Name: Kriti  
College: Lovely Professional University  
Week1 Assignment  
Email: kritijulia@gmail.com

**Deploy Linux and Windows virtual machine and access them using SSH and RDP**

**Answers:** I created a new account on azure portal then click on create resources then created virtual machine using free versions then created and deployed linux virtual machine using ssh. For windows operating system used RDP. (Create a resource → Compute → Virtual machine )

**Configure Networking (Open Ports)**

**In the "Networking" tab:**

* **Select Virtual Network/Subnet** (default is okay)
* Enable **Public IP** (required for SSH/RDP access)
* Under **Inbound port rules**:
  + **Linux VM**:
    - Allow **SSH (TCP 22)**
  + **Windows VM**:
    - Allow **RDP (TCP 3389)**

**Connect to VMs**

**Linux via SSH**

After VM deployment:

1. Copy the public IP of your Linux VM.
2. Run the following command from your terminal:

ssh -i ~/Downloads/ubuntu\_key.pem azureuser@<Public-IP>

Replace <Public-IP> with my VM IP.

**Windows via RDP**

After VM deployment:

1. Go to Windows VM in the Azure portal.
2. Click on Connect → RDP → Download RDP File.
3. Open the downloaded .rdp file in Remote Desktop Connection.
4. Enter:
   * Username: azureuser
   * Password: the one you set

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

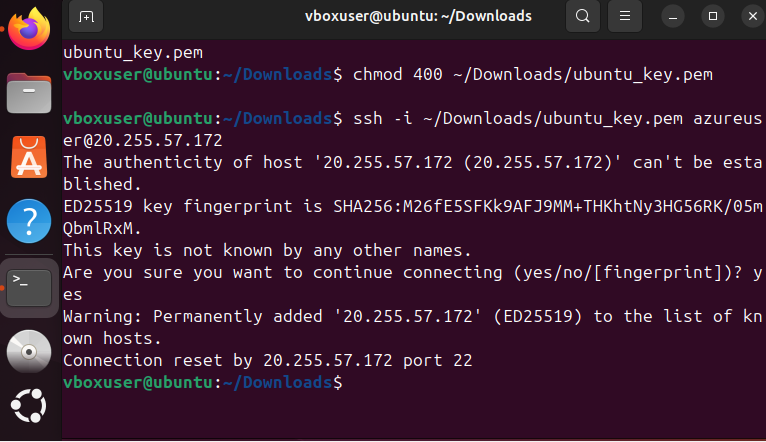
Created and deployed windows virtual machine using RDPA screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer program

AI-generated content may be incorrect. A screenshot of a chat

AI-generated content may be incorrect.

Connecting to azure virtual machine with my virtual machine



**Create an App Service Plan Provision a Web App in the existing App Service Plan and deploy a simple welcome page on it .**

**Answer:   
A. Creating a Virtual Network with 2 Subnets for VMs and Database**

**I started by creating a Virtual Network (VNet) to logically group and isolate Azure resources for communication.**

**🔧 Steps I Followed:**

1. **Go to Azure Portal → Search for Virtual Networks → Click Create.**
2. **In the Basics tab:**
   * **Subscription: Azure for Students**
   * **Resource group: vnet\_project\_group (I created a new one)**
   * **Name: MainVNet**
   * **Region: East Asia**
3. **In the IP Address tab:**
   * **Left the default IP range: 10.0.0.0/16**
4. **Then I created two subnets:**
   * **Subnet-1: VMSubnet → 10.0.1.0/24 (For Linux VM and Windows VM)**
   * **Subnet-2: DBSubnet → 10.0.2.0/24 (For SQL Database)**
5. **I reviewed and created the VNet.**

**🖥️ Deployed VMs in Subnet-1**

* **Linux VM: Named linuxvm, created in VMSubnet**
* **Windows VM: Named windowsvm, also in VMSubnet**

**During VM creation, in the Networking tab, I selected:**

* **Virtual Network: MainVNet**
* **Subnet: VMSubnet**

**🗃️ Planned for SQL DB in Subnet-2**

**Although I didn’t deploy a real SQL Server, I reserved DBSubnet to simulate a secure backend subnet that can later host:**

* **Azure SQL Managed Instance**
* **SQL Server VM (if required)**

**🧭 B. Hub-and-Spoke Network: 4 VNets + VM Communication Test**

**Now I created 4 VNets to design a Hub-and-Spoke architecture, a common Azure enterprise pattern for network security and control.**

**📌 VNets I Created:**

1. **Management VNet – HUB**
   * **Name: mgmt-vnet**
   * **Address space: 10.10.0.0/16**
   * **Subnet: mgmt-subnet → 10.10.1.0/24**
2. **Production VNet – Spoke**
   * **Name: prod-vnet**
   * **Address space: 10.20.0.0/16**
   * **Subnet: prod-subnet → 10.20.1.0/24**
3. **Testing VNet – Spoke**
   * **Name: test-vnet**
   * **Address space: 10.30.0.0/16**
   * **Subnet: test-subnet → 10.30.1.0/24**
4. **Development VNet – Spoke**
   * **Name: dev-vnet**
   * **Address space: 10.40.0.0/16**
   * **Subnet: dev-subnet → 10.40.1.0/24**

**🔄 Created Peering (Hub-and-Spoke Architecture)**

**Then I configured VNet peering between:**

* **mgmt-vnet ↔ prod-vnet**
* **mgmt-vnet ↔ test-vnet**
* **mgmt-vnet ↔ dev-vnet**

**➡️ This allows VMs in the management network to communicate with VMs in all spokes, but the spokes cannot communicate directly with each other (default behavior of hub-and-spoke).**

**Steps:**

1. **Go to mgmt-vnet → Peerings → Add**
2. **Peer to prod-vnet, check allow traffic both ways**
3. **Repeat for test-vnet and dev-vnet**

**💻 Launched VMs in Each VNet**

**I created 1 VM in each of the 4 VNets:**

* **vm-mgmt in mgmt-vnet**
* **vm-prod in prod-vnet**
* **vm-test in test-vnet**
* **vm-dev in dev-vnet**

**🧪 Verified Communication (Ping Test)**

**Then I logged in to the Management VM (vm-mgmt) using SSH and tested connectivity using ping:**

**bash**

**CopyEdit**

**ping 10.20.1.4 # vm-prod IP**

**ping 10.30.1.4 # vm-test IP**

**ping 10.40.1.4 # vm-dev IP**

**✅ Ping was successful to all 3 VMs, which verified that VNet peering and routing were working correctly.  
It also demonstrated that the Management VM could control and monitor all other VMs across different VNets, just like a real-world management setup.**