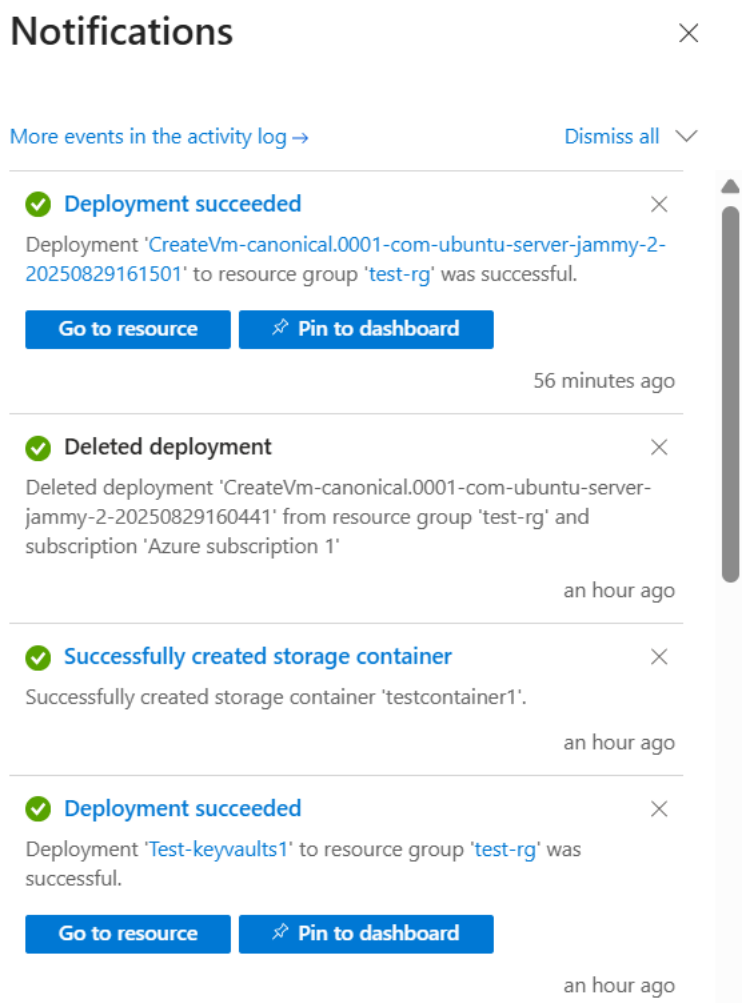
Day3

Created services of VM and other services



Notifications

More events in the activity log → Dismiss all

Deployment succeeded

Deployment 'CreateLoadBalancerBladeV2-20250829154905' to resource group 'test-rg' was successful.

Go to resource Go to resource group

an hour ago

Deployment succeeded

Deployment 'teststore0_1756462546552' to resource group 'test-rg' was successful.

Go to resource Pin to dashboard

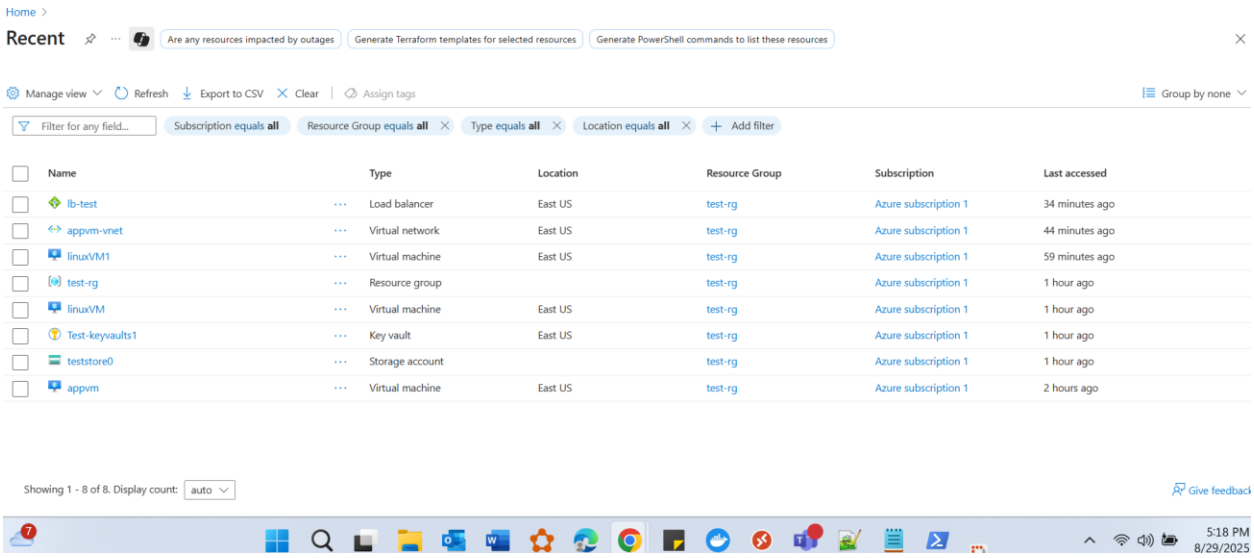
an hour ago

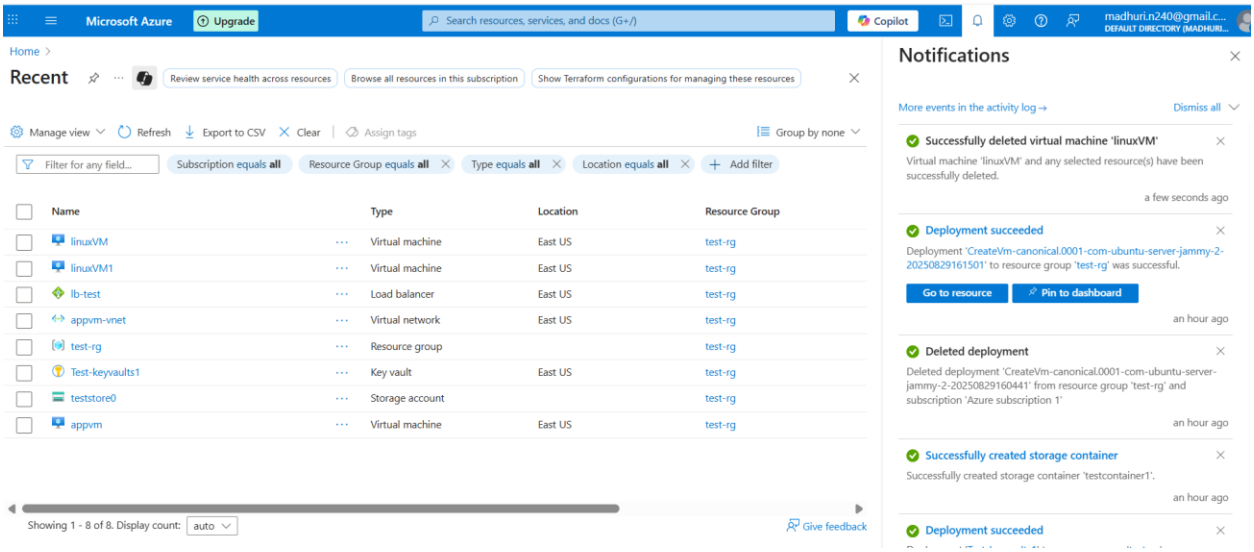
Deployment succeeded

Deployment 'CreateVm-MicrosoftWindowsServer.WindowsServer-202-20250829151005' to resource group 'test-rg' was successful.

Go to resource Go to resource group

2 hours ago





To Create 2 VMs

Windows and ubuntu

Define variables

$location = "East US"

$resourceGroup = "Test-rg"

$vnetName = "TestVNet"

$subnetName = "TestSubnet"

$nsgName = "TsetNSG"

$windowsVMName = "WinVM"

$ubuntuVMName = "UbuntuVM"

$windowsAdmin = "winadmin"

$ubuntuAdmin = "ubuntuadmin"

```
$windowsPassword = ************

$ubuntuPassword = **************


# Create Resource Group

New-AzResourceGroup -Name $resourceGroup -Location $location


# Create Virtual Network and Subnet

$vnet = New-AzVirtualNetwork -ResourceGroupName $resourceGroup -Location $location
`
   -Name $vnetName -AddressPrefix "10.0.0.0/16"

Add-AzVirtualNetworkSubnetConfig -Name $subnetName -AddressPrefix "10.0.0.0/24" -
VirtualNetwork $vnet

$vnet | Set-AzVirtualNetwork


# Create Network Security Group

$nsg = New-AzNetworkSecurityGroup -ResourceGroupName $resourceGroup -Location
$location -Name $nsgName


# Create Public IPs

$winPublicIP = New-AzPublicIpAddress -Name "$windowsVMName-PIP" -
ResourceGroupName $resourceGroup `
   -Location $location -AllocationMethod Dynamic

$ubuntuPublicIP = New-AzPublicIpAddress -Name "$ubuntuVMName-PIP" -
ResourceGroupName $resourceGroup `
   -Location $location -AllocationMethod Dynamic
```

```
# Create NICs

$subnet = Get-AzVirtualNetworkSubnetConfig -VirtualNetwork $vnet -Name $subnetName


$winNIC = New-AzNetworkInterface -Name "$windowsVMName-NIC" -
ResourceGroupName $resourceGroup `

  -Location $location -SubnetId $subnet.Id -PublicIpAddressId $winPublicIP.Id -
NetworkSecurityGroupId $nsg.Id


$ubuntuNIC = New-AzNetworkInterface -Name "$ubuntuVMName-NIC" -
ResourceGroupName $resourceGroup `

  -Location $location -SubnetId $subnet.Id -PublicIpAddressId $ubuntuPublicIP.Id -
NetworkSecurityGroupId $nsg.Id


# Create Windows VM

$winVMConfig = New-AzVMConfig -VMName $windowsVMName -VMSize
"Standard_DS1_v2"

$winVMConfig = Set-AzVMOperatingSystem -VM $winVMConfig -Windows -
ComputerName $windowsVMName `

  -Credential (New-Object
System.Management.Automation.PSCredential($windowsAdmin, (ConvertTo-SecureString
$windowsPassword -AsPlainText -Force)))

$winVMConfig = Set-AzVMSourceImage -VM $winVMConfig -PublisherName
"MicrosoftWindowsServer" `

  -Offer "WindowsServer" -Skus "2022-datacenter" -Version "latest"

$winVMConfig = Add-AzVMNetworkInterface -VM $winVMConfig -Id $winNIC.Id

New-AzVM -ResourceGroupName $resourceGroup -Location $location -VM $winVMConfig


# Create Ubuntu VM
```

```
$ubuntuVMConfig = New-AzVMConfig -VMName $ubuntuVMName -VMSize "Standard_DS1_v2"

$ubuntuVMConfig = Set-AzVMOperatingSystem -VM $ubuntuVMConfig -Linux -ComputerName $ubuntuVMName `

   -Credential (New-Object System.Management.Automation.PSCredential($ubuntuAdmin, (ConvertTo-SecureString $ubuntuPassword -AsPlainText -Force)))

$ubuntuVMConfig = Set-AzVMSourceImage -VM $ubuntuVMConfig -PublisherName "Canonical" `

   -Offer "UbuntuServer" -Skus "18.04-LTS" -Version "latest"

$ubuntuVMConfig = Add-AzVMNetworkInterface -VM $ubuntuVMConfig -Id $ubuntuNIC.Id

New-AzVM -ResourceGroupName $resourceGroup -Location $location -VM $ubuntuVMConfig
```

To Create load balancer

```
# Variables

$resourceGroup = "Test-rg"

$location = "East US"

$lbName = "TestLoadBalancer"

$frontendIPName = "TestFrontendIP"

$backendPoolName = "TestBackendPool"

$probeName = "TestHealthProbe"

$lbruleName = "TestLoadBalancingRule"

$subnetName = "TestSubnet"
```

```
# Get subnet and NICs
$vnet = Get-AzVirtualNetwork -Name "MyVNet" -ResourceGroupName $resourceGroup
$subnet = Get-AzVirtualNetworkSubnetConfig -Name $subnetName -VirtualNetwork $vnet
$winNIC = Get-AzNetworkInterface -Name "WinVM-NIC" -ResourceGroupName $resourceGroup
$ubuntuNIC = Get-AzNetworkInterface -Name "UbuntuVM-NIC" -ResourceGroupName $resourceGroup

# Create Public IP for Load Balancer
$lbPublicIP = New-AzPublicIpAddress -Name "$lbName-PIP" -ResourceGroupName $resourceGroup `
  -Location $location -AllocationMethod Static -Sku Standard

# Create Frontend IP Configuration
$frontendIP = New-AzLoadBalancerFrontendIpConfig -Name $frontendIPName -PublicIpAddress $lbPublicIP

# Create Backend Address Pool
$backendPool = New-AzLoadBalancerBackendAddressPoolConfig -Name $backendPoolName

# Create Health Probe
$probe = New-AzLoadBalancerProbeConfig -Name $probeName -Protocol Tcp -Port 80 -IntervalInSeconds 15 -ProbeCount 2

# Create Load Balancing Rule
$lbrule = New-AzLoadBalancerRuleConfig -Name $lbruleName -FrontendIpConfiguration $frontendIP `
```

-BackendAddressPool $backendPool -Probe $probe -Protocol Tcp -FrontendPort 80 -BackendPort 80 `

-IdleTimeoutInMinutes 5 -EnableFloatingIP $false -LoadDistribution Default

Create Load Balancer

$lb = New-AzLoadBalancer -ResourceGroupName $resourceGroup -Location $location -Name $lbName `

-Sku Standard -FrontendIpConfiguration $frontendIP -BackendAddressPool $backendPool `

-Probe $probe -LoadBalancingRule $lbrule

Associate NICs with Backend Pool

$winNIC.IpConfigurations[0].LoadBalancerBackendAddressPools = $lb.BackendAddressPools

$ubuntuNIC.IpConfigurations[0].LoadBalancerBackendAddressPools = $lb.BackendAddressPools

Set-AzNetworkInterface -NetworkInterface $winNIC

Set-AzNetworkInterface -NetworkInterface $ubuntuNIC

To create Azure Storge Account

Variables

$resourceGroup = "Test-rg"

$location = "East US"

$storageAccountName = "testaccount" # Must be globally unique and lowercase

$containerName = "test1"

```
# Create Storage Account
$storageAccount = New-AzStorageAccount -ResourceGroupName $resourceGroup -Name $storageAccountName `
  -Location $location -SkuName Standard_LRS -Kind StorageV2

# Get Storage Context
$ctx = $storageAccount.Context

# Create Blob Container
New-AzStorageContainer -Name $containerName -Context $ctx -Permission Off
```