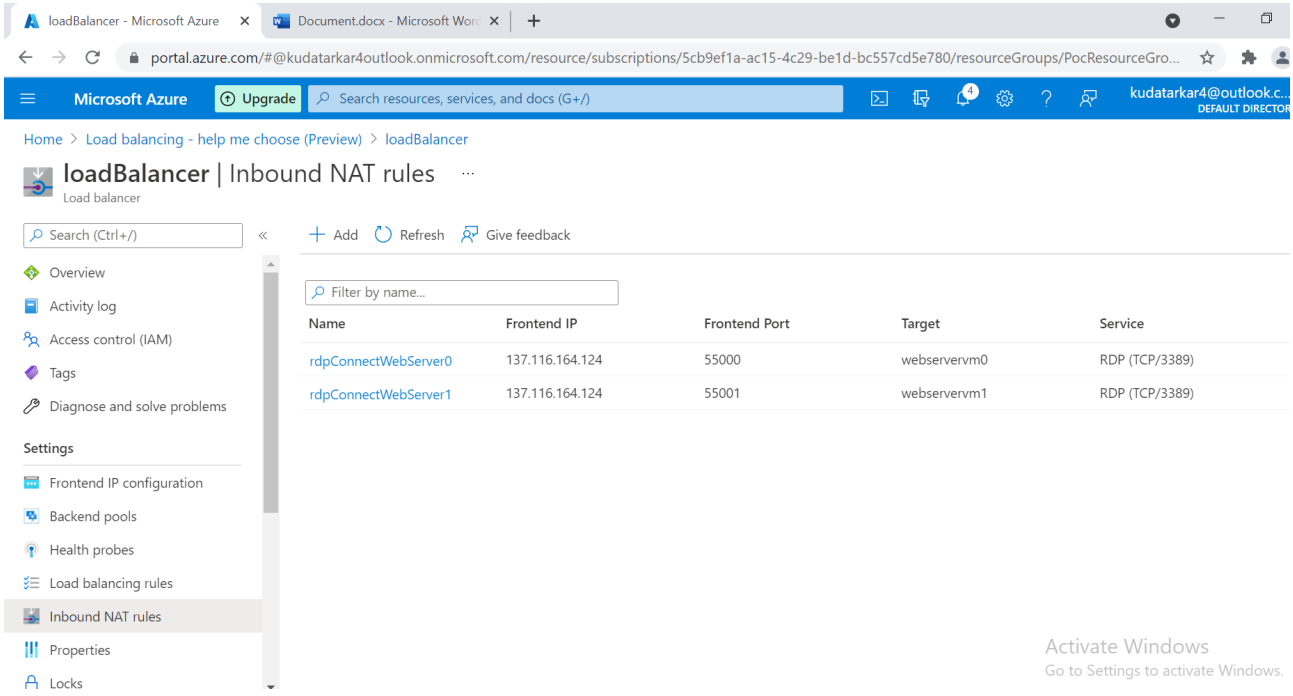
```
provider "azurerm" {
  features {}
}

resource "azurerm_resource_group" "varpoc" {
 name     = "PocResourceGroup"
 location = "East Asia"
}

resource "azurerm_virtual_network" "varpoc" {
 name                = "webservervn"
 address_space       = ["10.0.0.0/16"]
 location            = azurerm_resource_group.varpoc.location
 resource_group_name = azurerm_resource_group.varpoc.name
}

resource "azurerm_subnet" "varpoc" {
 name                 = "webserversub"
 resource_group_name  = azurerm_resource_group.varpoc.name
 virtual_network_name = azurerm_virtual_network.varpoc.name
 address_prefix       = "10.0.2.0/24"
}

resource "azurerm_public_ip" "varpoc" {
 name                         = "publicIPForLB"
 location                     = azurerm_resource_group.varpoc.location
 resource_group_name          = azurerm_resource_group.varpoc.name
 allocation_method            = "Static"
}

```

*Complete script is uploaded on GitHub

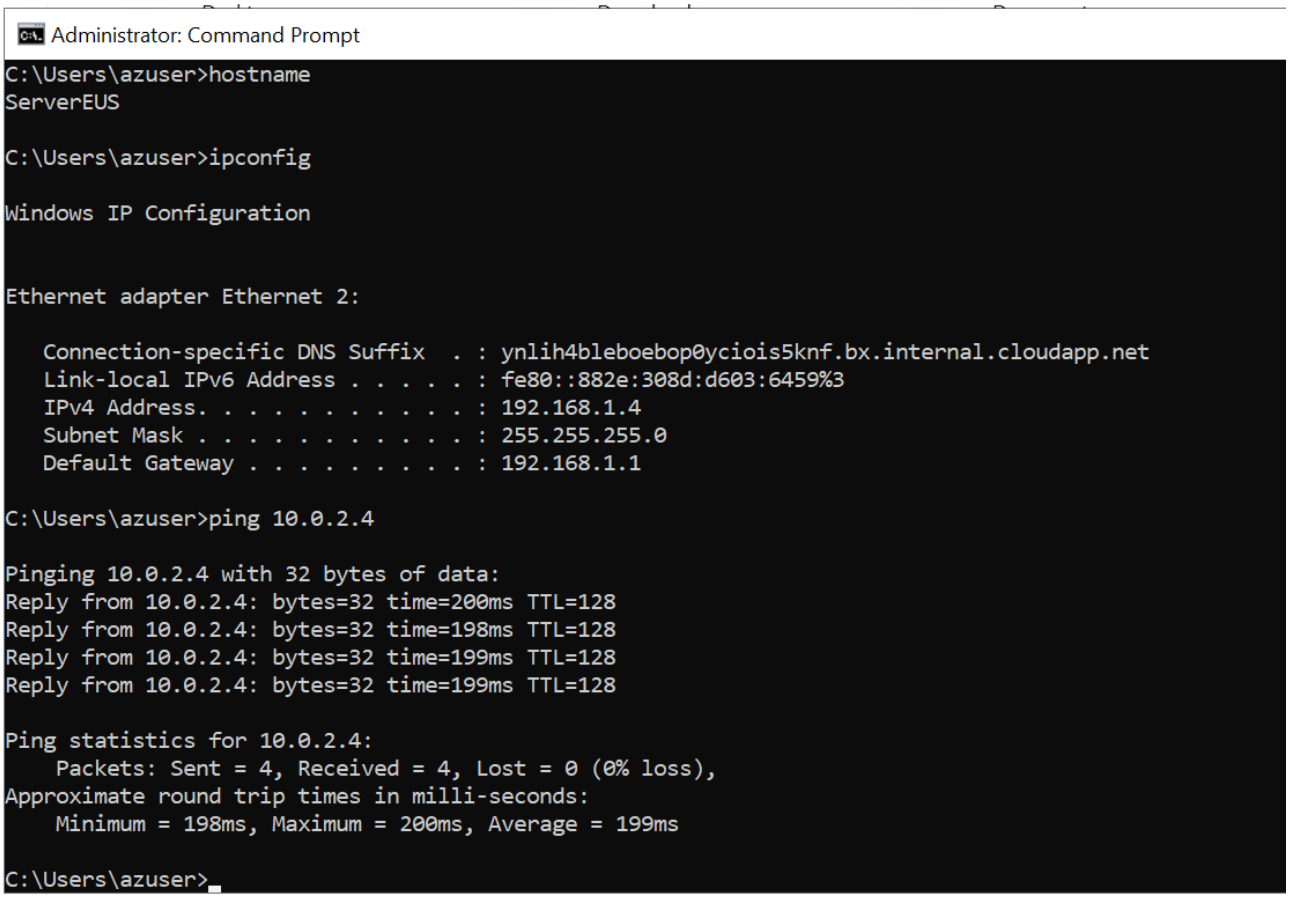
- Configured load balancer



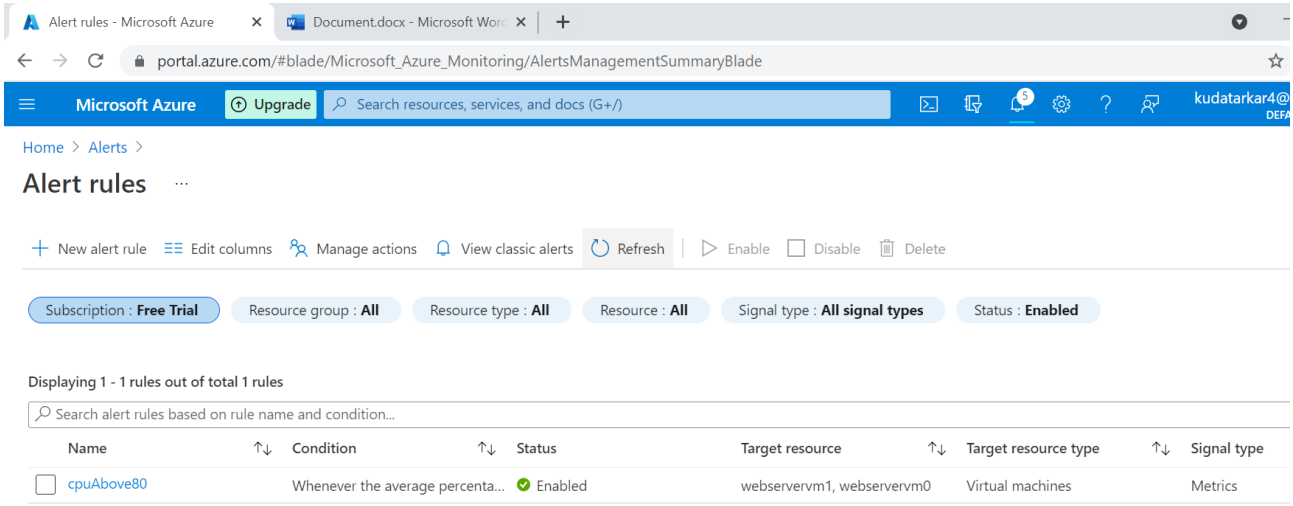
- Created NAT rule to access server via RDP



- Private ip connection testing



- Created alert in case of CPU usages are above 80%



- Created 1 VM in East US along with all networking components using PowerShell script

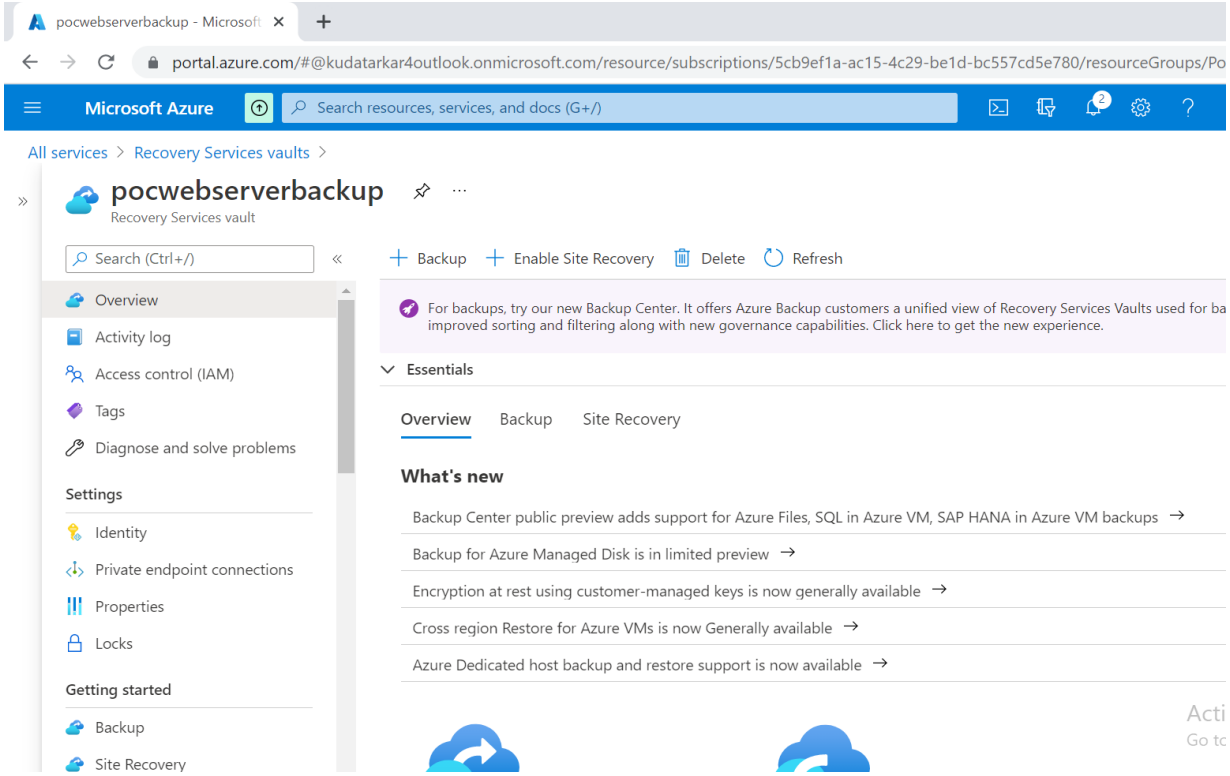
main.tf | scriptVM.ps1

D: > POC > scriptVM.ps1 > ...

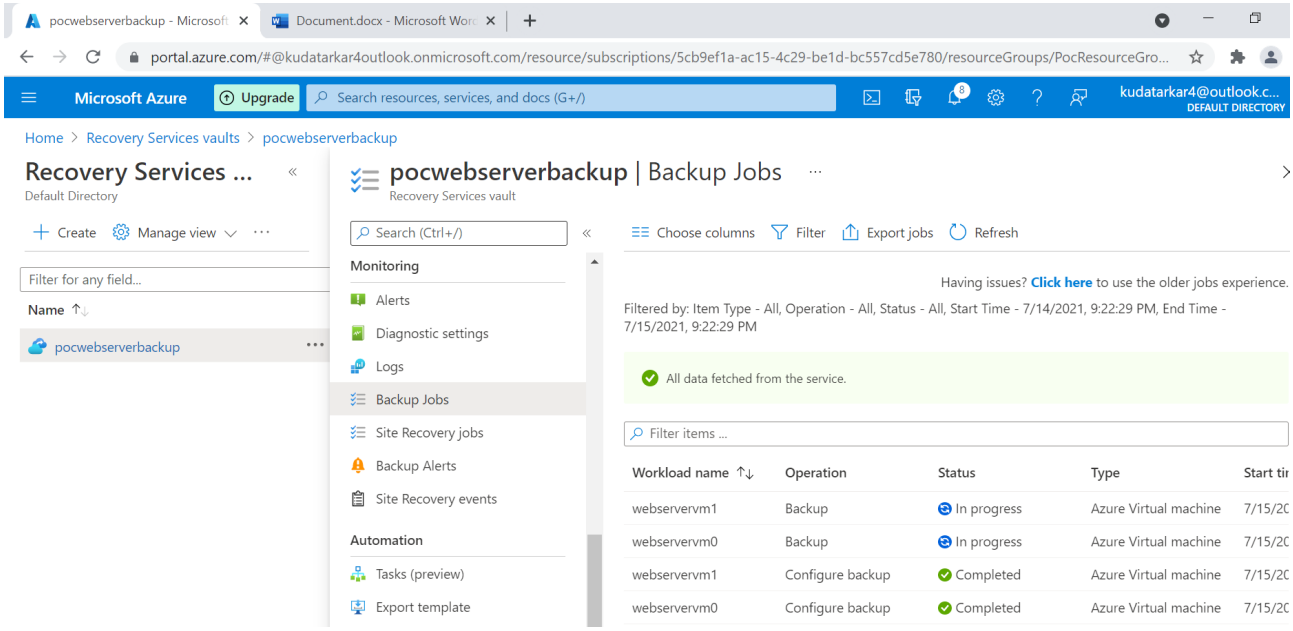
```powershell
# Variables
$resourceGroup = "PocResourceGroup"
$location = "East US"
$vmName = "ServerEUS"

# user
$cred = Get-Credential -Message " username and password "

# Create a subnet
$subnetConfig = New-AzVirtualNetworkSubnetConfig -Name subnet1 -AddressPrefix 192.168.1.0/24

# Create a vn
$vnet = New-AzVirtualNetwork -ResourceGroupName $resourceGroup -Location $location `
  -Name vneteus -AddressPrefix 192.168.0.0/16 -Subnet $subnetConfig

# Create a dns
$pip = New-AzPublicIpAddress -ResourceGroupName $resourceGroup -Location $location `
  -Name "publicdns$(Get-Random)" -AllocationMethod Static -IdleTimeoutInMinutes 4

# rdp
$nsgRuleRDP = New-AzNetworkSecurityRuleConfig -Name RuleRDP  -Protocol Tcp `
  -Direction Inbound -Priority 1000 -SourceAddressPrefix * -SourcePortRange * -DestinationAddressPrefix * `
  -DestinationPortRange 3389 -Access Allow

# Create nsg
$nsg = New-AzNetworkSecurityGroup -ResourceGroupName $resourceGroup -Location $location `
  -Name nsgUS -SecurityRules $nsgRuleRDP

# Create nic
$nic = New-AzNetworkInterface -Name nicUS -ResourceGroupName $resourceGroup -Location $location `
```

*Complete script is uploaded on GitHub

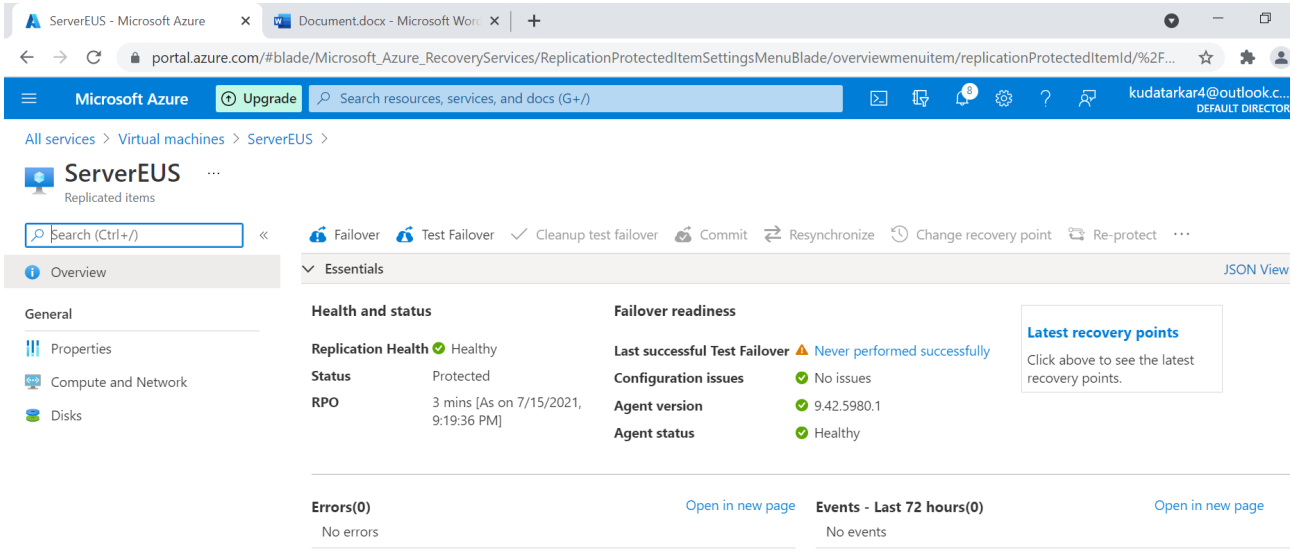
- Created Recovery Services vault for backups and disaster recovery



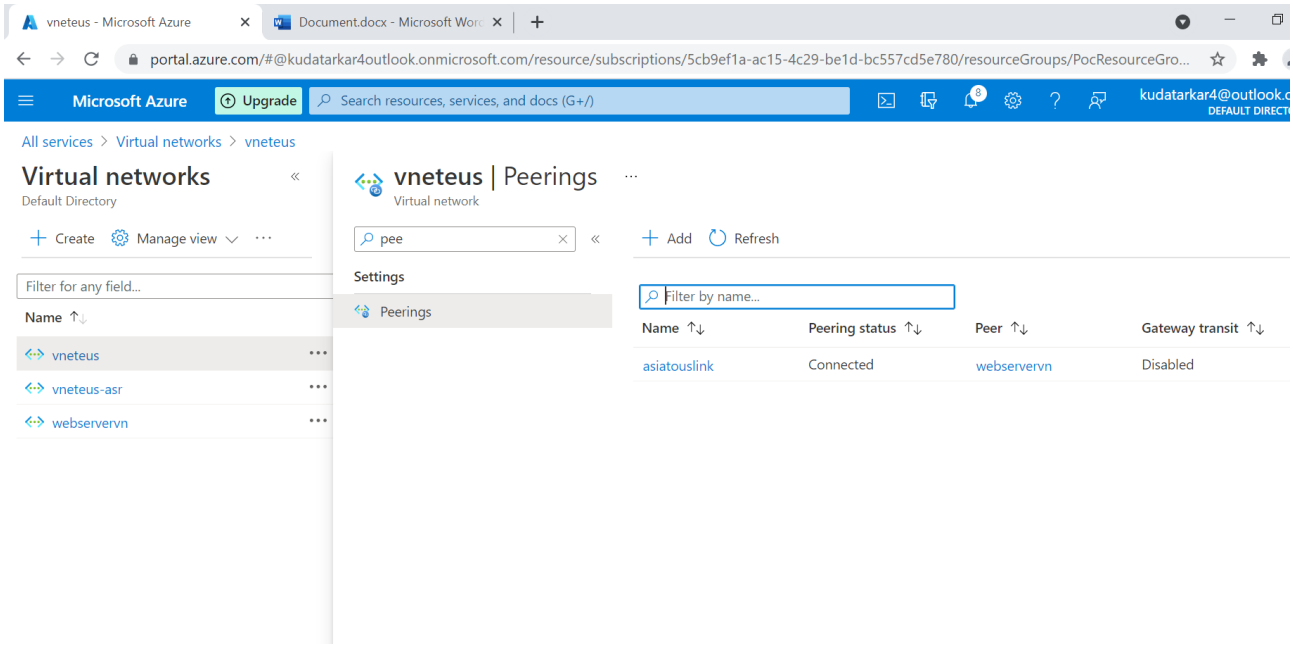
- Created backup jobs for 2 webserver backups



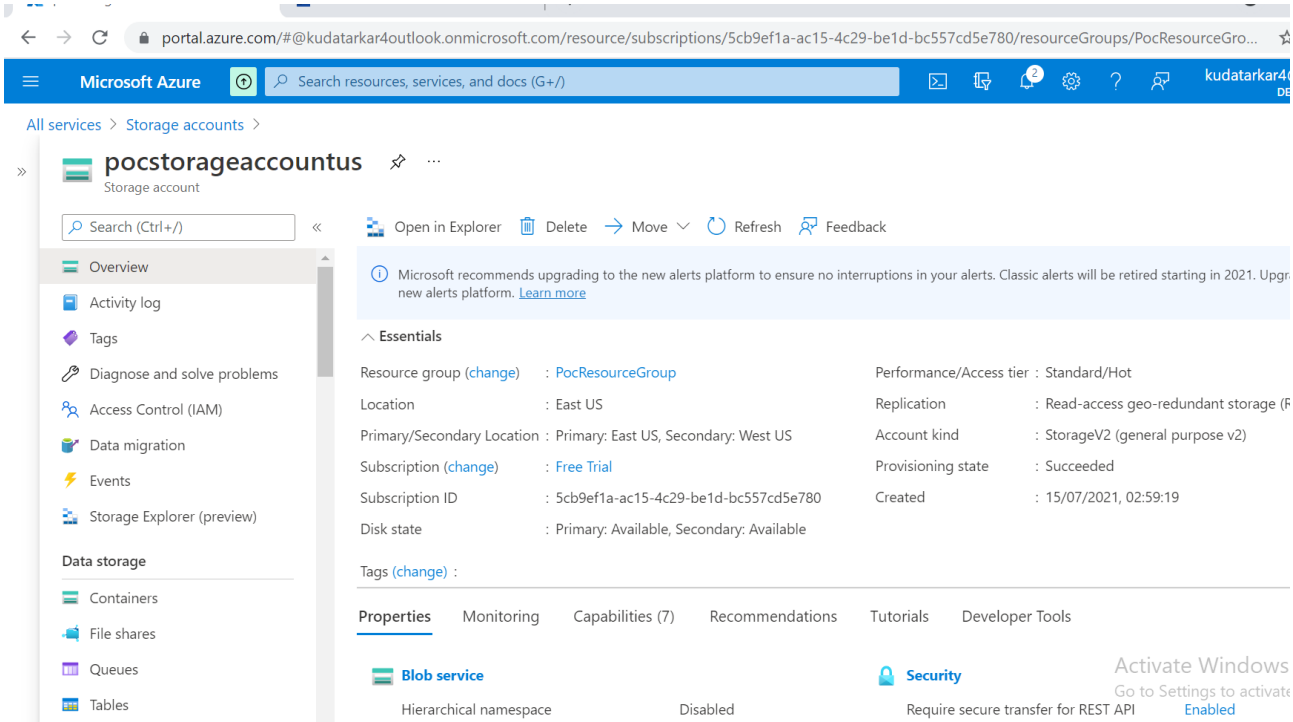
- Created Disaster recovery service in West US region for East US server and East Asia to SEA



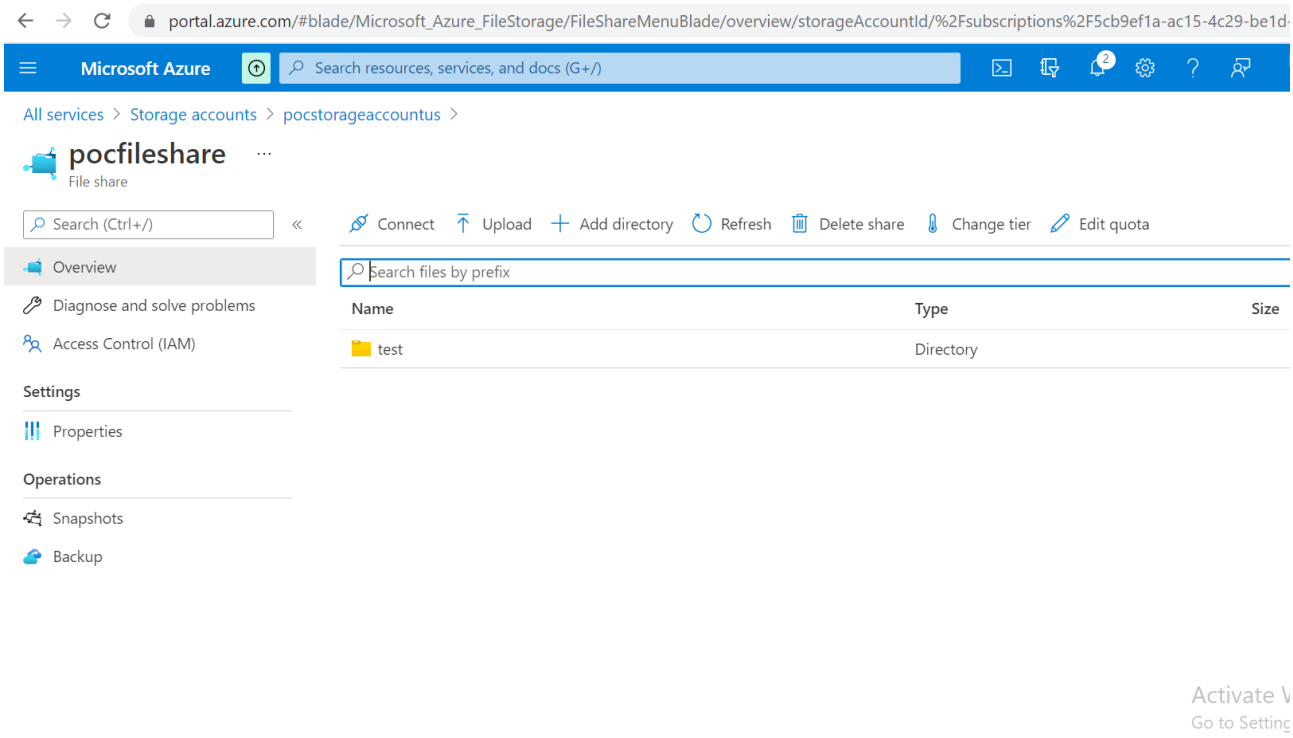
- Created peering secure Connection to EA-EUS so that All servers will be reachable with internal ip addresses



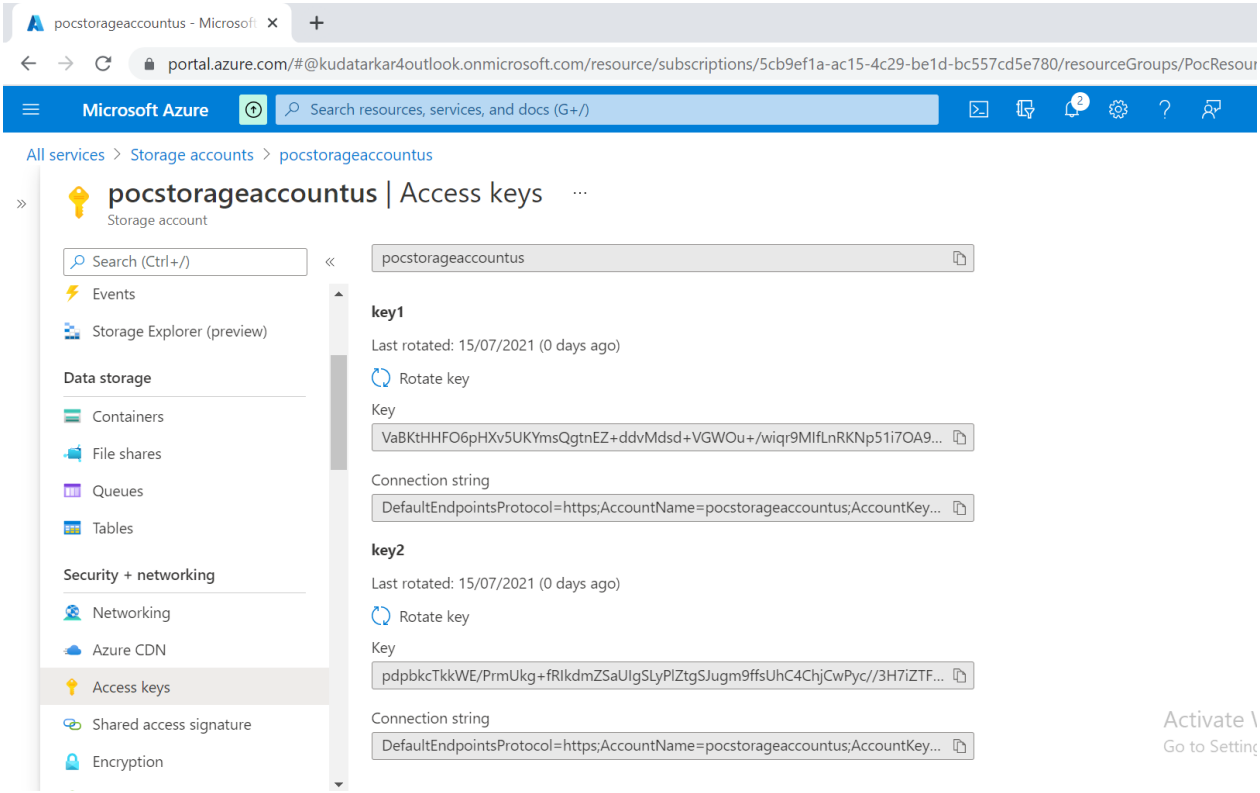
- Created storage account for storage requirements



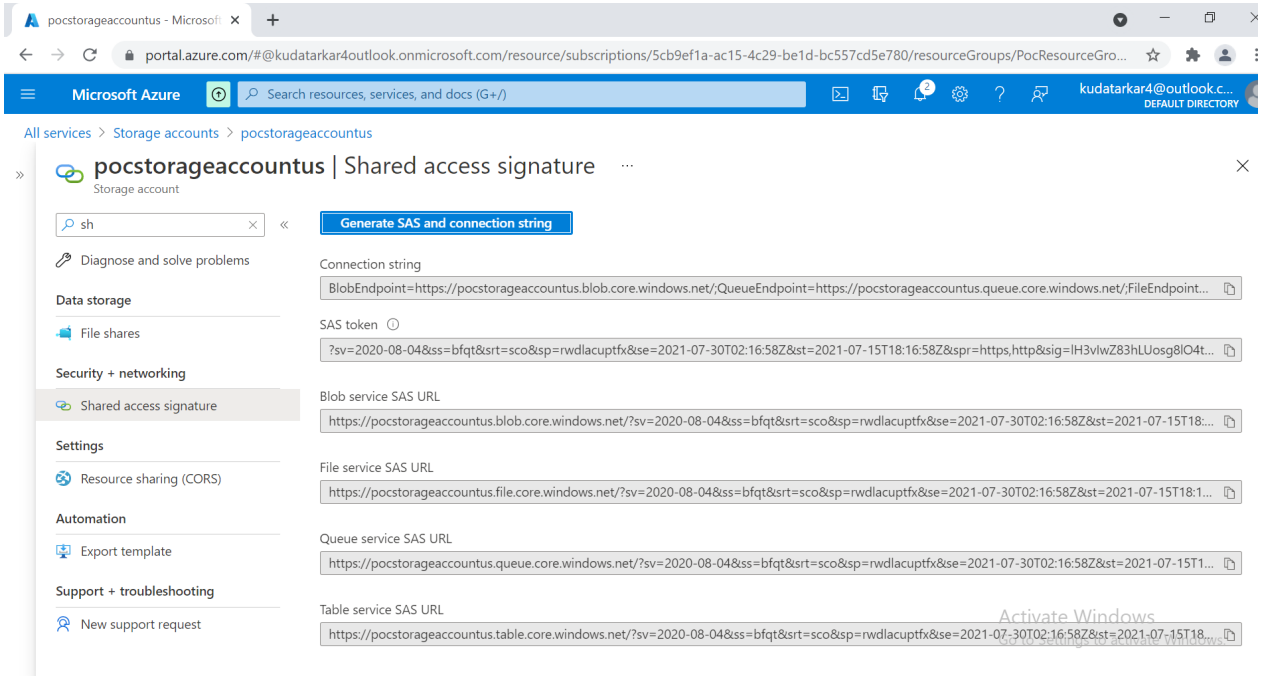
- Created File shares to access resources from windows explorer and can be used to store/share data across virtual server or physical machine



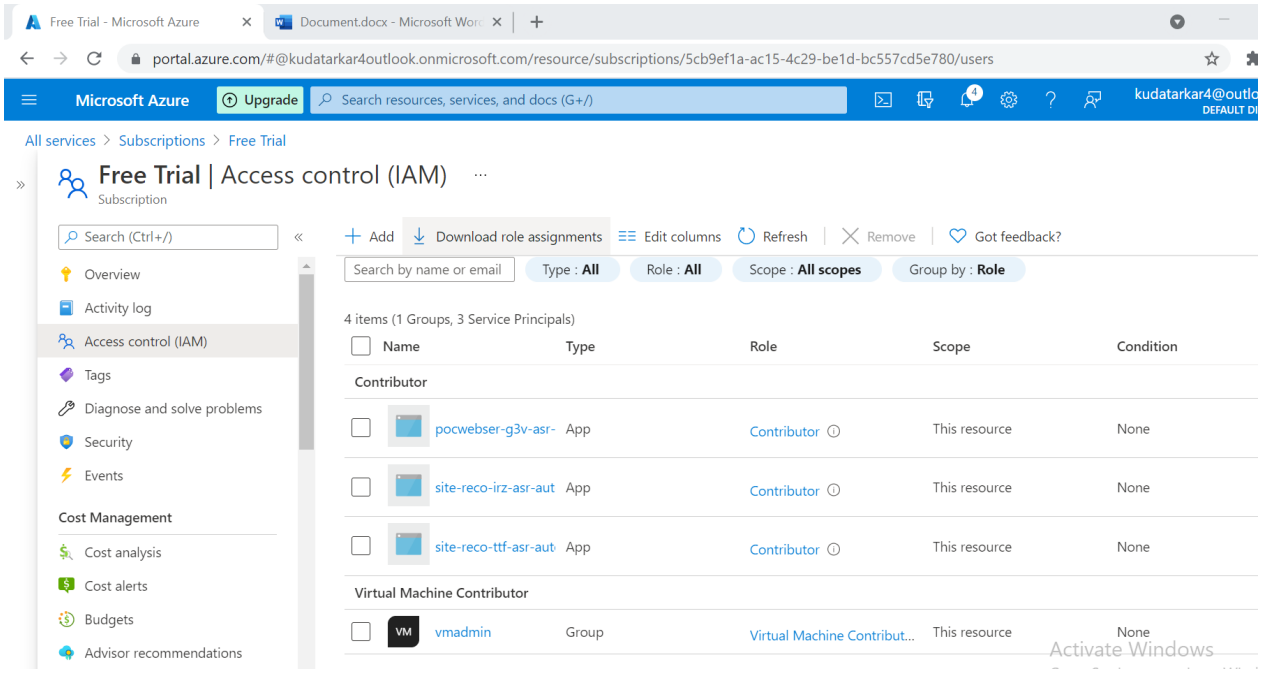
- Below are access key and connection string which will be used by applications



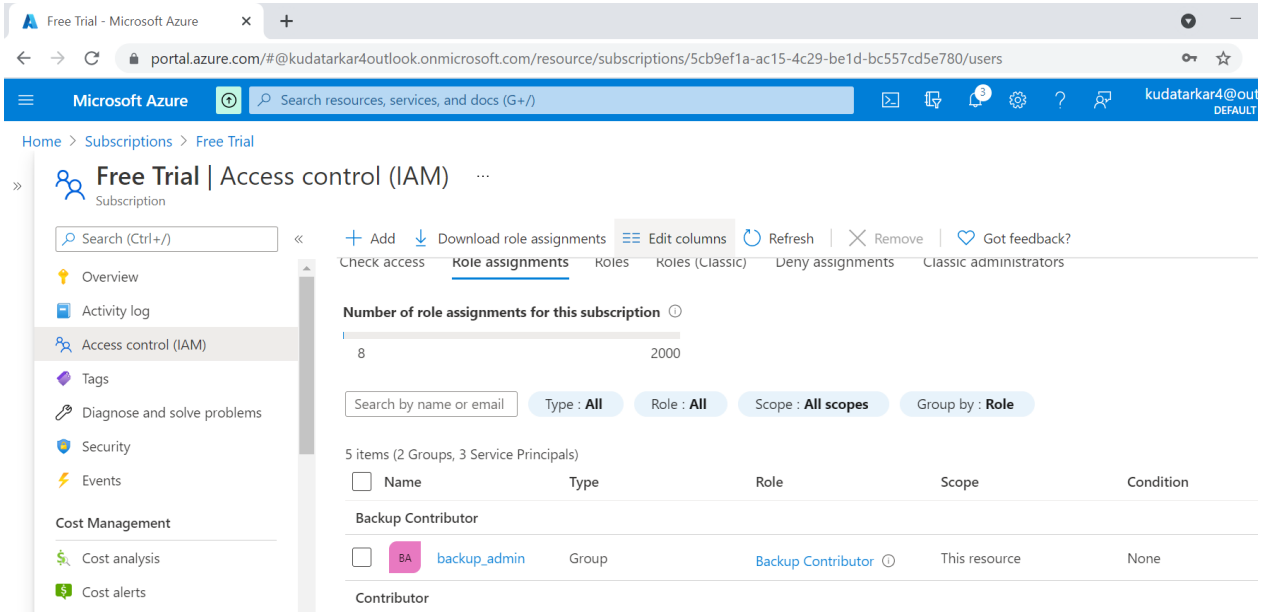
- Shared access keys can be used for limited access



- Created Vmadmin user who can manage all VM in the subscription

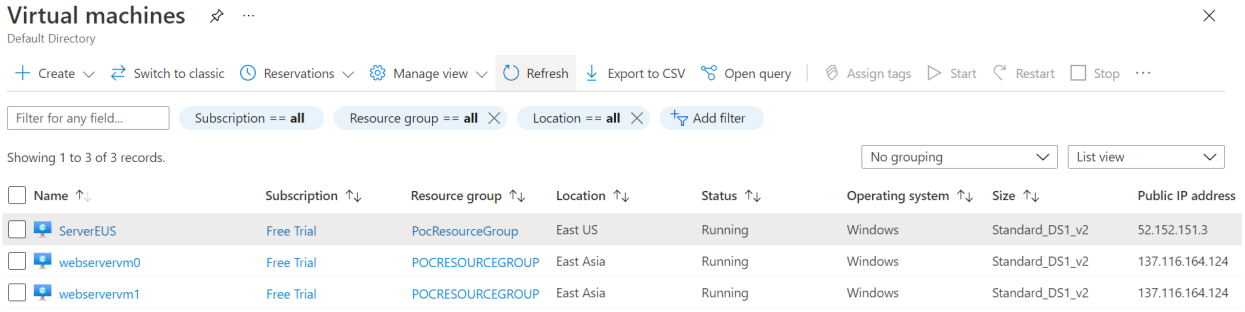


- Created backup admin to manage backups

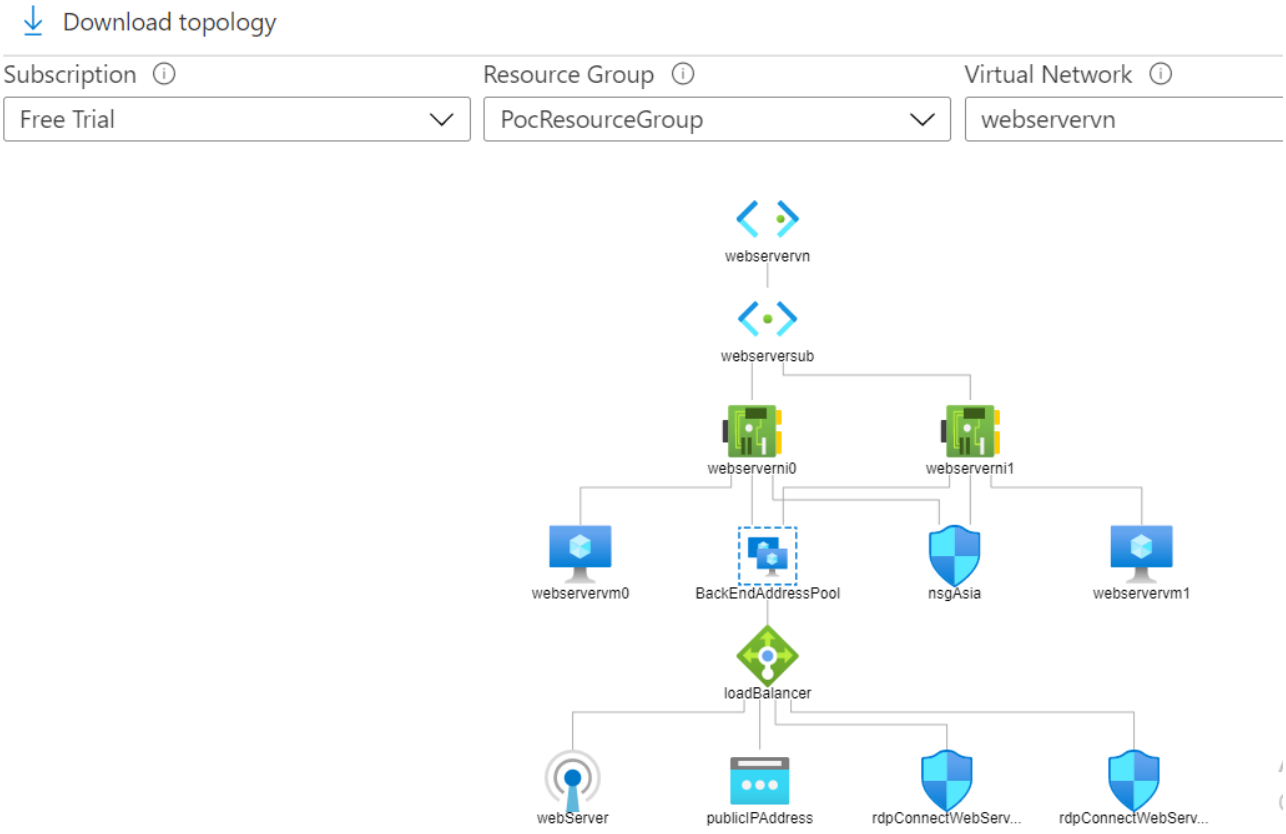


Other information

- All virtual machines



- Vnet Diagram for web servers East Asia



- Vnet Diagram for East US server