In this lab, you will learn how to provision a new Azure Virtual Network (VNet) using the Azure CLI. After you have the VNet provisioned, new subnets can be created under it. You can deploy Azure resources into these subnets, providing network traffic isolation and security. For example, an Azure network security group (NSG) can be assigned to a subnet to control ingress (incoming) and egress (outgoing) traffic.

Learning Objectives
In this lab, you will learn how to do the following:
Log in to the Azure CLI
Deploy a new VNet
Confirm that the new VNet was successfully created
Clean up the VNet

Login to azure :
Az login -u $username -p $password

Understanding Azure Virtual Networks
Azure Virtual Network is the foundation of private networks in Microsoft Azure. Many Azure services support deployment to a VNet.
Here are a few Azure resource types that support VNet deployment for more secure communication:

Azure Virtual Machines (it is mandatory to deploy this resource to a VNet)
Azure App Services (premium)
Azure Function Apps (premium)
Azure Cosmos DB
Azure Storage Accounts

In addition, deploying your resources to VNets enables you to protect them using Azure Firewall and Azure NSGs.
Similar to other computer networks, you need to specify an address range (a.k.a. address space) when creating a VNet. You also have the option to divide your VNet to different address spaces using subnets.

Address ranges (address spaces) are defined using the CIDR (Classless Inter-Domain Routing) format. For example, 10.0.0.0/24 means:

First IP address: 10.0.0.0
Last IP address: 10.0.0.255
Netmask: 255.255.255.0

Provision a New Azure Virtual Network
The following command deploys an Azure VNet to your allocated Azure resource group:

```
$ az network vnet create -g $resource --name $vnetName --address-prefix 10.0.0.0/24 --subnet-name subnet01 --subnet-prefix 10.0.0.0/28
{
  "newVNet": {
    "addressSpace": {
      "addressPrefixes": [
        "10.0.0.0/24"
      ]
    },
    "bgpCommunities": null,
    "ddosProtectionPlan": null,
    "dhcpOptions": {
      "dnsServers": []
    },
    "enableDdosProtection": false,
    "enableVmProtection": null,
    "encryption": null,
    "etag": "W/\"f63ffb22-b169-4a8a-b49d-ae1d9915872e\"",
    "extendedLocation": null,
    "flowTimeoutInMinutes": null,
    "id": "/subscriptions/a95a2b50-e96c-4ee4-8ef8-3147196dd85b/resourceGroups/user-zbwnanycdpqm/providers/Microsoft.Network/virtualNetw
    "ipAllocations": null,
```

Vnet is created in azure portal also . The command creates a new Azure Virtual Network with the CIDR address of 10.0.0.0/24 (255 addresses); it also creates a single subnet in it.

Here are a few parameters for this command:

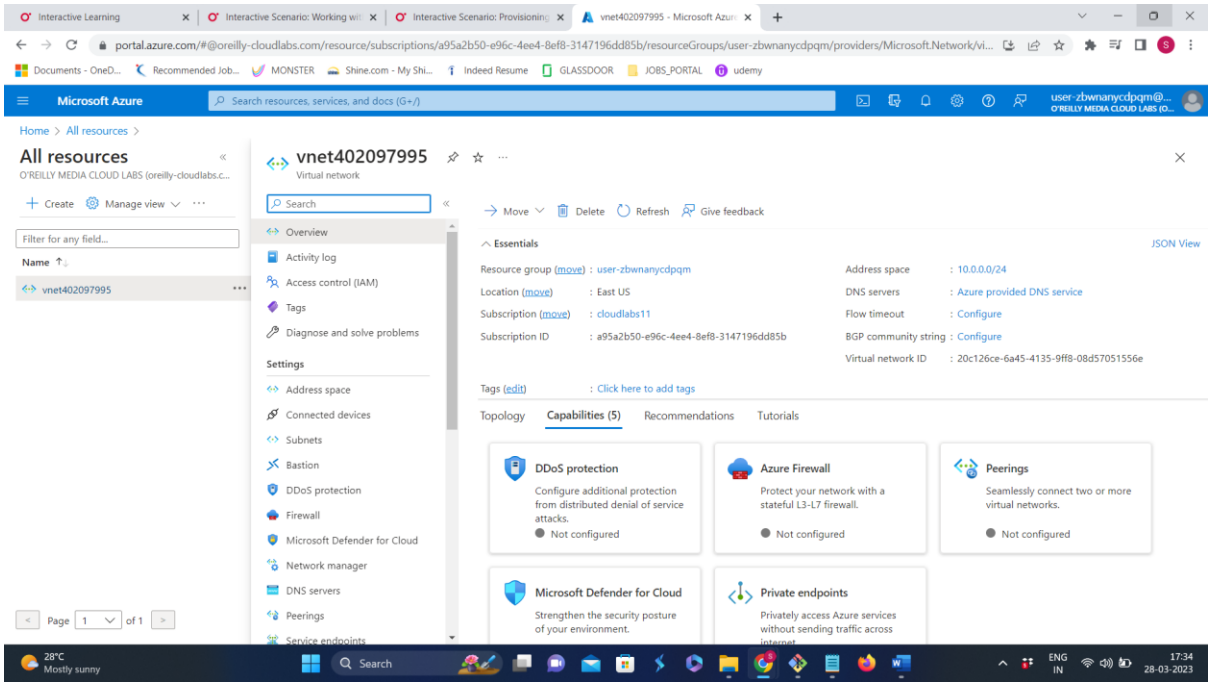
--resource-group: Name of the parent resource group

--name: Name of the new VNet

--address-prefix: List of the network IP addresses in CIDR (address space)

--subnet-name: Name of a new subnet to create within the VNet

--subnet-prefix: List of the subnet IP addresses (this should be a subset of the parent VNet address range)



Now that we've created your first Azure Virtual Network, we can confirm its provisioning in the next step.

Confirm the Azure Virtual Network is Created
It's time to see if we have created our VNet correctly. Use the following command to confirm that the new Azure VNet is created within your allocated resource group:

```
$ az network vnet show -g $resource --name $vnetName
{
  "addressSpace": {
    "addressPrefixes": [
      "10.0.0.0/24"
    ]
  },
  "bgpCommunities": null,
  "ddosProtectionPlan": null,
  "dhcpOptions": {
    "dnsServers": []
  },
  "enableDdosProtection": false,
  "enableVmProtection": null,
  "encryption": null,
  "etag": "W/\"f63ffb22-b169-4a8a-b49d-ae1d9915872e\"",
  "extendedLocation": null,
}
$ az network vnet show -g $resource --name $vnetName --query "{Name:name}"
{
  "Name": "vnet402097995"
}
$
```

Delete the Azure Virtual Network

Use the following command to clean up the VNet from your allocated resource group:

```
$ az network vnet delete --name $vnetName -g $resource
$
```

Use the following command to list all VNets in your allocated resource group:

```
$ az network vnet list -g $resource
[]
```