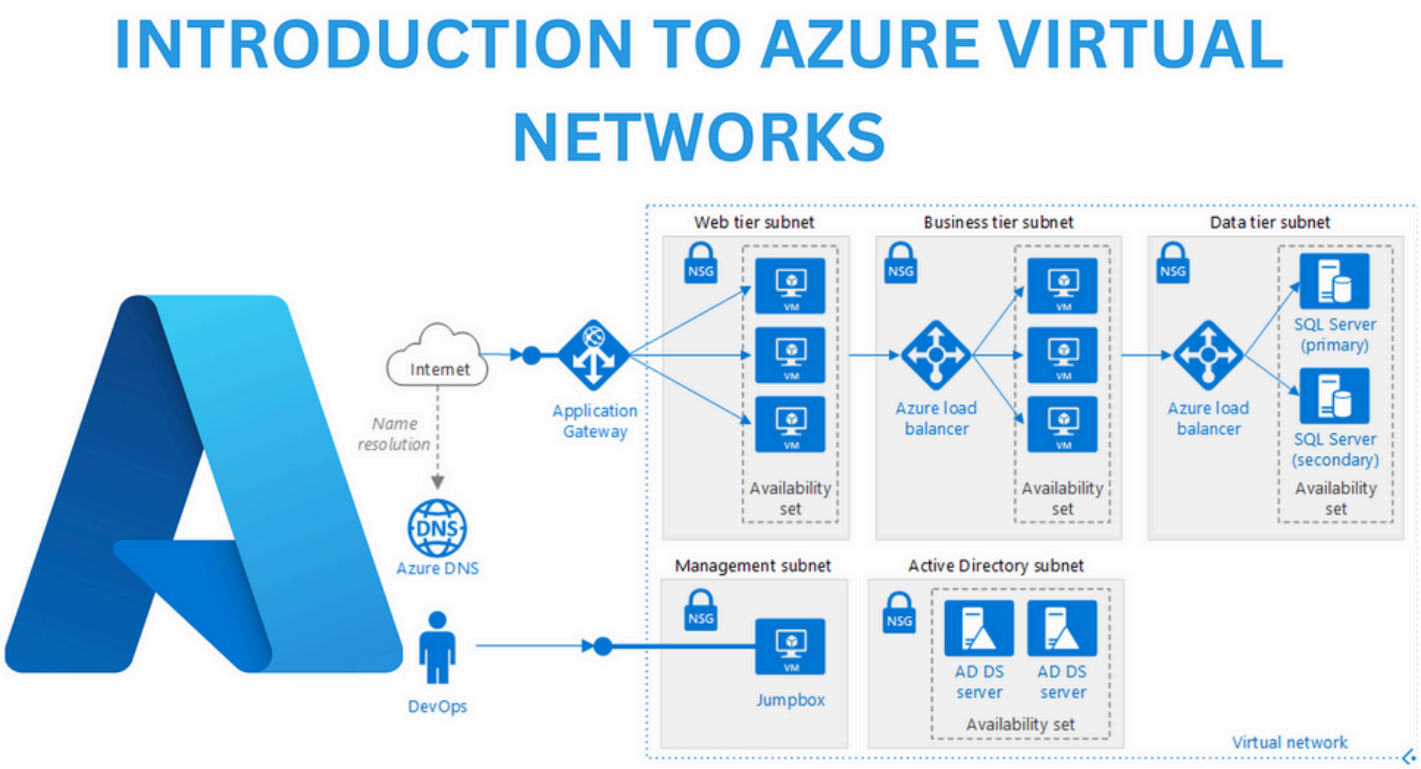
Azure Private DNS Lab: Implementing Secure Cross-VNet Connectivity



Introduction

In this post, I'll walk you through a lab that demonstrates how to set up Azure Private DNS for name resolution across multiple Virtual Networks (VNets). Azure Private DNS provides a secure and scalable way to manage domain names for your virtual machines (VMs) and other resources within Azure, without needing to deploy your own DNS servers. This is especially useful in scenarios where you have multiple VNets that need to communicate with each other. This blog post is structured around a Microsoft Learn lab, and I'll provide detailed steps and screenshots to help you navigate the Azure portal.

Prerequisites

- An Azure subscription
- Azure PowerShell installed
- Familiarity with Azure Virtual Networks

Lab Overview

This lab consists of the following key steps:

1. Create a Resource Group
2. Deploy a Bicep Template
3. Create a Private DNS Zone
4. Link Subnets for Auto Registration
5. Deploy Virtual Machines
6. Create VNet Peerings
7. Verify DNS Resolution

Step-by-Step Instructions

Module 1: Design and Implement a Virtual Network in Azure

Consider the fictional organization Contoso Ltd, which is in the process of migrating infrastructure and applications to Azure. In your role as network engineer, you must plan and implement three virtual networks and subnets to support resources in those virtual networks.

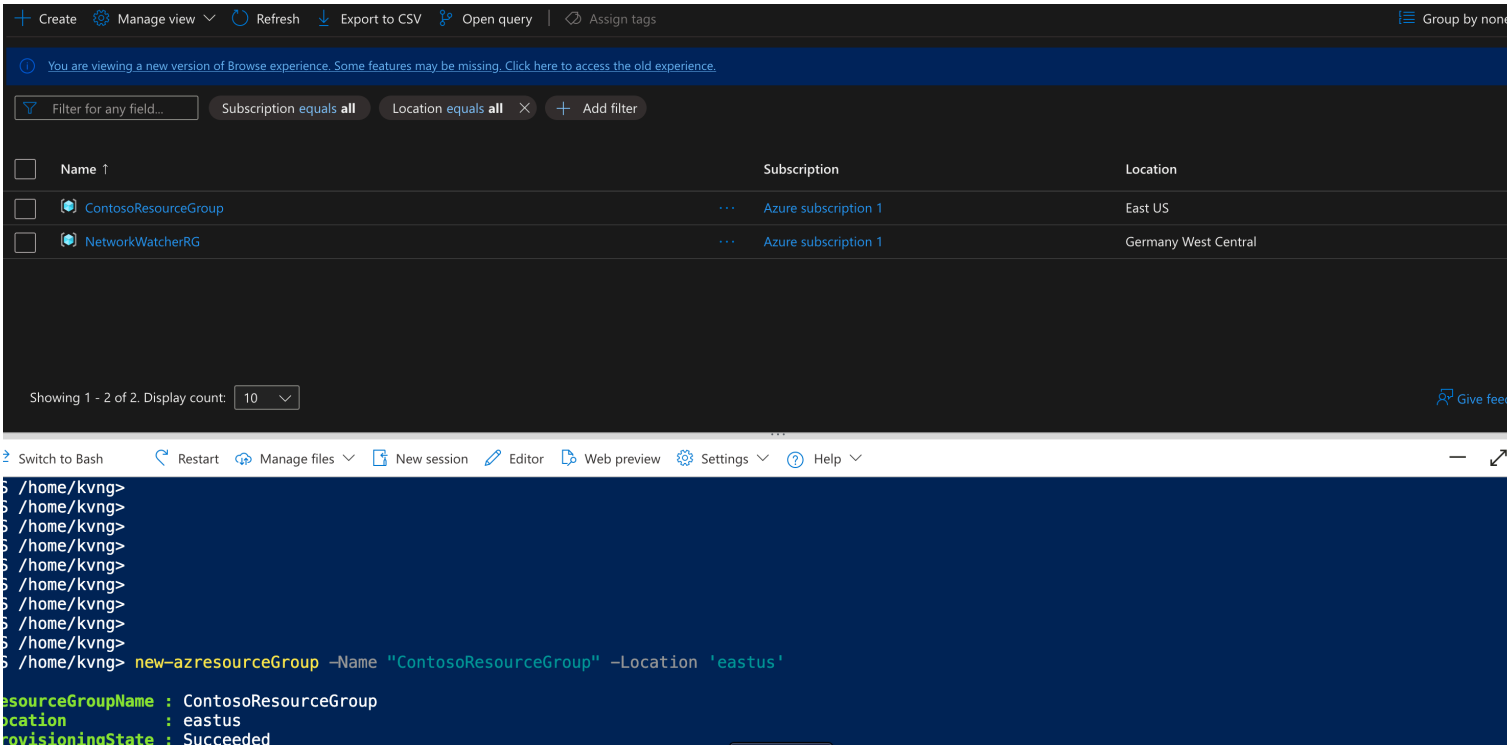
+ Azure Virtual Network is a service that provides the fundamental building block for your private network in Azure. An instance of the service (a virtual network) enables many types of Azure resources to securely communicate with each other, the internet, and on-premises networks. Ensure nonoverlapping address spaces. Make sure your virtual network address space (CIDR block) doesn't overlap with your organization's other network ranges.

+ All Azure resources in a virtual network are deployed into subnets within the virtual network. Subnets enable you to segment the virtual network into one or more subnetworks and allocate a portion of the virtual network's address space to each subnet. Your subnets shouldn't cover the entire address space of the virtual network. Plan ahead and reserve some address space for the future.

Task 1: Create the Contoso Resource Group

Use Azure PowerShell to create a resource group. This command creates a container for all the resources used in this lab.

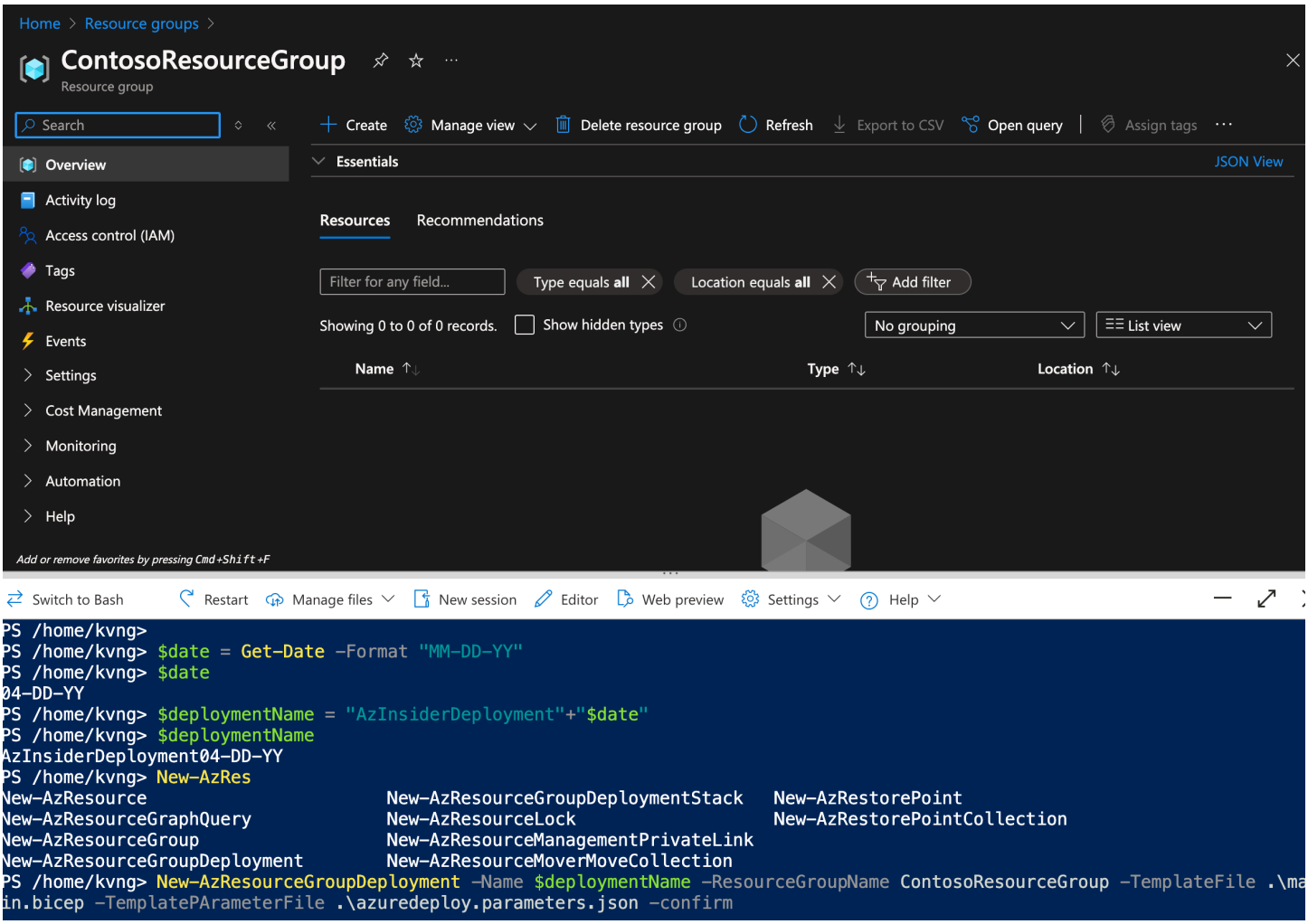
New-AzResourceGroup -Name Contoso -Location <YourAzureRegion>



- **To verify in the Azure portal:**
  - Go to the Azure portal.
  - Search for "Resource groups".
  - Click on "Resource groups".
  - Confirm that "Contoso" is in the list.

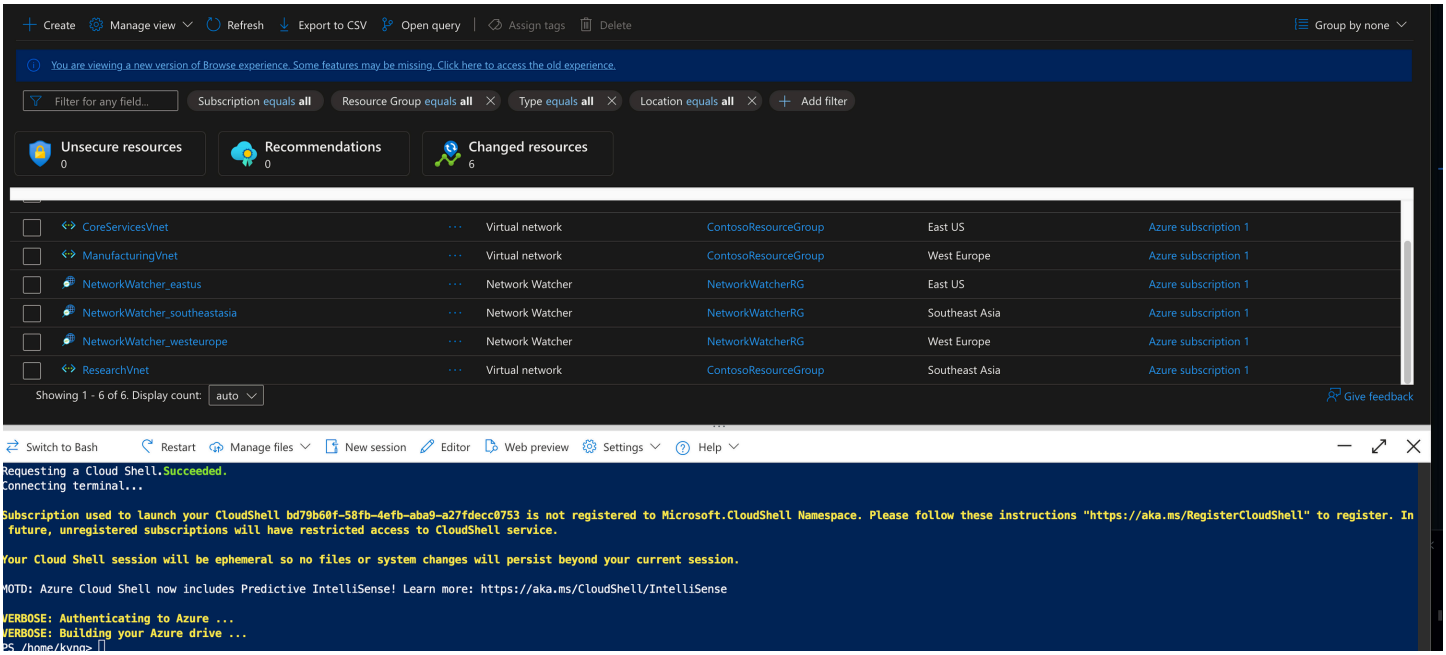
Task 2: Create the CoreServicesVnet Virtual Network and Subnets

The Bicep template will handle this, but you can verify the configuration.



Task 3: Create the ManufacturingVnet Virtual Network and Subnets

The Bicep template will handle this, but you can verify the configuration.

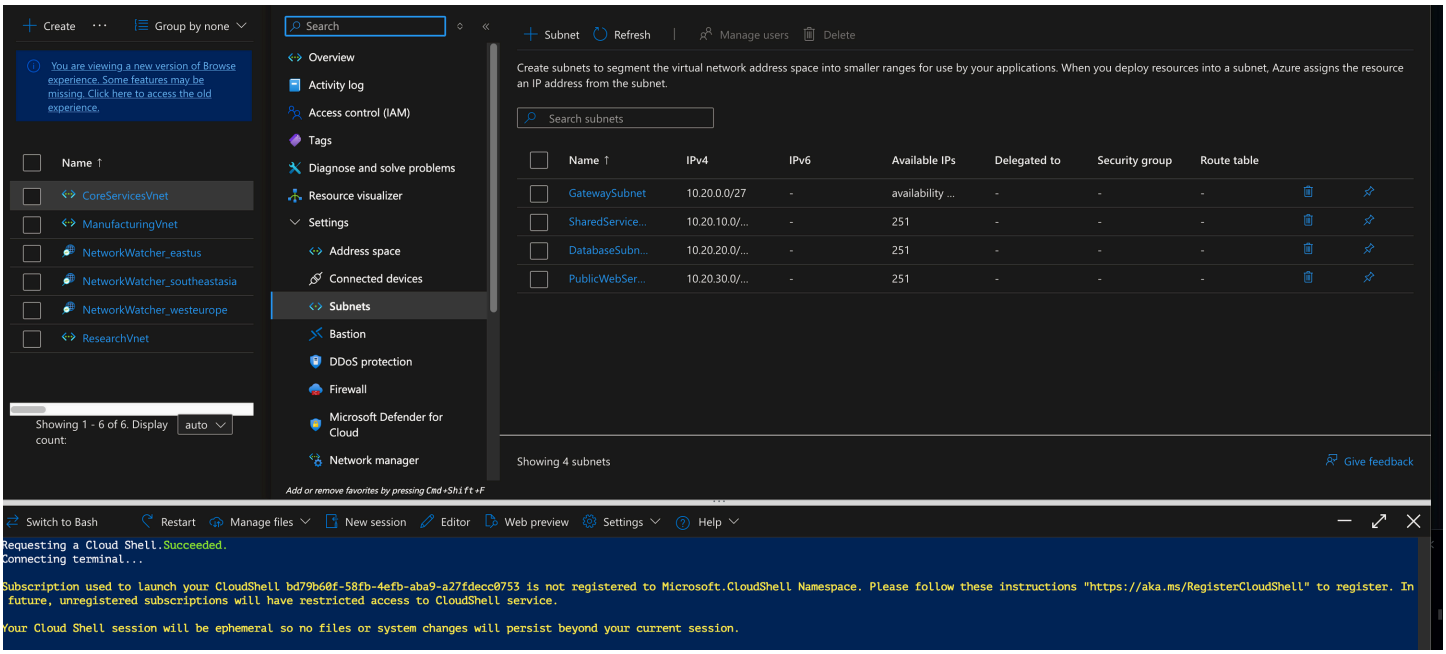


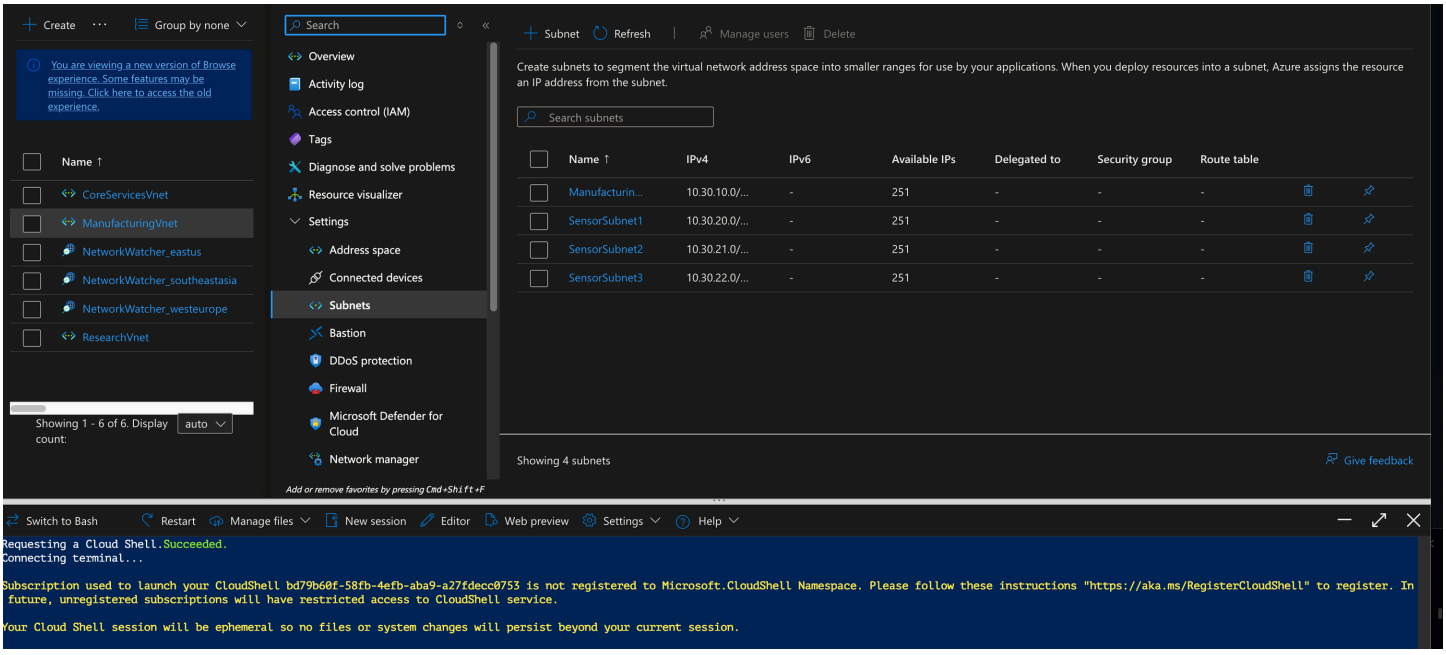
Task 4: Create the ResearchVnet Virtual Network and Subnets

The Bicep template will handle this, but you can verify the configuration.

Task 5: Verify the Creation of VNets and Subnets

- **To verify in the Azure portal:**
  - Go to the Azure portal.
  - Search for "Virtual networks".
  - Click on "Virtual networks".
  - Confirm that "CoreServicesVnet", "ManufacturingVnet", and "ResearchVnet" are in the list.
  - Click on each VNet to verify the subnets:
    - CoreServicesVnet should have subnets: "CoreServicesSubnet"
    - ManufacturingVnet should have subnets: "ManufacturingSubnet"
    - ResearchVnet should have subnets: "ResearchSubnet"





Module 2: Configure DNS Settings in Azure

This module focuses on configuring the Private DNS Zone.

In this unit, you will configure DNS name resolution for Contoso Ltd. You will create a private DNS zone named contoso.com, link the VNets for registration and resolution, and then create two virtual machines and test the configuration.

+ Azure DNS is a cloud service that allows you to host and manage domain name system (DNS) domains, also known as DNS zones.

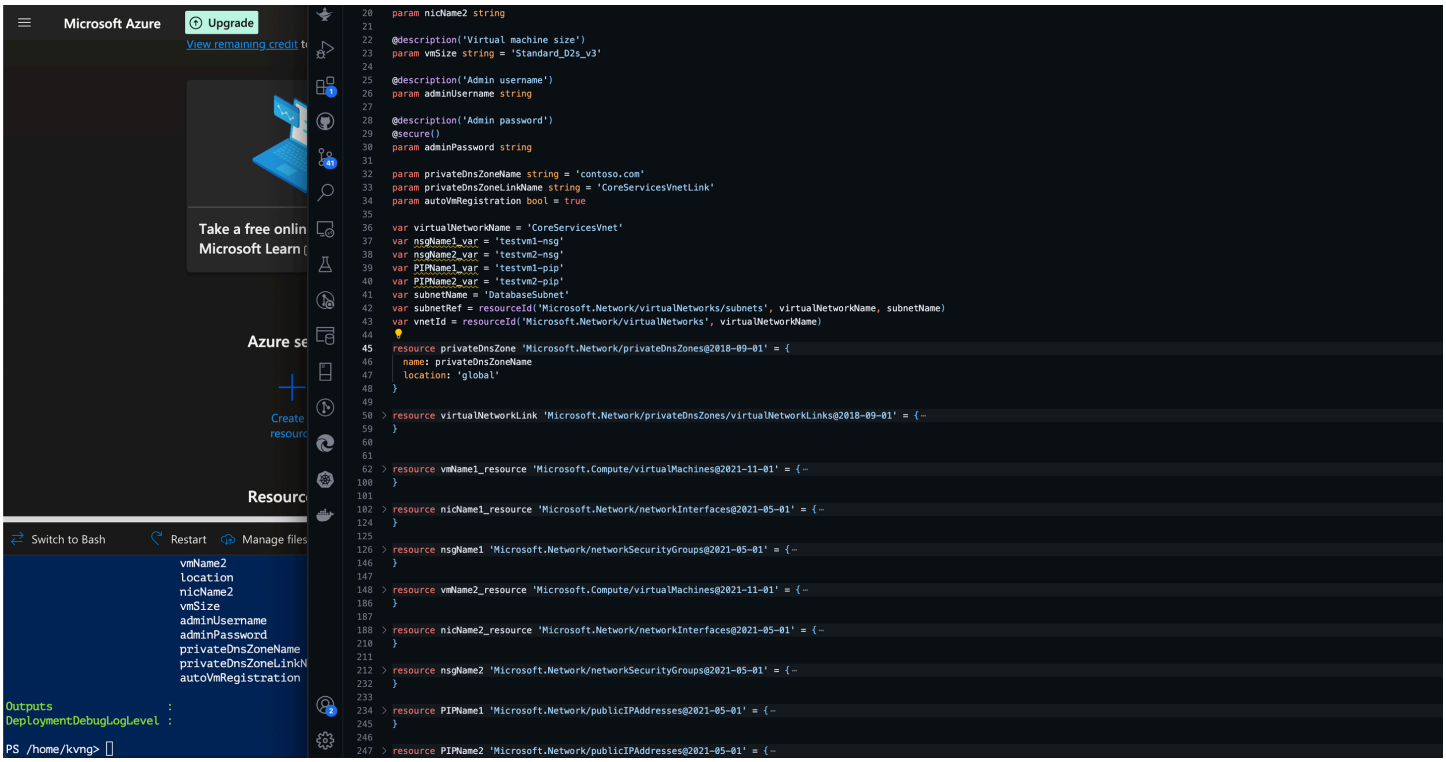
+ Azure DNS public zones host domain name zone data for records that you intend to be resolved by any host on the internet.

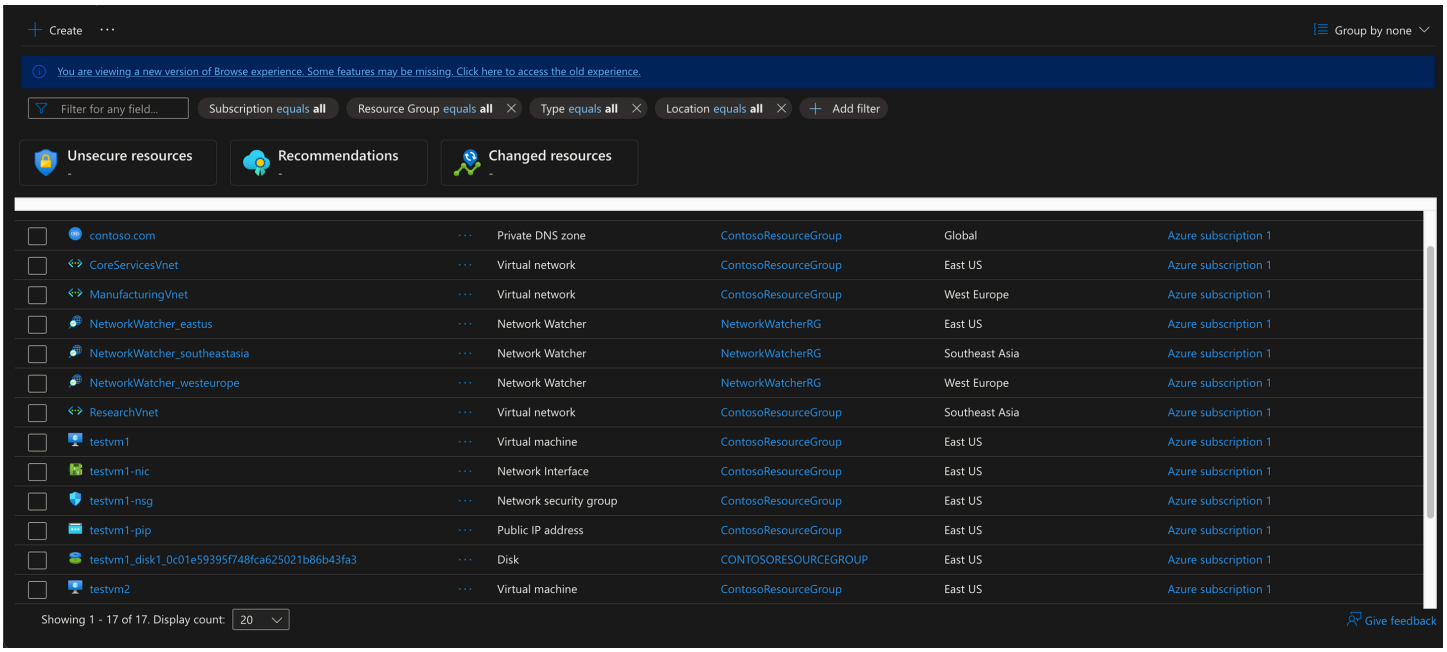
+ Azure Private DNS zones allow you to configure a private DNS zone namespace for private Azure resources.

+ A DNS zone is a collection of DNS records. DNS records provide information about the domain.

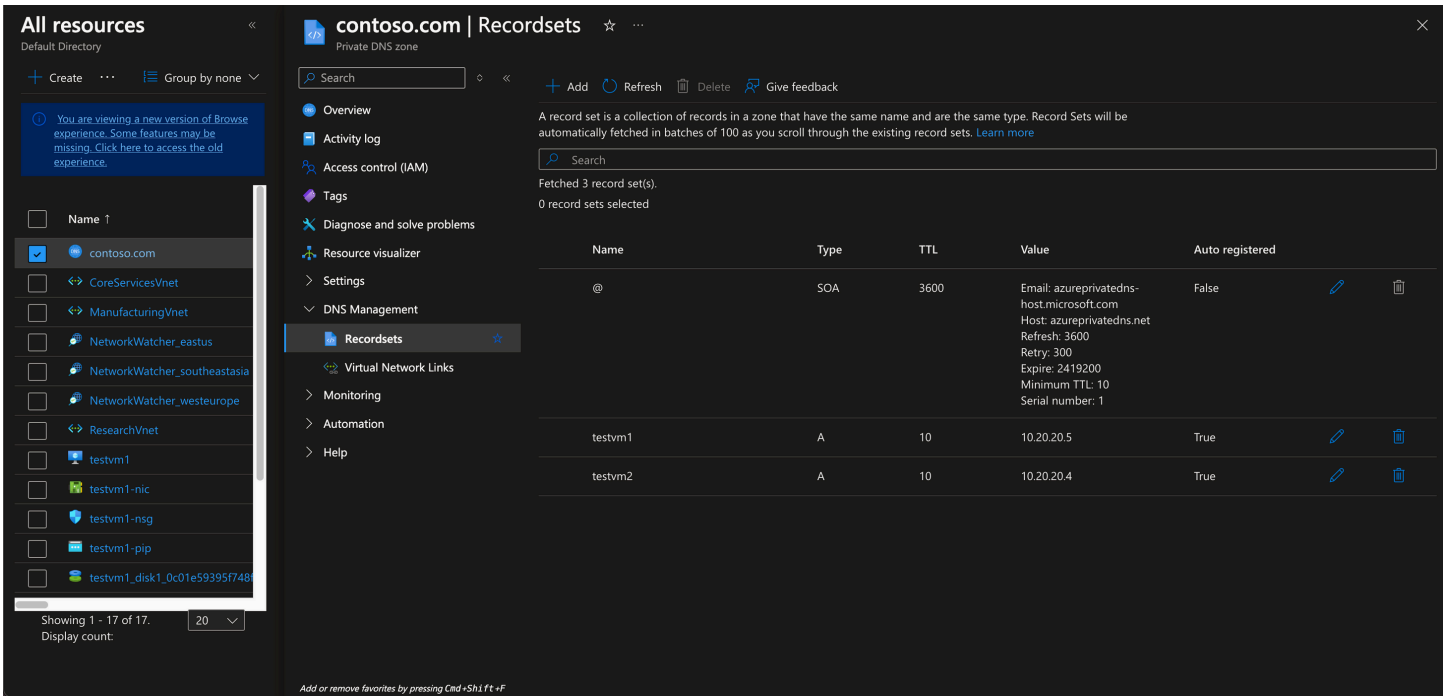
Task 1: Create a Private DNS Zone

The Bicep template automates this.





- **To verify in the Azure portal:**
  - Go to the Azure portal.
  - Search for "Private DNS zones".
  - Click on "Private DNS zones".
  - Confirm that the private DNS zone (e.g., "contoso.com") is in the list.
  - Click on the private DNS zone to view its properties.
  - Verify the Name servers are listed.



Task 2: Link Subnet for Auto Registration

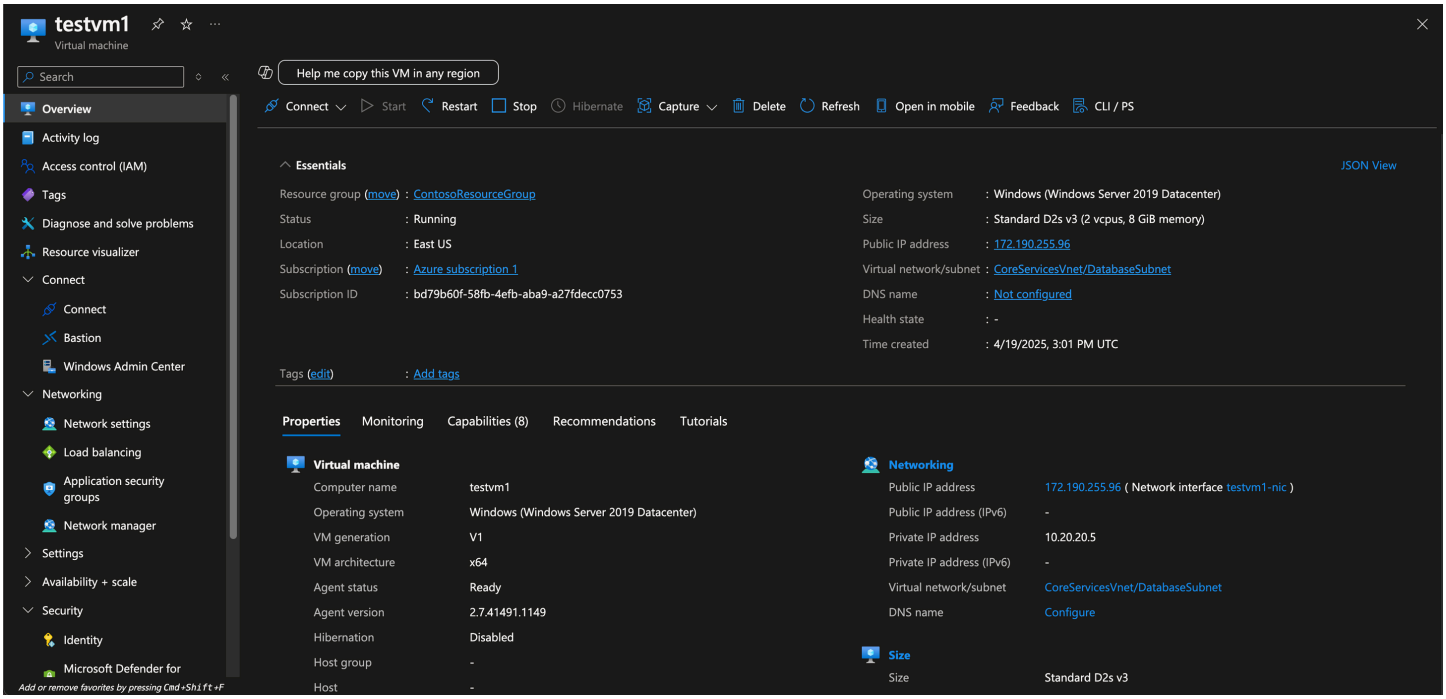
The Bicep template links the subnets to the Private DNS Zone, enabling automatic registration of VMs in those subnets.

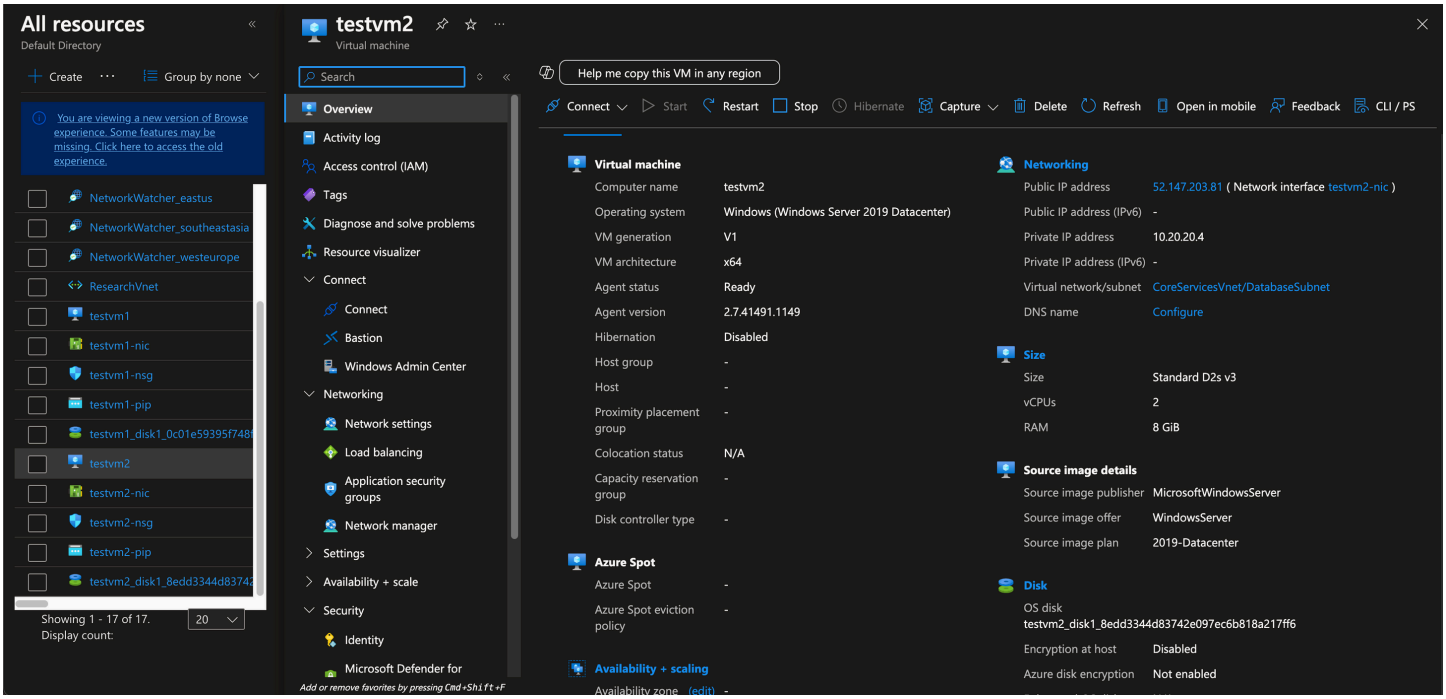
- As mentioned before, when you link a VNet to your private DNS Zone (think of it as a local organization phonebook), you're telling the VNet's resources to use this private zone for name resolution. If multiple VNets are linked to the same private DNS Zone, it's like giving all those separate network spaces the same custom phonebook. This allows VMs in different VNets to find each other using the names in your private DNS Zone.
- **To verify in the Azure portal:**

- In the Private DNS zone, go to "Virtual network links".
- Confirm that the virtual networks (CoreServicesVnet, ManufacturingVnet, and ResearchVnet) and their corresponding subnets are listed and linked.
- Check the "Auto registration" column to ensure it is enabled.

Task 3: Create Virtual Machines to Test the Configuration

The Bicep template creates the VMs.



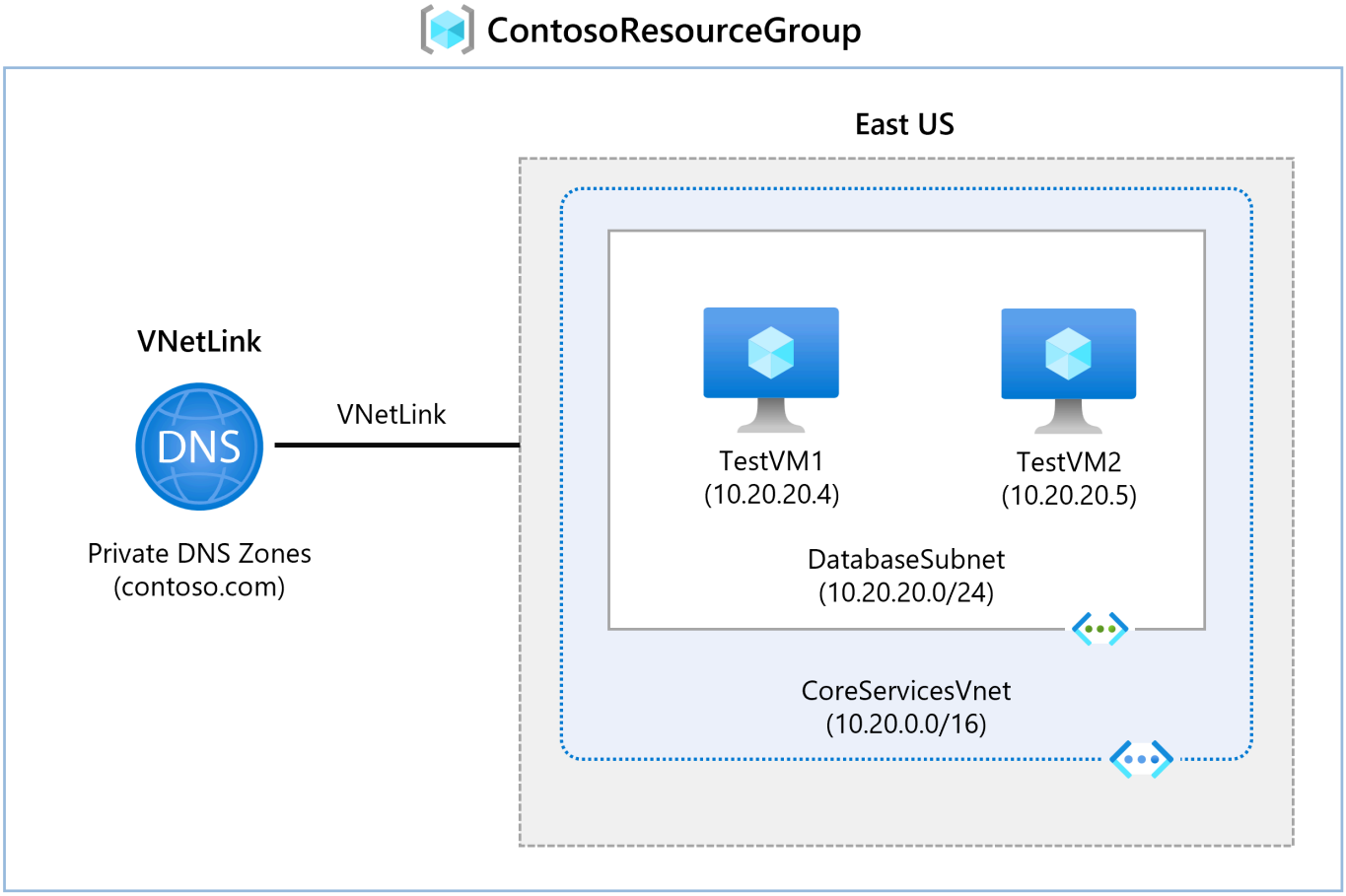


Module 3: Connect two Azure Virtual Networks using global virtual network peering

Virtual network peering enables you to seamlessly connect two Azure virtual networks. The virtual networks appear as one for connectivity purposes. Azure supports connecting virtual networks within the same Azure region and across Azure regions (global). The traffic between virtual machines in peered virtual networks is routed directly through the Microsoft backbone infrastructure, not through a gateway or over the public Internet. You can resize the address space of Azure virtual networks that are peered without incurring any downtime on the currently peered address space. Enhance Security by not providing default Outbound access. Therefore, to enable Connectivity for VMs within this Subnet, it's necessary to explicitly grant Outbound access. A NAT Gateway is the Recommended way to provide connectivity for Virtual machines in the subnet "Subnets without public IP Address".

Enable Cross-Virtual Network Connectivity with Peering: Allows you to connect separate VNets with Optimal Network performance, whether they are in the Same Region [Regional VNet peering] or in different regions [Global VNet peering]. Traffic between peered VNets is private. For connectivity purposes, once peered, the VNets appear as One. This offers the following Benefits:

- Low Latency, high-bandwidth connection b/w Resources in different VNets.
- Ability to apply Network Security Groups in either VNet to block access to other VNets or subnets.
- Ability to transfer data b/w VNets across Azure subscriptions, Entra Tenants, Deployment models, and Regions.
- Ability to peer a VNet created through the Resource Manager to one created through classic deployment.
- No downtime to Resources in other VNets is required when creating/after creating peering.



Azure Private DNS Domains

AZURE DNS - A hosting service for DNS domains, providing name resolution using AZURE infrastructure. Supports private DNS domains. In addition to internet-facing public domains, AZURE supports private DNS Domains used locally within Azure.

- AZURE DNS resolves domain names in a Virtual network without the need to add a Custom DNS solution.
- Private DNS Zones allow you to use your own custom domain names rather than the Azure-provided names.
- It provides name Resolution for VMs within a virtual network and between virtual networks.
- You can configure zone names with a split-horizon view - allowing a private & public DNS zone to share the same name.
- VNet is used to store & organize information about various elements of your cloud.

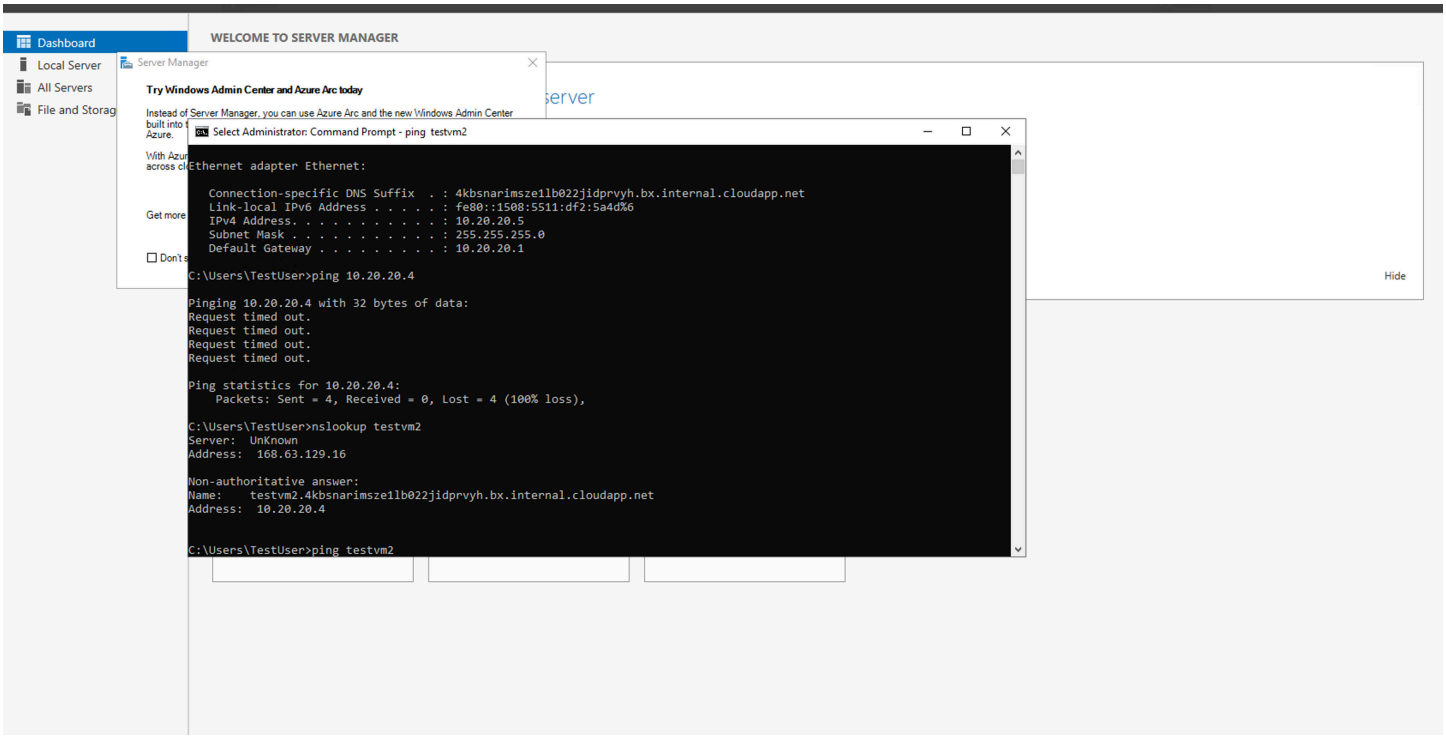
```
resource nsgName1 'Microsoft.Network/networkSecurityGroups@2021-05-01' = {
}

resource PIPName1 'Microsoft.Network/publicIPAddresses@2021-05-01' = {
  name: PIPName1_var
  location: location
  sku: {
    name: 'Basic'
    tier: 'Regional'
  }
  properties: {
    publicIPAddressVersion: 'IPv4'
    publicIPAllocationMethod: 'Dynamic'
  }
}


//This creates a peering from CoreServicesVnet-to-ManufacturingVnet
resource peer1 'microsoft.network/virtualNetworks/virtualNetworkPeerings@2020-05-01' = {
  name: '${vnet0}/to-ManufacturingVnet'
  properties: {
    allowVirtualNetworkAccess: true
    allowForwardedTraffic: false
    allowGatewayTransit: false
    useRemoteGateways: false
    remoteVirtualNetwork: {
      id: resourceId(remoteVnetRg, 'Microsoft.Network/virtualNetworks', vnet1)
    }
  }
  dependsOn:[
    vmName1_resource
  ]
}

//This creates a peering from ManufacturingVnet-to-CoreServicesVnet
resource peer4 'microsoft.network/virtualNetworks/virtualNetworkPeerings@2020-05-01' = {
  name: '${vnet1}/to-CoreServicesVnet'
  properties: {
    allowVirtualNetworkAccess: true
    allowForwardedTraffic: false
    allowGatewayTransit: false
    useRemoteGateways: false
    remoteVirtualNetwork: {
      id: resourceId(remoteVnetRg, 'Microsoft.Network/virtualNetworks', vnet0)
    }
  }
  dependsOn:[
    vmName1_resource
  ]
}
```

Task 1: Create a Virtual Machine to test the configuration

The Bicep template creates the VMs.

Task 2: Connect to the Test VMs using RDP

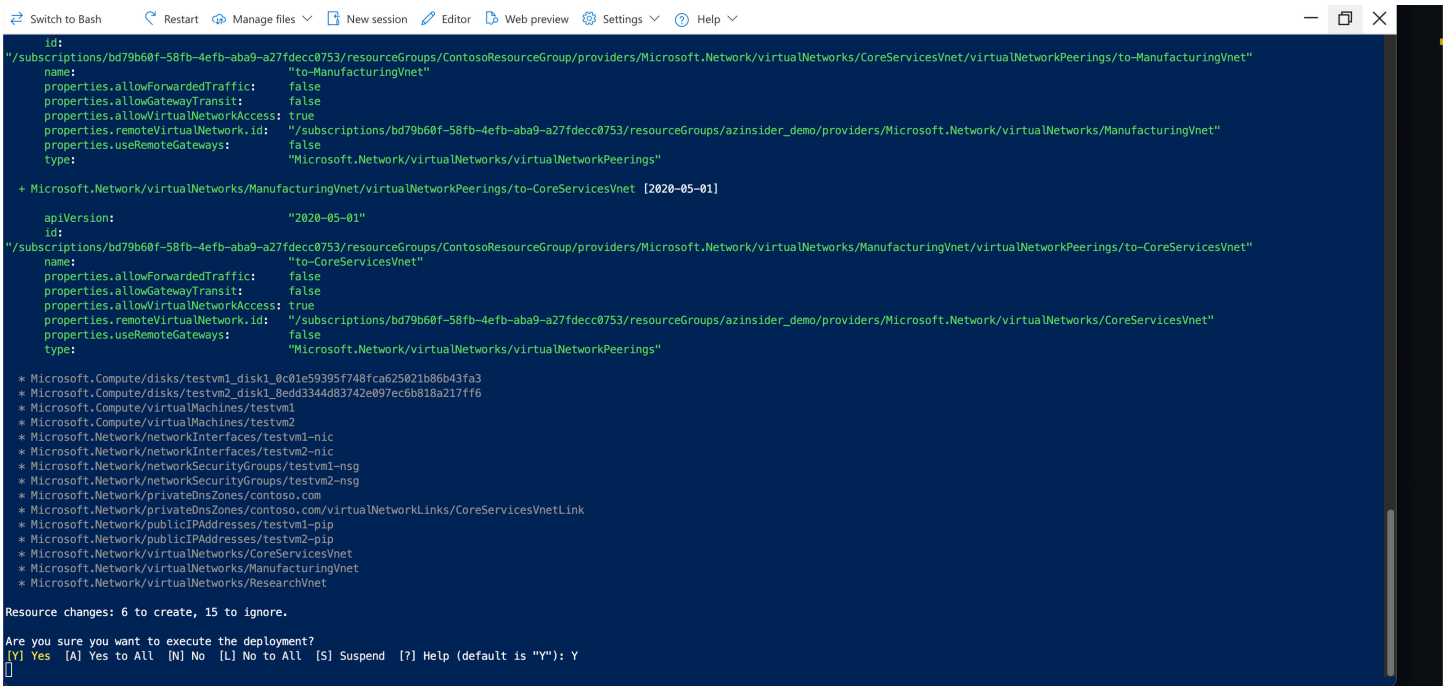
Use Remote Desktop Protocol (RDP) to connect to the VMs.



- **To connect via RDP:**
  - In the Azure portal, go to the Virtual Machine.
  - Click "Connect".
  - Select "RDP".
  - Download the RDP file.
  - Use the downloaded RDP file and the username/password you provided during VM creation to connect.

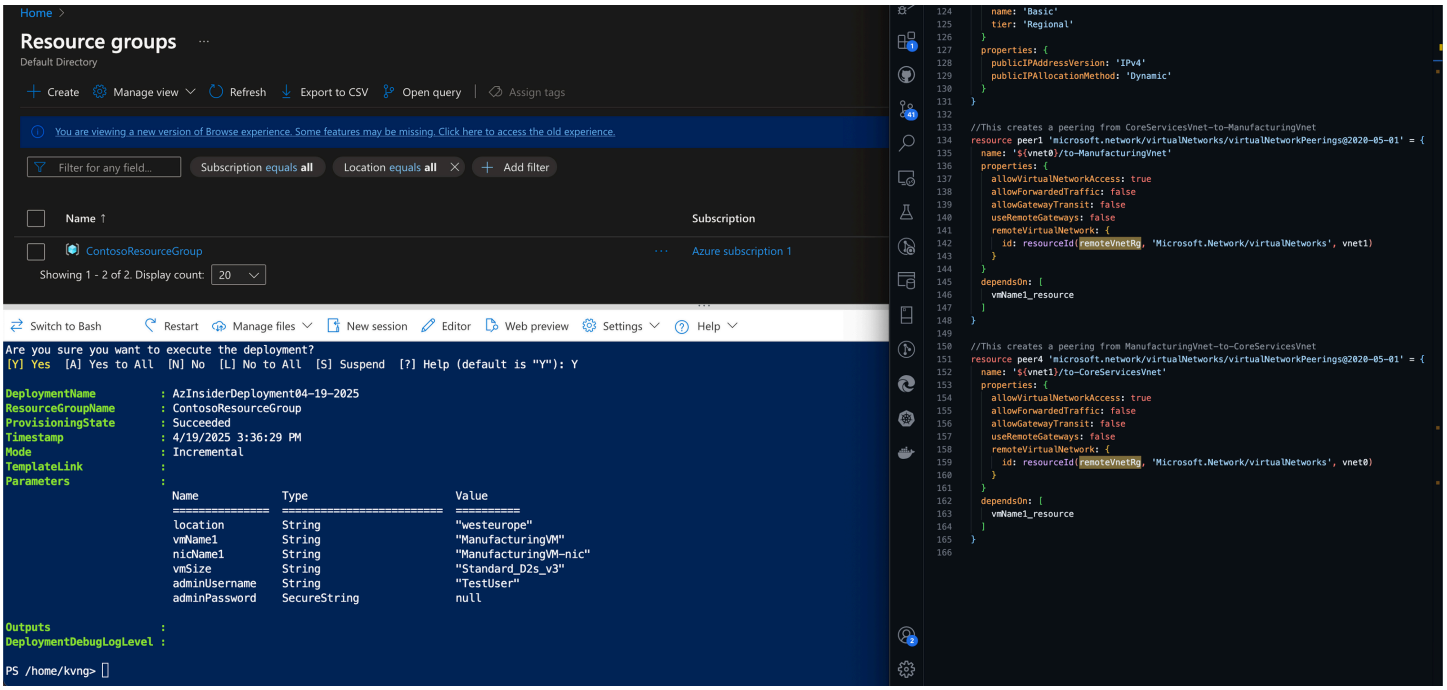
Task 3: Test the connection between the VMs

The Bicep template creates the VMs.

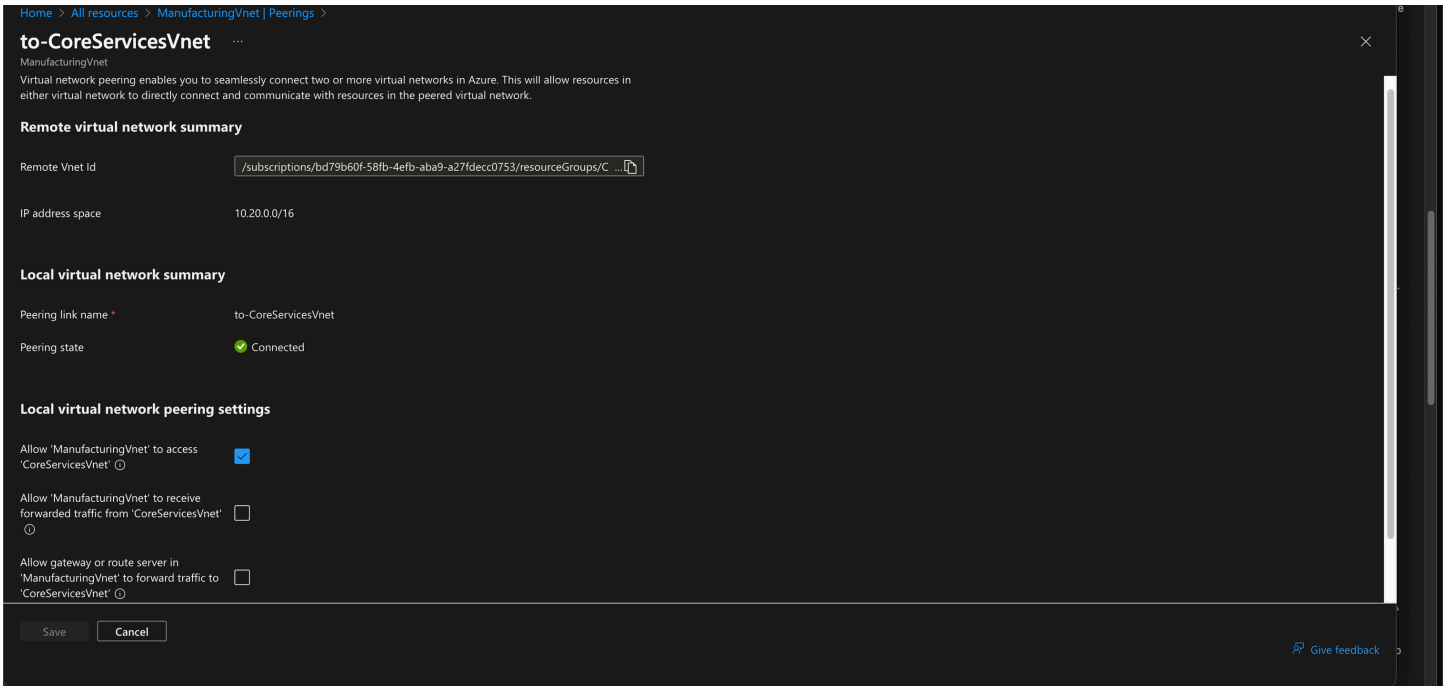


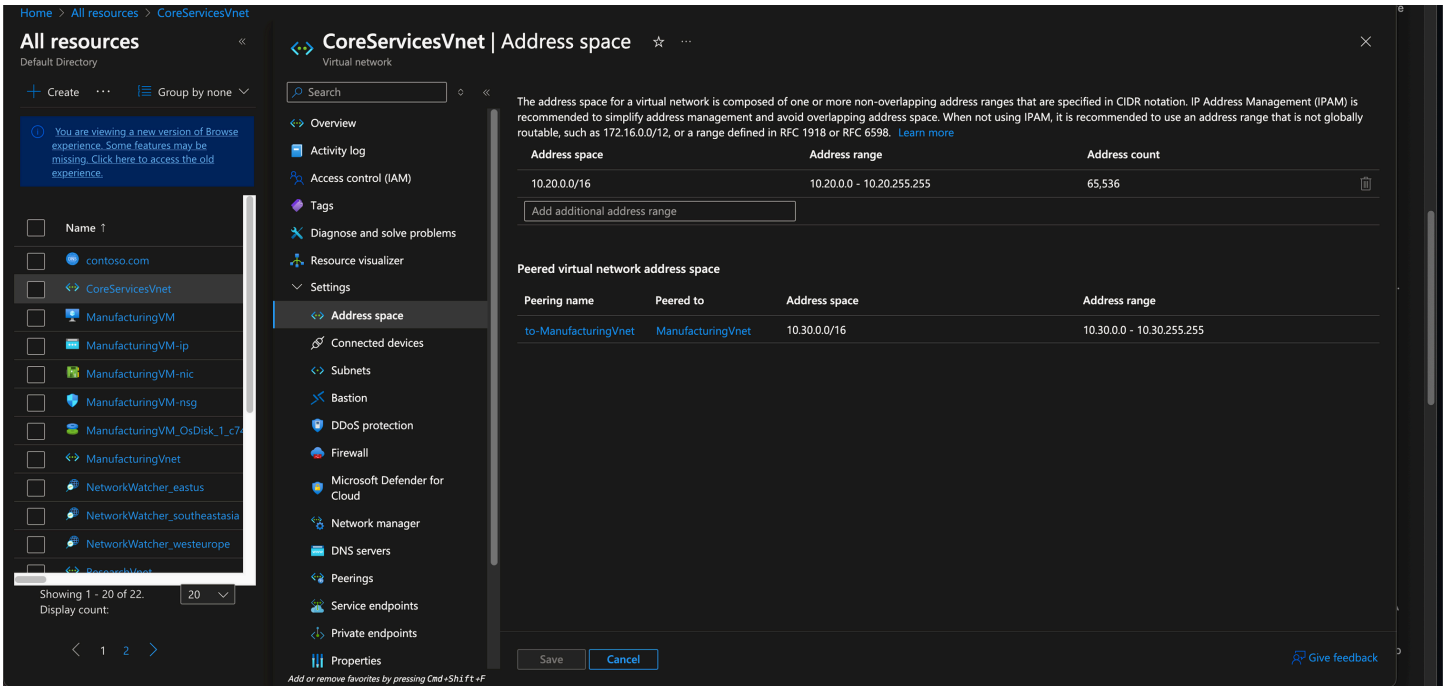
Task 4: Create VNet peerings between CoreServicesVnet and ManufacturingVnet

The Bicep template creates the VNet peerings.



- **To verify VNet peering in the Azure portal:**
  - Go to the Azure portal.
  - Search for "Virtual networks".
  - Click on "Virtual networks".
  - Click on "CoreServicesVnet".
  - Go to "Peerings".
  - Confirm that a peering exists with "ManufacturingVnet" and that the peering status is "Connected".
  - Repeat the process for "ManufacturingVnet" to confirm the reciprocal peering.
- **Important Note:** Before configuring VNet peering, you might find that pinging a host in one VNet from a host in another VNet fails due to firewall restrictions. This highlights a key benefit of VNet peering.



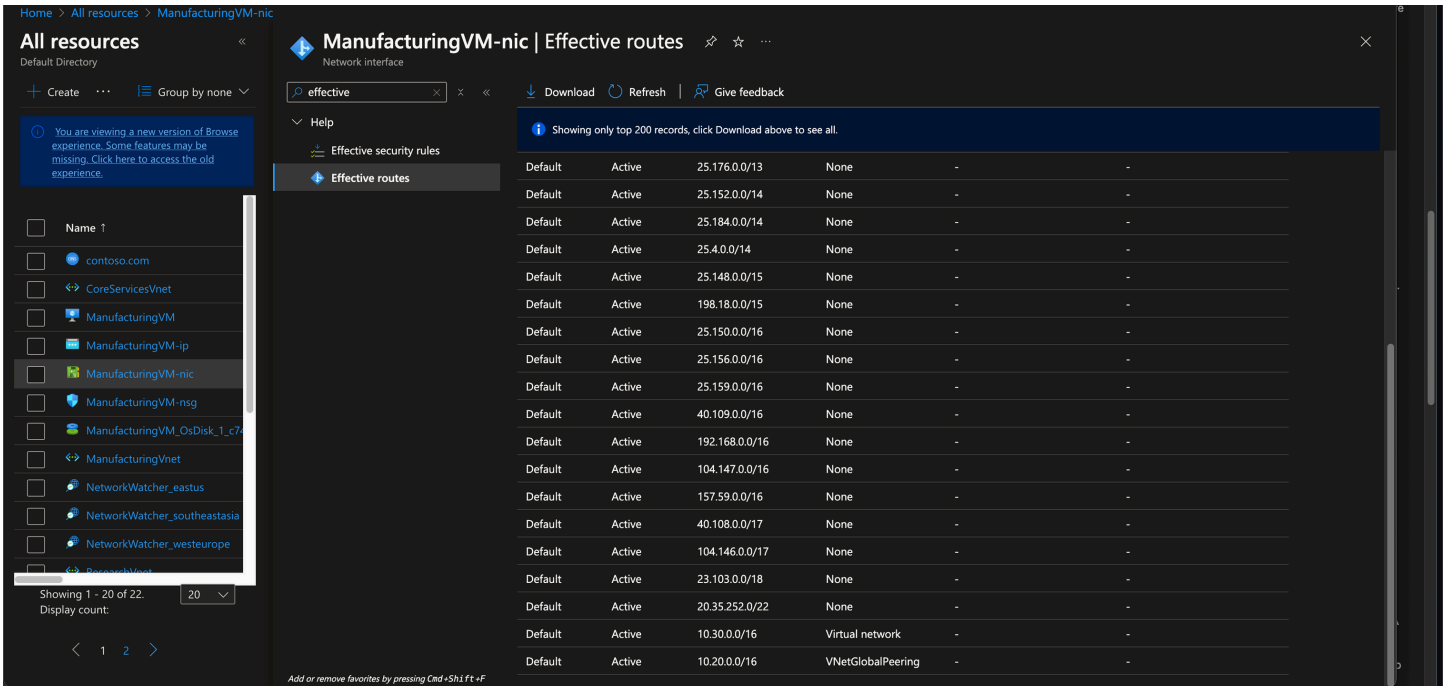


Task 5: Test the connection between the VMs

Once the deployment is complete, we need to verify that DNS resolution and network connectivity are working correctly.

When you link a VNet to your private DNS Zone (local organization phonebook), it tells the VNet's resources to use this private zone for name resolution. If multiple VNets were linked to the same private DNS Zone, it's like giving all those separate network spaces the same custom phonebook; this allows VMs in different VNets to find each other using the names in your private DNS Zone.

- **Verify records are present in the private DNS Zone:** Check the Azure portal to confirm that the DNS records for the VMs have been created in the Private DNS Zone.
  - In the Private DNS zone, go to "Overview".
  - Confirm that the expected DNS records (A records) for your VMs are present. The names should match the VM names.
- **RDP into any of the deployed VMs to do a nslookup:**
  - Use Remote Desktop Protocol (RDP) to connect to one of the VMs.
  - Open Command Prompt.
  - Use the nslookup command to query the DNS records of the other VM. For example: nslookup <other_vm_name>. This will confirm that the private DNS zone is correctly resolving the VM names to their IP addresses.
  - *(Include screenshots of the Azure portal showing the DNS records and the nslookup command output)*
- **Test VM-to-VM Connectivity:**
  - From one VM, ping the private IP address of the other VM. For example: ping <other_vm_private_ip_address>.
  - The test connection should succeed, and you will see a result similar to the following:
  - ![Powershell window with Test-NetConnection 10.20.20.4 -port 3389 showing TCP test succeeded: true]
  - **Crucially, after successfully configuring VNet peering, this ping should now succeed.** This demonstrates that VNet peering has correctly configured the network to allow traffic to flow between the two VNets.



Conclusion

By following these steps, you've successfully set up Azure Private DNS for cross-VNet name resolution. This provides a secure and efficient way for your VMs and services in different VNets to communicate with each other. Azure Private DNS simplifies network management and eliminates the need for manual DNS configuration or custom DNS servers. The Bicep template automates much of the deployment, and the Azure portal provides a user-friendly way to verify the configuration and troubleshoot any issues. The importance of VNet peering in enabling this cross-VNet communication is clearly demonstrated by the change in ping behavior before and after its configuration. For More Resources, Check the link below

+ [Design an IP addressing schema for your Azure deployment] (https://learn.microsoft.com/training/modules/design-ip-addressing-for-azure/).

+ [Introduction to Azure Virtual Networks] (https://learn.microsoft.com/training/modules/introduction-to-azure-virtual-networks/).

+ [Introduction to Azure Virtual Networks] (https://learn.microsoft.com/training/modules/introduction-to-azure-virtual-networks/).

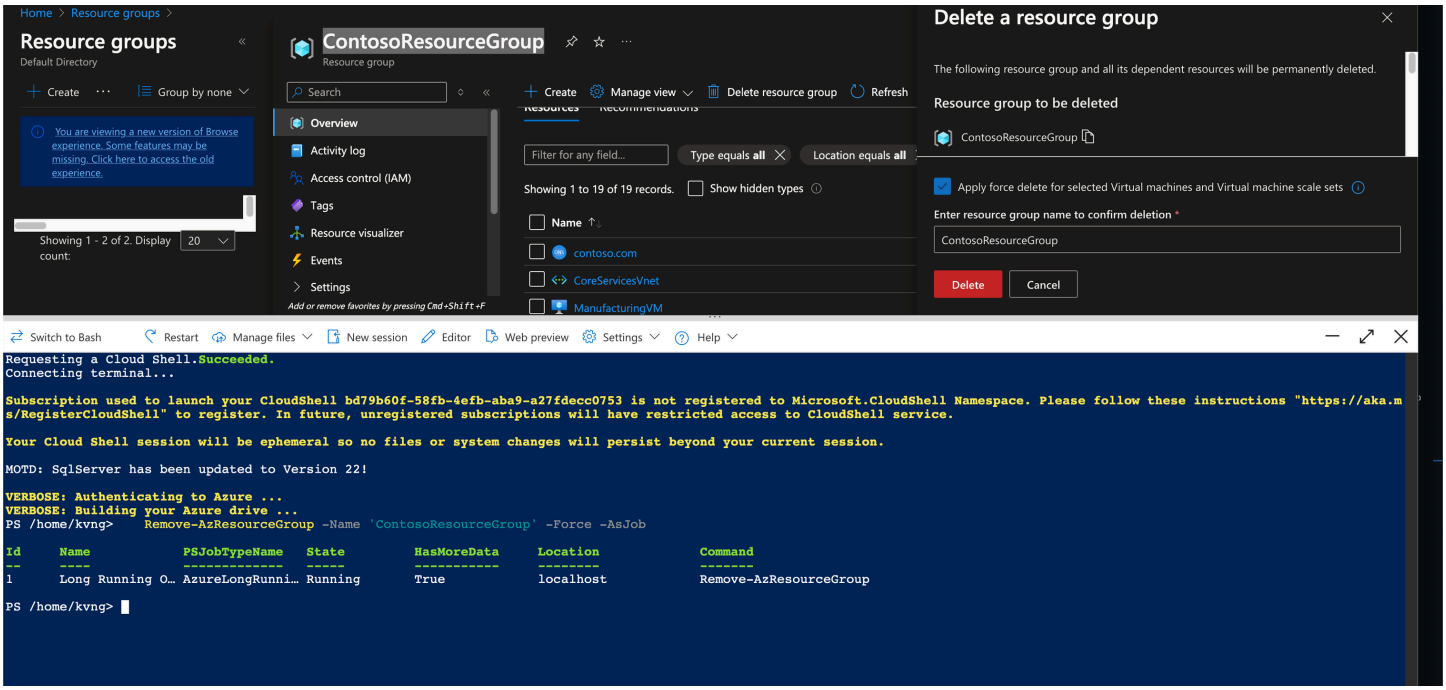
+ [Distribute your services across Azure virtual networks and integrate them by using virtual network peering](https://learn.microsoft.com/training/modules/integrate-vnets-with-vnet-peering/).

+ [Introduction to Azure DNS](https://learn.microsoft.com/training/modules/intro-to-azure-dns/).

+ [Host your domain on Azure DNS](https://learn.microsoft.com/training/modules/host-domain-azure-dns/).

Clean up resources

Note: Remember to remove any newly created Azure resources that you no longer use. Removing unused resources ensures you will not see unexpected charges.


Home > Resource groups >
Resource groups
Default Directory
Create
Group by none
You are viewing a new version of Browse experience. Some features may be missing. Click here to access the old experience.
Showing 1 - 2 of 2. Display count: 20
ContosoResourceGroup
Resource group
Search
Overview
Activity log
Access control (IAM)
Tags
Resource visualizer
Events
Settings
Add or remove favorites by pressing Cmd+Shift+F
Create
Manage view
Delete resource group
Refresh
Filter for any field...
Type equals all
Location equals all
Showing 1 to 19 of 19 records.
Show hidden types
Name
contoso.com
CoreServicesVnet
ManufacturingVM
Delete a resource group
The following resource group and all its dependent resources will be permanently deleted.
Resource group to be deleted
ContosoResourceGroup
Apply force delete for selected Virtual machines and Virtual machine scale sets
Enter resource group name to confirm deletion *
ContosoResourceGroup
Delete
Cancel
Switch to Bash
Restart
Manage files
New session
Editor
Web preview
Settings
Help
Requesting a Cloud Shell.Succeeded.
Connecting terminal...
Subscription used to launch your CloudShell bd79b60f-58fb-4efb-aba9-a27fdecc0753 is not registered to Microsoft.CloudShell Namespace. Please follow these instructions "https://aka.ms/RegisterCloudShell" to register. In future, unregistered subscriptions will have restricted access to CloudShell service.
Your Cloud Shell session will be ephemeral so no files or system changes will persist beyond your current session.
MOTD: SqlServer has been updated to Version 22!
VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
PS /home/kvng> Remove-AzResourceGroup -Name 'ContosoResourceGroup' -Force -AsJob
Id Name PSJobTypeName State HasMoreData Location Command
1 Long Running O… AzureLongRunni… Running True localhost Remove-AzResourceGroup
PS /home/kvng>