ChatGPT

**Research and Development Document: Azure Virtual Network (VNet) Setup**

**Objective:**  
The goal of this document is to outline the steps required to create two Azure VNets with their respective subnets, deploy Windows and Linux virtual machines (VMs) in each subnet, and establish VNet peering between the two VNets. Additionally, ensure that VMs across different subnets can ping each other.

**Steps to Achieve the Objective:**

**1. Create Azure VNets and Subnets**

1. **Log in to Azure Portal:**
   * Navigate to the Azure portal at <https://portal.azure.com>.
2. **Create First VNet (VNet1):**
   * Go to **Create a resource** > **Networking** > **Virtual network**.
   * Provide a name for the VNet (e.g., VNet1).
   * Select a region and leave other settings to default unless specific configurations are required.
   * Create two subnets within VNet1:
     + Subnet1: Name it and assign an appropriate IP range.
     + Subnet2: Name it and assign an appropriate IP range.
3. **Create Second VNet (VNet2):**
   * Follow the same steps as above, ensuring to create different IP ranges for subnets.
   * Create two subnets within VNet2:
     + Subnet3: Name it and assign an appropriate IP range.
     + Subnet4: Name it and assign an appropriate IP range.

**2. Deploy Virtual Machines in Each Subnet**

1. **Deploy VM in Subnet1 (VNet1):**
   * Choose the VM OS (Windows or Linux) as per requirements.
   * Configure the VM settings including size, disk type, and networking (attach to Subnet1).
   * Repeat for both Windows and Linux VMs in Subnet1.
2. **Deploy VM in Subnet3 (VNet2):**
   * Similarly, deploy VMs in Subnet3 of VNet2, choosing appropriate OS and configurations.

**3. Configure Network Security Groups (NSGs)**

1. **Allow ICMP (Ping) Between VMs:**
   * Create NSGs for each subnet if not already created.
   * Configure inbound and outbound rules to allow ICMP (ping) traffic between VMs within the same VNet/subnet.

**4. Establish VNet Peering Between VNet1 and VNet2**

1. **Create VNet Peering:**
   * In the Azure portal, navigate to VNet1 > **Peerings** > **Add**.
   * Select VNet2 as the peer VNet, and configure peering settings (such as allowing gateway transit and forwarding, if needed).
2. **Verify Peering Connection:**
   * Ensure that peering status is 'Connected' and no issues are reported.

**5. Test Connectivity Between VMs**

1. **Ping Test:**
   * From a VM in Subnet1 (VNet1), ping VMs in Subnet1 (VNet1), Subnet2 (VNet1), Subnet3 (VNet2), and Subnet4 (VNet2).
   * Repeat from a VM in Subnet3 (VNet2) to VMs in Subnet1 (VNet1), Subnet2 (VNet1), Subnet3 (VNet2), and Subnet4 (VNet2).
   * All pings should be successful if NSGs and peering configurations are correctly set up.

**6. Documentation and Summary**

1. **Document Setup:**
   * Create a summary of the configuration steps followed.
   * Include screenshots of Azure portal configurations (VNets, subnets, VMs, NSGs, peering).
   * Document any troubleshooting steps taken and resolutions found during setup.
2. **Summary:**
   * Azure VNets and subnets were created with appropriate IP ranges.
   * Windows and Linux VMs were successfully deployed in respective subnets.
   * NSGs were configured to allow ICMP traffic for ping tests.
   * VNet peering was established between VNet1 and VNet2, enabling communication across VNets.
   * Connectivity tests confirmed that VMs in different subnets and VNets can ping each other.

**Conclusion:**

This document provides a structured approach to setting up Azure VNets, deploying VMs, configuring network security, establishing VNet peering, and ensuring connectivity between VMs across different subnets and VNets. Following these steps should enable seamless communication and networking within an Azure environment.