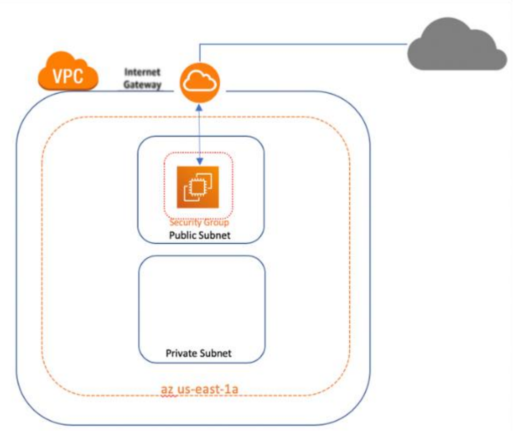
IT3061 – Massive Data Processing and Cloud Computing
Year 3, Semester 2
Practical Sheet 3

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

Build an Amazon Virtual Private Cloud (VPC)

- Here you are going to create a new virtual private cloud (VPC) and deploy an Amazon EC2 instance in the network you create.
- The *default* VPC will not be used in this practical. You will manually create your Amazon VPC, create and launch a webserver using a t-2 micro Amazon EC2 instance, and deploy your instance in the Amazon VPC you create.
- You will create the new Amazon VPC without using the Wizard tool or default VPC and availability zones Amazon Web Services (AWS) provides.
- Following are the steps that will be covered within the practical.
    - Create an Amazon VPC
    - Create route tables
    - Configure and associate the route tables
    - Create and attach an internet gateway (IGW)
    - Create and launch an Amazon EC2
    - Create a security group
    - Test your webpage

The diagram below shows the infrastructure you will build in this practical:



Create a virtual private cloud

An Amazon VPC is a virtual network logically isolated from other networks in the AWS Cloud. Follow these steps to get started:

1. In the **AWS Management Console,** find and select VPC within the Network and Content Delivery category.
2. On the **VPC Dashboard** page, find and select *Your VPC.*
3. Click and create a VPC with the following attributes:
    - **a. Name tag:** VPC_SLIIT1 VPC
    - **b. IPv4CIDR block:** 10.0.0.0/16
    - **c. Tenancy:** Default
4. Click **Create.**

Now that you have created your VPC_SLIIT1 VPC, let's create subnets, route tables, an internet gateway (IGW), and configure the subnets and routing tables accordingly.

Subtask: Subnets

Subnets contain logical groupings of resources and are often how you segment a network for security. A public subnet is a subnet that's associated with a route table that has a route to the internet via an internet gateway. Follow these steps to get started:

1. In the Virtual Private Cloud category of the VPC Dashboard, find and click on **Subnets**.
2. Click **Create subnet** and create a public subnet with the following attributes:

    a. **Name tag:** Public Subnet
    b. **VPC:** VPC_SLIIT1 VPC
    c. **Availability Zone:** us-east-1a
    d. **IPv4 CIDR block*:** 10.0.1.0/24

3. Click **Create**, then **Close**.

Create a private subnet by repeating Steps 1 through 3:

a. **Name tag:** Private Subnet
b. **VPC:** VPC_SLIIT1 VPC
c. **Availability Zone:** us-east-1a
d. **IPv4 CIDR block*:** 10.0.2.0/24

A private subnet is a subnet that's associated with a route table that allows communication of resources within your virtual private cloud and does not connect to the internet via an internet gateway. Resources in a private subnet are typically resources you want to keep secure from exposure to the internet. You will not be provisioning resources into the public subnet in this practical.

Create an internet gateway

An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between your VPC and the internet. An internet gateway serves two purposes: to provide a target in your VPC route tables for internet-routable traffic and to perform network address translation (NAT) for instances that have been assigned public IPv4 addresses. An internet gateway supports IPv4 and IPv6 traffic. It does not cause availability risks or bandwidth constraints on your network traffic. Essentially, the internet gateway connects your virtual private cloud to the internet. Follow these steps to get started:

1. In the Virtual Private Cloud category, click **Internet gateways** in the left sidebar.
2. Click **Create internet gateway.**
3. Enter a Name tag: VPC_SLIIT1 IGW
4. Click **Create internet gateway.**

Subtask: Attach your internet gateway to your VPC

The internet gateway has been created and now needs to be attached to your VPC.

1. In the Virtual Private Cloud category, click **Internet gateways** in the left sidebar.
2. Find your Internet Gateway VPC_SLIIT1 IGW and notice the state:
3. Select and highlight your internet gateway and go to **Actions** -> **Attach to VPC.**
4. In the available VPCs box, click and select the VPC_SLIIT1 VPC option from the list and click **attach internet gateway.**
5. Your IGW is now attached to your VPC_SLIIT1 VPC.

Create route tables

A *route table* contains a set of rules, called *routes*, that are used to determine where network traffic from your subnet or gateway is directed. Your VPC has an implicit router, and you use route tables to control where network traffic is directed. Each subnet in your VPC must be associated with a route table, which controls the routing for the subnet (subnet route table). You can explicitly associate a subnet with a particular route table. Otherwise, the subnet is implicitly associated with the main route table. A subnet can only be associated with one route table at a time, but you can associate multiple subnets with the same subnet route table.

Follow these steps to get started:

1. In the virtual private cloud category, find and click on **Route tables**.
2. Click the blue button **Create route table** and Create route table with **Name tag** *Public Route Table* and **VPC** *VPC_SLIIT1 VPC.*
3. Click **Create**, then **Close**.

Repeat the above three steps—this time to create your private route table. Use the following:

Name tag: *Private Route Table*

VPC: *VPC_SLIIT1 VPC*

Create route

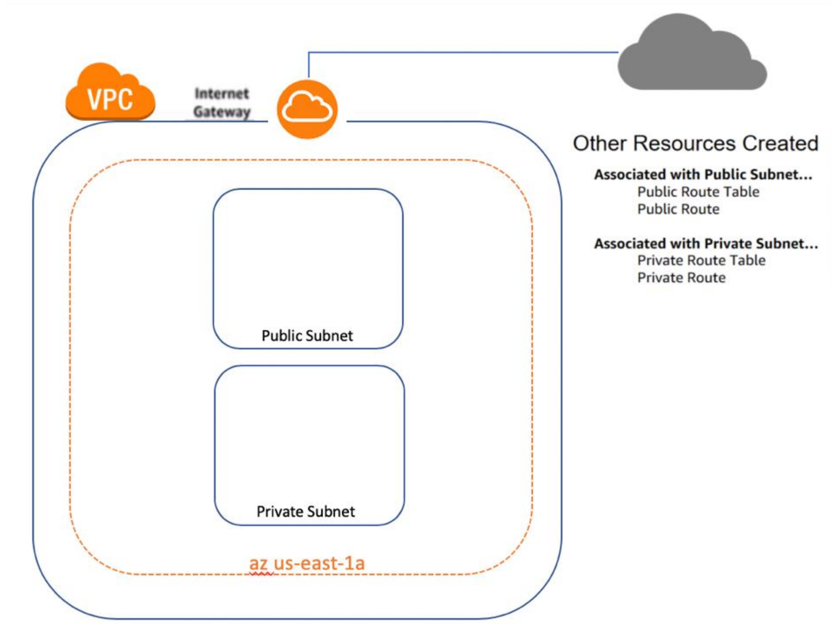
1. In the virtual private cloud category, page find and click on **Route tables**.
2. Locate and select the box next to your public route table.
3. Click on the **Routes** tab and notice the route is 10.0.0.0/16 and local. You need to add a route to the internet using the IGW.
4. Click on the **Edit routes** button then click on the **Add route** button.
5. Enter 0.0.0.0/0 in the **Destination** field and in the **Target** field, use the dropdown window and click on internet gateway. Locate your *VPC_SLIIT1* VPC IGW, select the VPC_SLIIT1 VPC IGW, and click **Save routes**. Now, click **Close**.
6. With your public route table still selected, find and click the **Subnet associations** tab near the bottom of the page.
7. Click on the **Edit subnet associations** button. *You might have to resize your column headers to read correctly*. Then, click and highlight your **Public subnet**, and click **Save.**

The route table with the route you created to the IGW is now associated with your public subnet. Your public subnet now has access to the internet.

Create a private route table

1. Repeat the above seven steps; this time editing your **private route table**. (*Note: You will not need to create a new route for your Private Route table, but you will need to associate the private route table with your private subnet.*)
2. Pay close attention and ensure you select us-east-1a for the AZ and the **Private subnet** options**.**

The diagram below shows the infrastructure you've created so far in this practical:



Create an Amazon EC2

1. In the **AWS Management Console**, find and select the Amazon EC2 dashboard.
2. From the **Amazon EC2 dashboard**, click **Launch instances.**
3. Notice the variety of AMIs located on the AMI page. These are different templates for different types of machines. Select the **Amazon Linux 2 AMI** (HVM).
4. Notice the variety of instance types available. Select the **t2.micro instance**.
5. Select **Next: Configure instance details.**
6. In the **Step 3: Configure instance details** page, you need to configure the following settings:

   Network: VPC_SLIIT1 VPC

   Subnet: Public subnet | us-east 1a

   Auto-Assign Public IP: Enable

Scroll to the bottom of the page and locate the **Advanced details** section and expand if necessary. With the advanced details section expanded, insert the following bash script within the **User data** section:

```
#!/bin/bash
yum -y install httpd
systemctl enable httpd
systemctl start httpd
echo '<html><h1>Hello !!!, This is my first
VPC !!! </h1></html>' > /var/www/html/index.html
```

1. Click **Next: Add storage.** You will not need another Amazon Elastic Block Store (Amazon EBS) volume.
2. Click **Next: Add tags.**
3. Click **Add tag.** Then, configure:

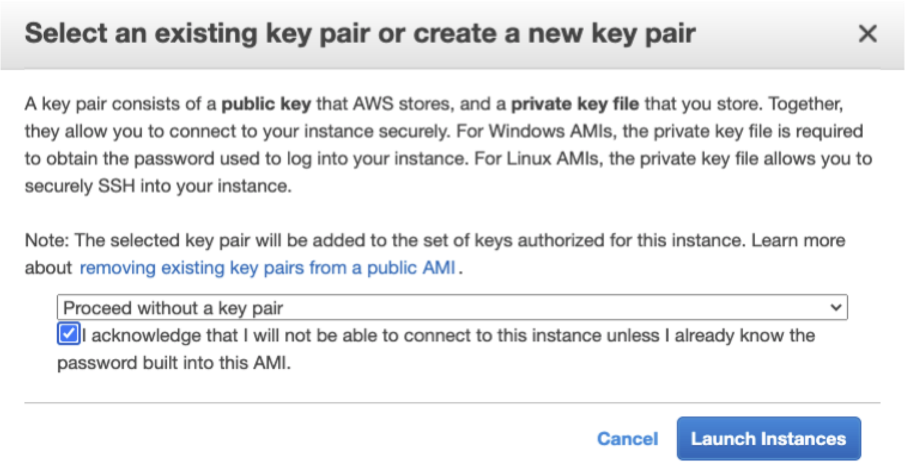
    a. **Key:** *Name* **Value**: *Testing Server*
    b. **Key:** *Department* **Value**: *Development*

4. Click **Next: Configure security group.**
5. Configure a *new* security group as follows:

    i. **Security Group Name**: SSH and HTTP SG.
    ii. **Description:** This security group allows for SSH and HTTP.
    iii. By default, the Type SSH with Port 22 has been added.
    iv. Click the **Add rule** button and locate HTTP under the **Type** header. Then, change Custom to *Anywhere* under the **Source** heading.
    v. Click **Review and launch**.

12. Review the details, scroll down, and click **Launch**.

This is an example snapshot from the AWS Management Console:



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Proceed without a key pair

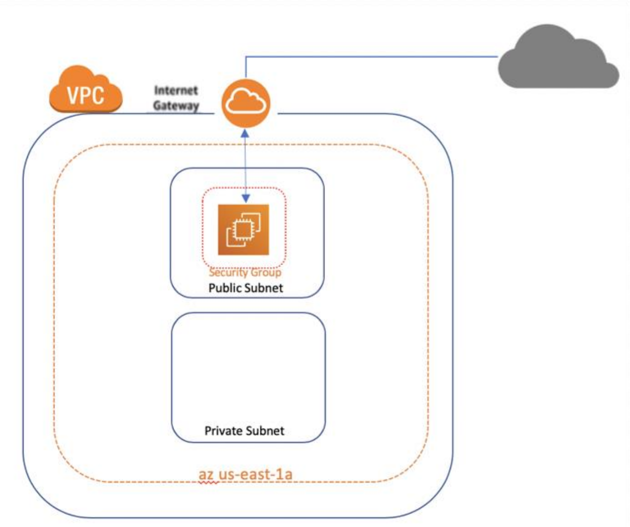
☑ I acknowledge that I will not be able to connect to this instance unless I already know the password built into this AMI.

13. Click on **View instances** or navigate to the **Instances** category within the Amazon EC2 dashboard page.

Test your webpage

1. Select your **Amazon EC2 server** Instance and copy the **IPV4 public IP** address to your clipboard.
2. Open a new browser tab, paste the **public IP** address into a new browser window, and observe the results.
3. You should see **Hello !!!, This is my first VPC !** in your browser.

This is a diagram of the infrastructure you've just built in this practical:



Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. VNet enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks. VNet is similar to a traditional network that you'd operate in your own data center, but brings with it additional benefits of Azure's infrastructure such as scale, availability, and isolation.

Build an Azure virtual network (VNet)

In this practical, you learn how to create a virtual network using the Azure portal. You deploy two virtual machines (VMs). Next, you securely communicate between VMs and connect to VMs from the internet. A virtual network is the fundamental building block for your private network in Azure. It enables Azure resources, like VMs, to securely communicate with each other and with the internet.

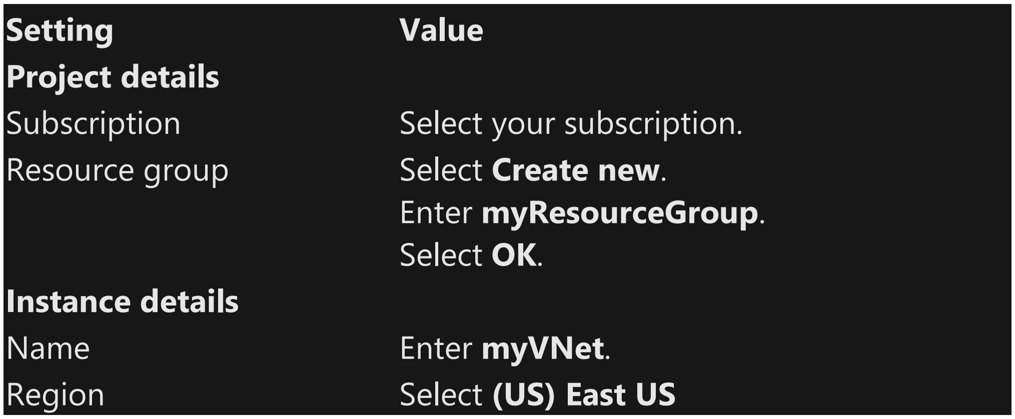
Create a virtual network

1. Select Create a resource in the upper left-hand corner of the portal.

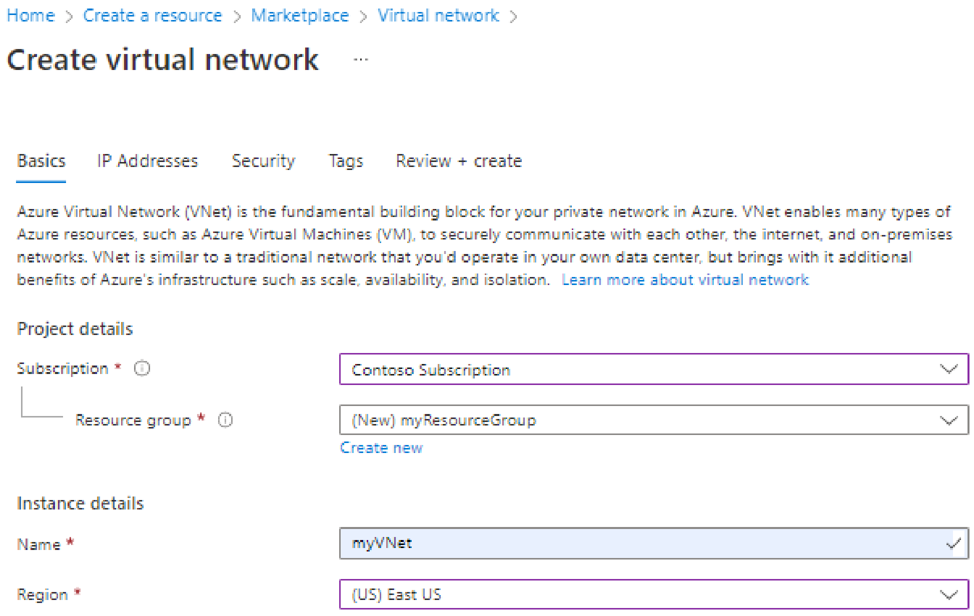
2. In the search box, enter Virtual Network. Select Virtual Network in the search results.

3. In the Virtual Network page, select Create.

4. In Create virtual network, enter or select this information in the Basics tab:



Setting	**Value**
Project details	
Subscription	Select your subscription.
Resource group	Select **Create new**. Enter **myResourceGroup**. Select **OK**.
Instance details	
Name	Enter **myVNet**.
Region	Select **(US) East US**



Home > Create a resource > Marketplace > Virtual network >

Create virtual network ···

Basics IP Addresses Security Tags Review + create

Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. VNet enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks. VNet is similar to a traditional network that you'd operate in your own data center, but brings with it additional benefits of Azure's infrastructure such as scale, availability, and isolation. Learn more about virtual network

Project details

Subscription * : Contoso Subscription

Resource group * : (New) myResourceGroup

Create new

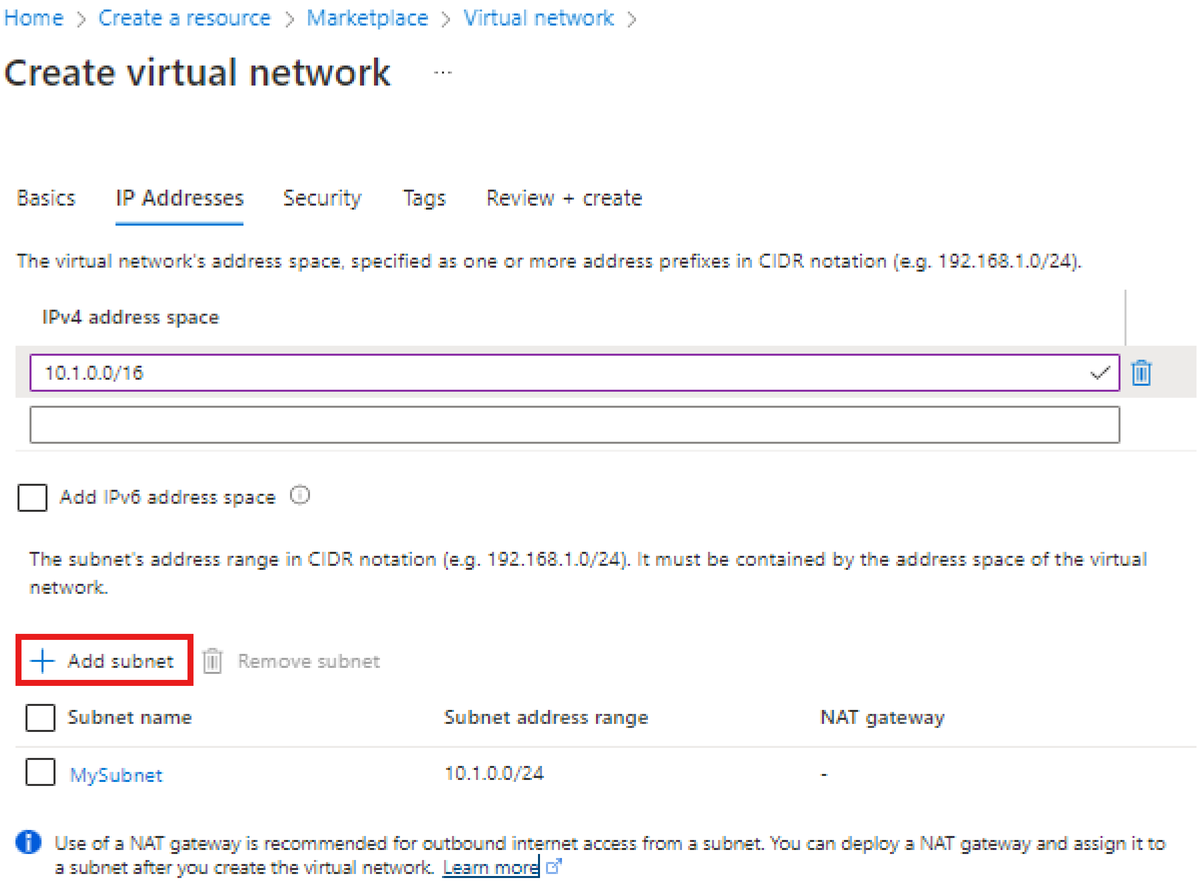
Instance details

Name * : myVNet

Region * : (US) East US

5. Select the IP Addresses tab, or select the Next: IP Addresses button at the bottom of the page and enter in the following information then select Add:

Setting	Value
IPv4 address space	Enter **10.1.0.0/16**.
Add subnet	
Subnet name	Enter **MySubnet**.
Subnet address range	Enter **10.1.0.0/24**.
Select **Add**.	


Home > Create a resource > Marketplace > Virtual network >
Create virtual network ...
Basics
IP Addresses
Security
Tags
Review + create
+ Add subnet
Remove subnet
Learn more

The virtual network's address space, specified as one or more address prefixes in CIDR notation (e.g. 192.168.1.0/24).

IPv4 address space

10.1.0.0/16

□ Add IPv6 address space

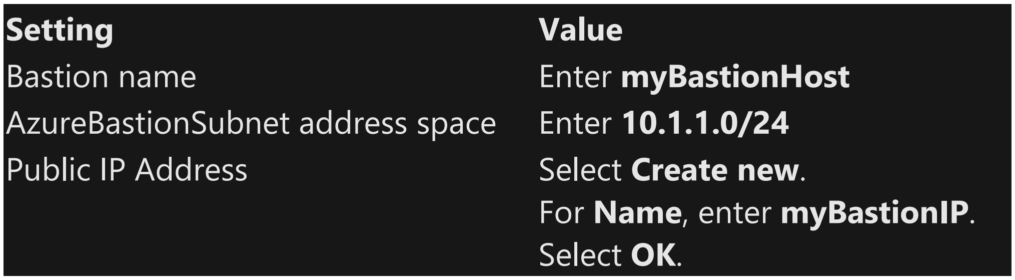
The subnet's address range in CIDR notation (e.g. 192.168.1.0/24). It must be contained by the address space of the virtual network.

□ Subnet name	Subnet address range	NAT gateway
□ MySubnet	10.1.0.0/24	-

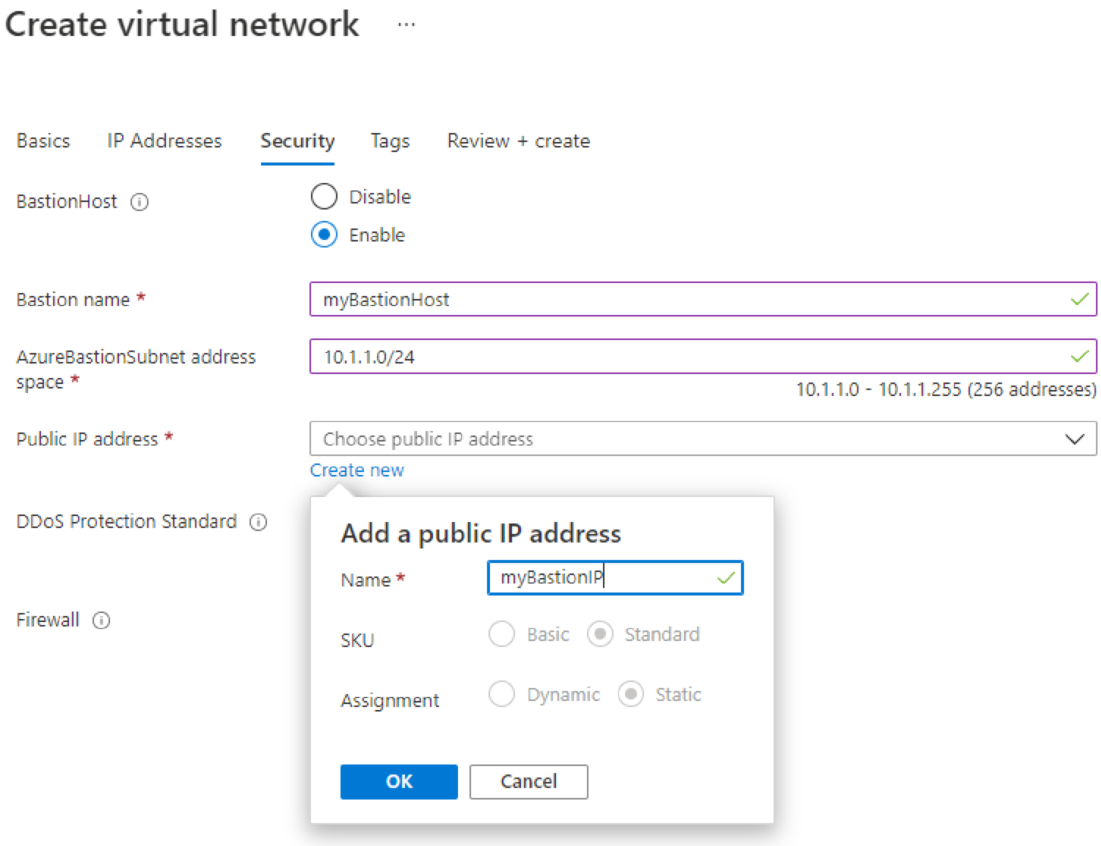
Use of a NAT gateway is recommended for outbound internet access from a subnet. You can deploy a NAT gateway and assign it to a subnet after you create the virtual network. Learn more

6. Select the Security tab, or select the Next: Security button at the bottom of the page.

7. Under BastionHost, select Enable. Enter this information:



Setting	Value
Bastion name	Enter **myBastionHost**
AzureBastionSubnet address space	Enter **10.1.1.0/24**
Public IP Address	Select **Create new**. For **Name**, enter **myBastionIP**. Select **OK**.



Create virtual network

BastionHost: ○ Disable ● Enable

Bastion name *: myBastionHost

AzureBastionSubnet address space *: 10.1.1.0/24

10.1.1.0 - 10.1.1.255 (256 addresses)

Public IP address *: Choose public IP address

Add a public IP address

Name *: myBastionIP

SKU: ○ Basic ● Standard

Assignment: ○ Dynamic ● Static

8. Select the Review + create tab or select the Review + create button.

9. Select Create.

Create virtual machines

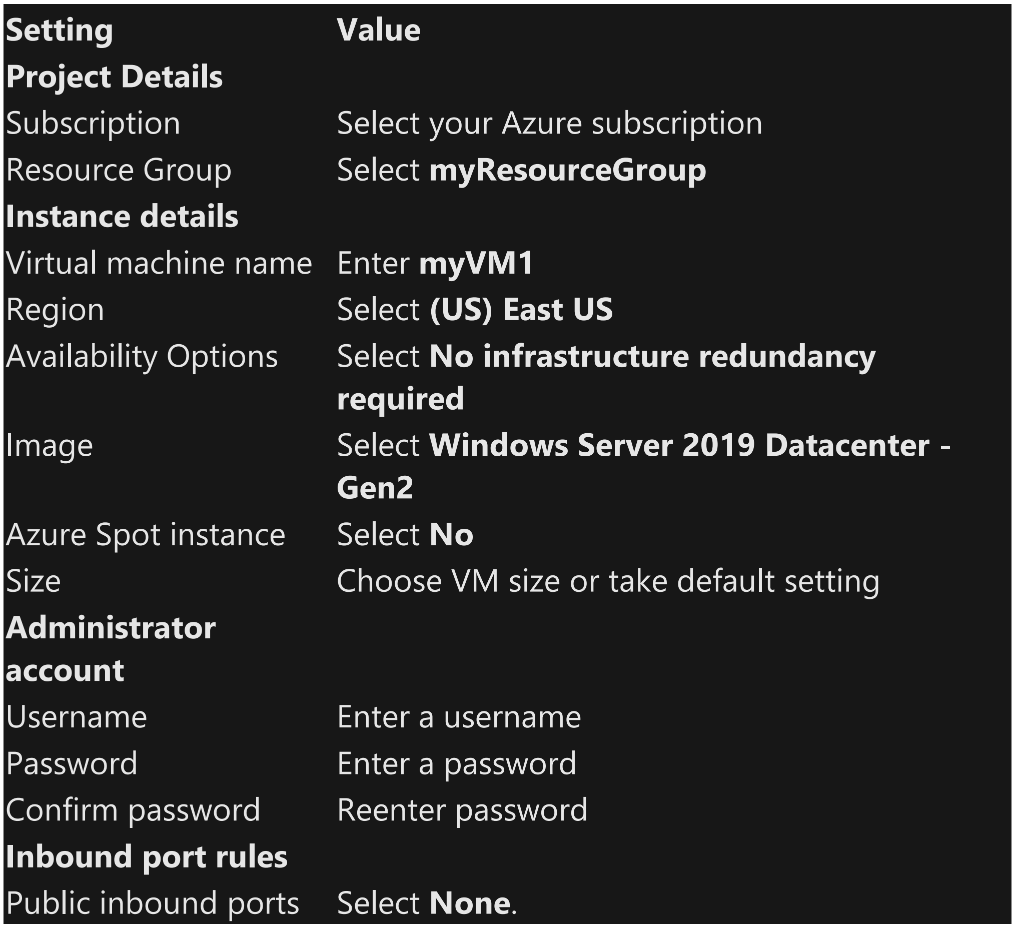
Create two VMs in the virtual network as follows.

1. On the upper-left side of the portal, select Create a resource > Compute > Virtual machine.

2. In Create a virtual machine, type or select the values as mentioned below separately for two virtual machines.

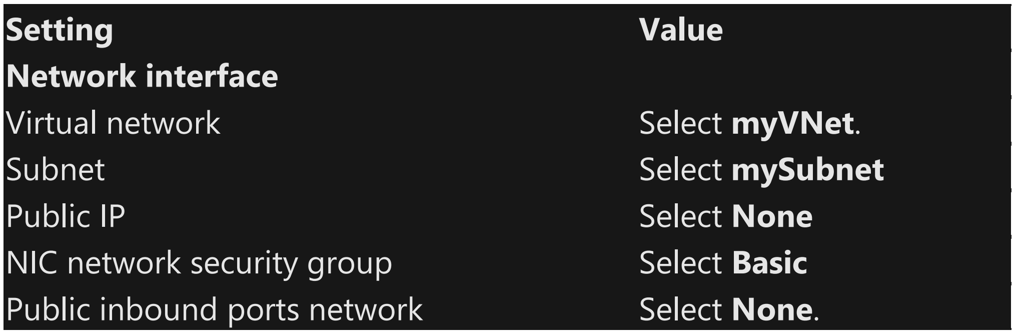
Virtual Machine 1

Basics tab



Setting	Value
Project Details	
Subscription	Select your Azure subscription
Resource Group	Select **myResourceGroup**
Instance details	
Virtual machine name	Enter **myVM1**
Region	Select **(US) East US**
Availability Options	Select **No infrastructure redundancy required**
Image	Select **Windows Server 2019 Datacenter - Gen2**
Azure Spot instance	Select **No**
Size	Choose VM size or take default setting
Administrator account	
Username	Enter a username
Password	Enter a password
Confirm password	Reenter password
Inbound port rules	
Public inbound ports	Select **None**.

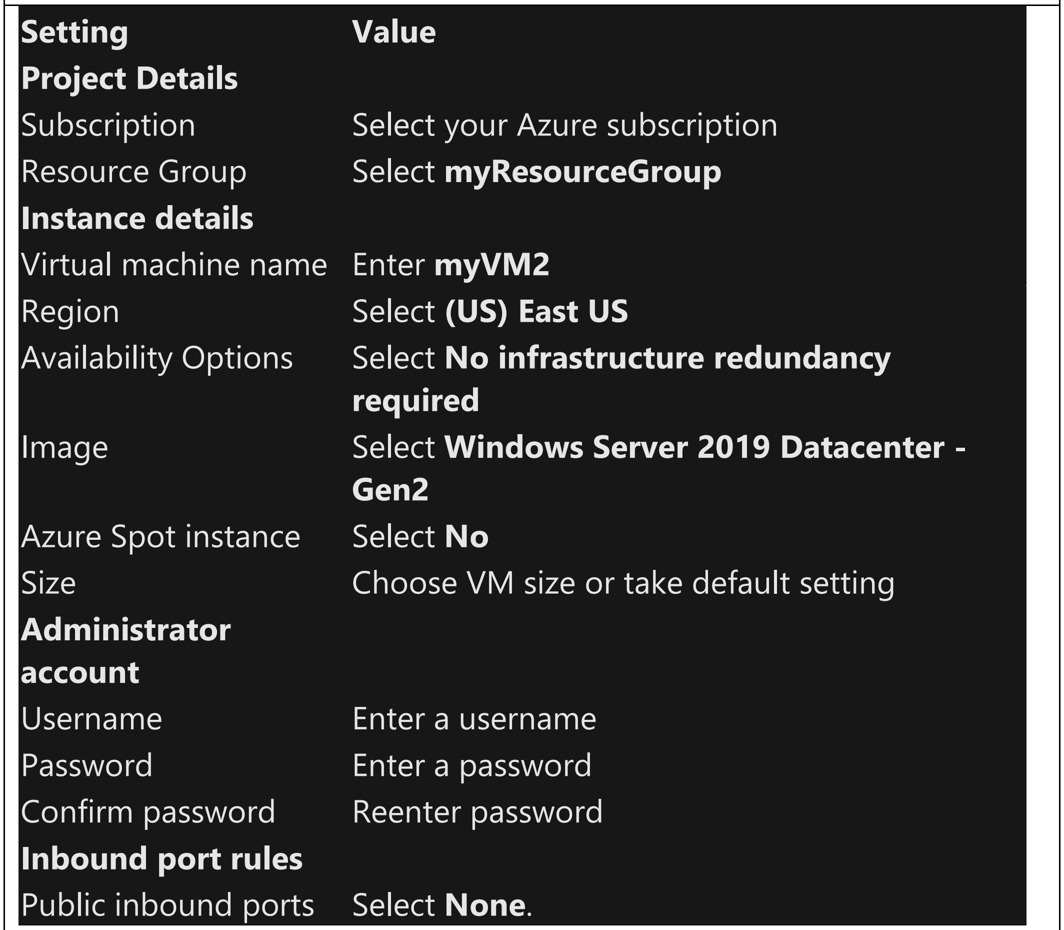
Networking tab



Setting	Value
Network interface	
Virtual network	Select **myVNet**.
Subnet	Select **mySubnet**
Public IP	Select **None**
NIC network security group	Select **Basic**
Public inbound ports network	Select **None**.

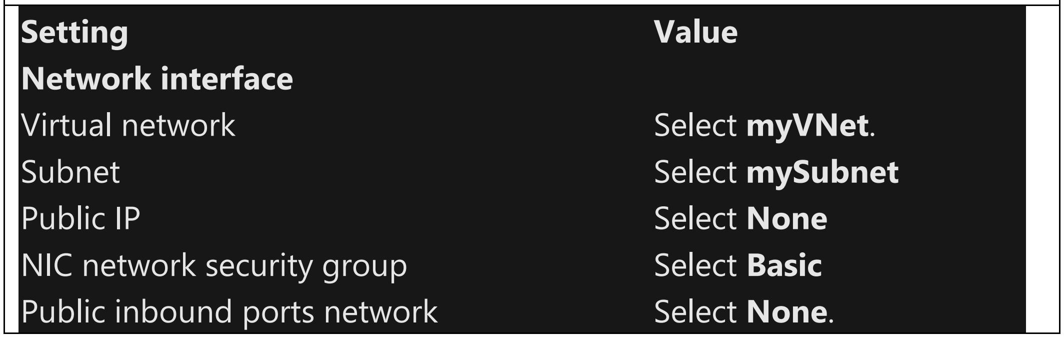
Virtual Machine 2

Basic tab



Setting	Value
Project Details	
Subscription	Select your Azure subscription
Resource Group	Select **myResourceGroup**
Instance details	
Virtual machine name	Enter **myVM2**
Region	Select **(US) East US**
Availability Options	Select **No infrastructure redundancy required**
Image	Select **Windows Server 2019 Datacenter - Gen2**
Azure Spot instance	Select **No**
Size	Choose VM size or take default setting
Administrator account	
Username	Enter a username
Password	Enter a password
Confirm password	Reenter password
Inbound port rules	
Public inbound ports	Select **None**.

Networking tab

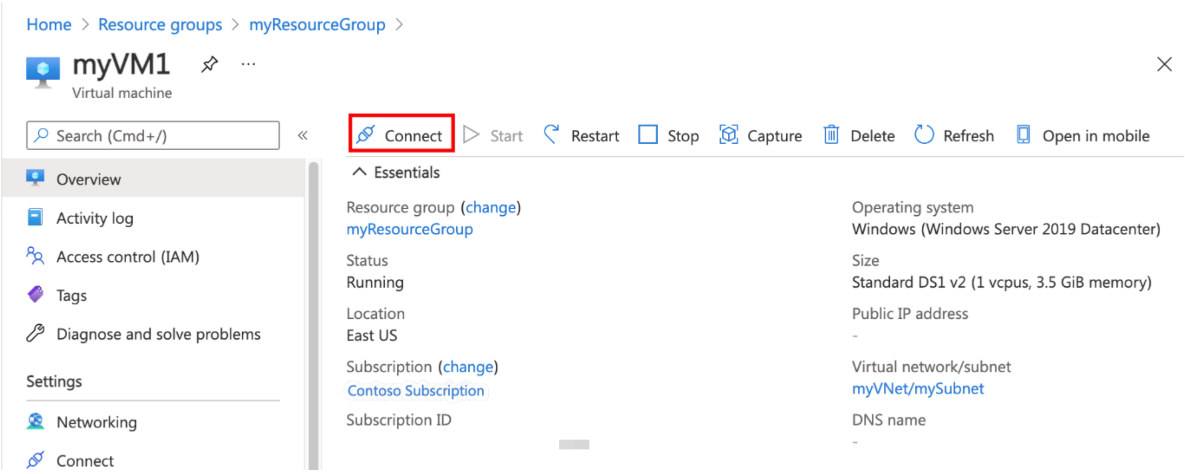


Setting	Value
Network interface	
Virtual network	Select **myVNet**.
Subnet	Select **mySubnet**
Public IP	Select **None**
NIC network security group	Select **Basic**
Public inbound ports network	Select **None**.

Connect to myVM1

1. Go to the Azure portal to manage your private VM. Search for and select Virtual machines.
2. Pick the name of your private virtual machine myVM1.

3. In the VM menu bar, select Connect, then select Bastion.



Resource group (change)
myResourceGroup

Status
Running

Location
East US

Subscription (change)
Contoso Subscription

Subscription ID

Operating system
Windows (Windows Server 2019 Datacenter)

Size
Standard DS1 v2 (1 vcpus, 3.5 GiB memory)

Public IP address
-

Virtual network/subnet
myVNet/mySubnet

DNS name
-

4. In the Connect page, select the blue Use Bastion button.

5. In the Bastion page, enter the username and password you created for the virtual machine previously.

6. Select Connect.

Communicate between VMs

1. In the Bastion connection of myVM1, open PowerShell.

2. Enter ping myVM2.

   You'll get a reply message like this:

```
PS C:\Users\myVM1> ping myVM2

Pinging myVM2.ovvzzdcazhbu5iczfvonhg2zrb.bx.internal.cloudapp.net
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.5:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

   The ping fails, because it uses the Internet Control Message Protocol (ICMP). By default, ICMP isn't allowed through your Windows firewall.

3. To allow myVM2 to ping myVM1 in a later step, enter this command:

   ```
   New-NetFirewallRule –DisplayName "Allow ICMPv4-In" –Protocol ICMPv4
   ```

   That command lets ICMP inbound through the Windows firewall.

4. Close the bastion connection to myVM1.

5. Complete the steps in Connect to myVM1, but connect to myVM2.

6. Open PowerShell on myVM2, enter ping myvm1.

   You'll receive a successful reply message like this:

   ```
   Pinging myVM1.cs4wv3rxdjgedggsfghkjrxuqf.bx.internal.cloudapp.net [10.1.0.4] with 32 bytes of data:
   Reply from 10.1.0.4: bytes=32 time=1ms TTL=128
   Reply from 10.1.0.4: bytes=32 time=1ms TTL=128
   Reply from 10.1.0.4: bytes=32 time=1ms TTL=128
   Reply from 10.1.0.4: bytes=32 time=1ms TTL=128

   Ping statistics for 10.1.0.4:
       Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
   Approximate round trip times in milli-seconds:
       Minimum = 1ms, Maximum = 1ms, Average = 1ms
   ```

7. Close the bastion connection to myVM2.

References

1. https://docs.aws.amazon.com/vpc/latest/userguide/vpc-getting-started.html
2. https://docs.microsoft.com/en-us/azure/virtual-network/quick-create-portal#create-a-virtual-network