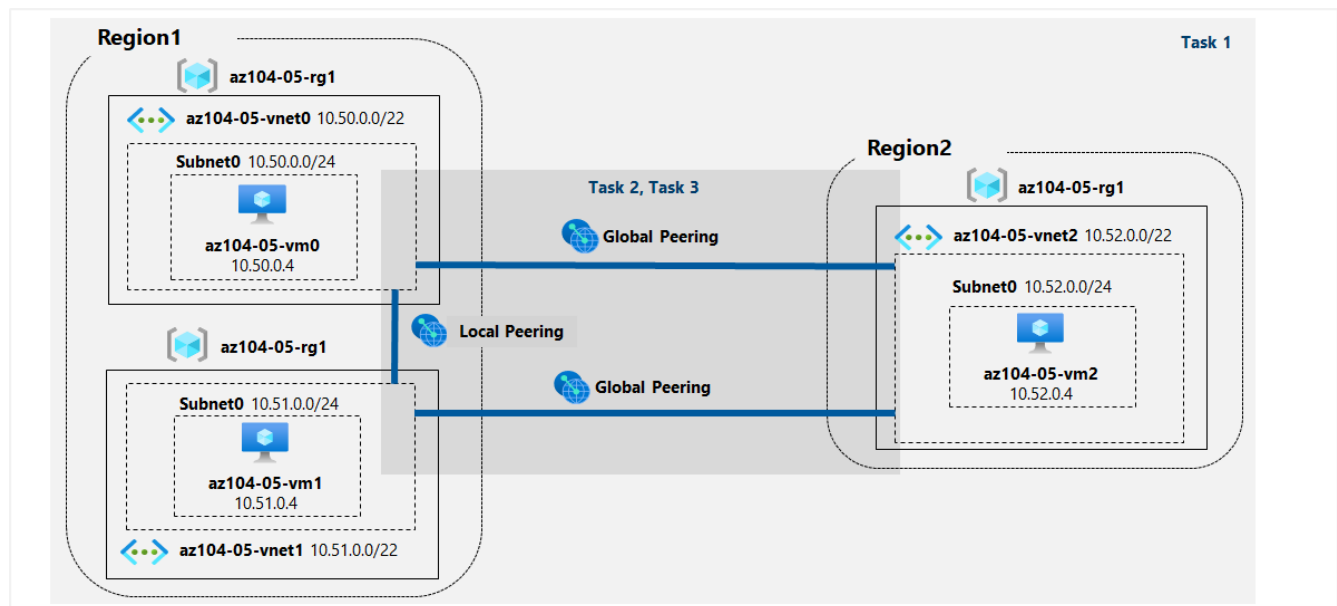


Lab scenario

Contoso has its datacenters in Boston, New York, and Seattle offices connected via a mesh wide-area network links, with full connectivity between them. You need to implement a lab environment that will reflect the topology of the Contoso's on-premises networks and verify its functionality.

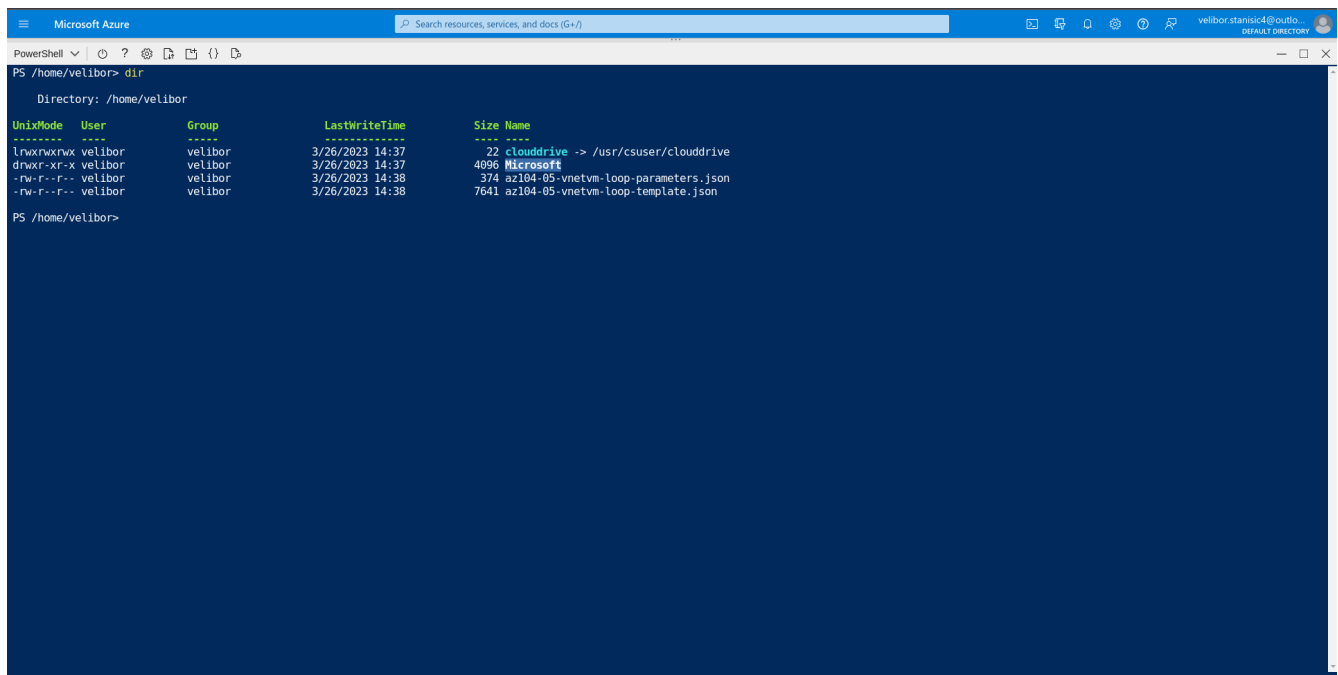
Architecture diagram



Task 1: Provision the lab environment

In this task, you will deploy three virtual machines, each into a separate virtual network, with two of them in the same Azure region and the third one in another Azure region.

4. In the toolbar of the Cloud Shell pane, click the **Upload/Download files** icon, in the drop-down menu, click **Upload** and upload the files `\Allfiles\Labs\05\az104-05-vnetvm-loop-template.json` and `\Allfiles\Labs\05\az104-05-vnetvm-loop-parameters.json` into the Cloud Shell home directory.



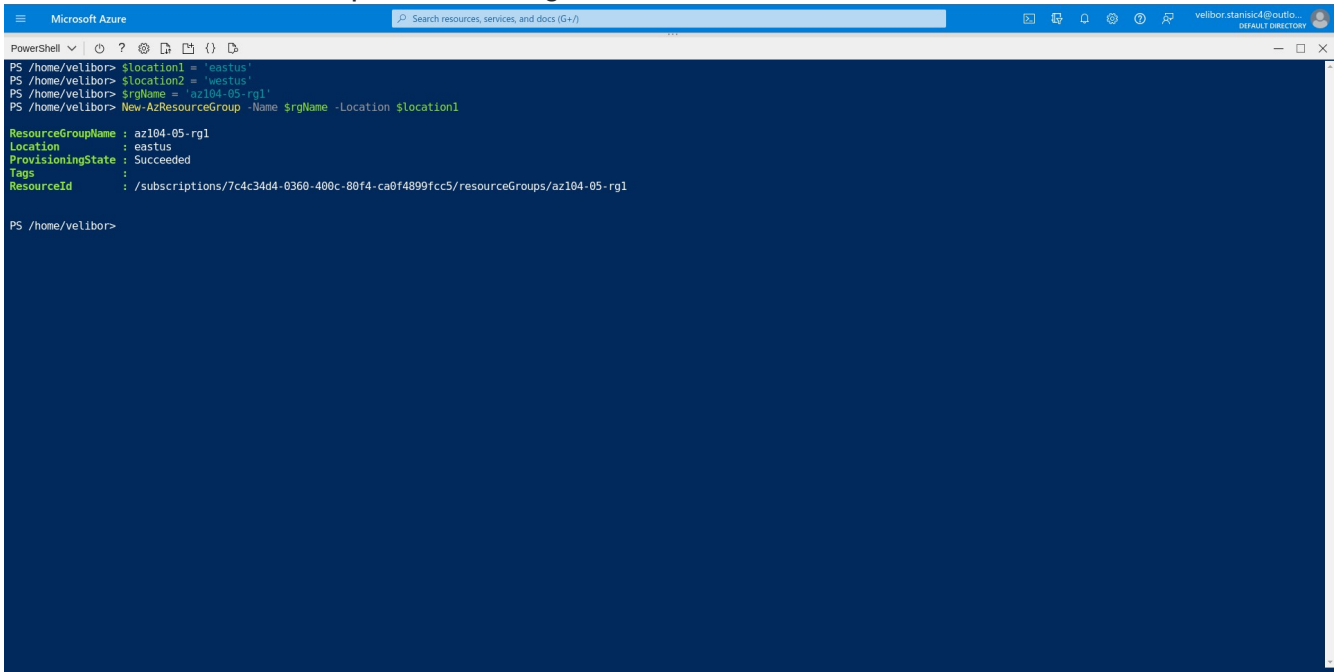
The screenshot shows the Microsoft Azure Cloud Shell interface. The top bar includes the 'Microsoft Azure' logo, a search bar, and a user profile icon. The main area is a PowerShell terminal window with a dark blue background. The prompt is 'PS /home/velibor>'. The user has entered the command 'dir', and the output shows the directory listing for '/home/velibor'. The output is formatted as a table with columns for UnixMode, User, Group, LastWriteTime, and Size Name.

UnixMode	User	Group	LastWriteTime	Size	Name
drwxrwxrwx	velibor	velibor	3/26/2023 14:37	22	clouddrive -> /usr/cuser/clouddrive
drwxr-xr-x	velibor	velibor	3/26/2023 14:37	4096	Microsoft
-rw-r--r--	velibor	velibor	3/26/2023 14:38	374	az104-05-vnetvm-loop-parameters.json
-rw-r--r--	velibor	velibor	3/26/2023 14:38	7641	az104-05-vnetvm-loop-template.json

The prompt is now 'PS /home/velibor>'.

6. From the Cloud Shell pane, run the following to create the resource group that will be hosting the lab environment. The first two virtual networks and a pair of virtual machines will be deployed in [Azure_region_1]. The third virtual network and the third virtual machine will be deployed in the same resource group but another [Azure_region_2]. (replace the [Azure_region_1] and [Azure_region_2] placeholder, including the square brackets, with the names of two different Azure regions where you intend to deploy these Azure virtual machines. An example is \$location1 = 'eastus'. You can use Get-AzLocation to list all locations.):

```
$location1 = 'eastus'  
$location2 = 'westus'  
$rgName = 'az104-05-rg1'  
New-AzResourceGroup -Name $rgName -Location $location1
```



The screenshot shows a PowerShell terminal window within the Microsoft Azure Cloud Shell interface. The user has executed four commands: setting \$location1 to 'eastus', setting \$location2 to 'westus', setting \$rgName to 'az104-05-rg1', and running New-AzResourceGroup with the specified name and location. The output shows the resource group 'az104-05-rg1' was created in the 'eastus' location, with a provisioning state of 'Succeeded'. The ResourceId is displayed as '/subscriptions/7c4c34d4-0360-408c-80f4-ca0f4899fcc5/resourceGroups/az104-05-rg1'.

```
PS /home/velibor> $location1 = 'eastus'  
PS /home/velibor> $location2 = 'westus'  
PS /home/velibor> $rgName = 'az104-05-rg1'  
PS /home/velibor> New-AzResourceGroup -Name $rgName -Location $location1  
  
ResourceGroupName : az104-05-rg1  
Location           : eastus  
ProvisioningState   : Succeeded  
Tags               :  
ResourceId          : /subscriptions/7c4c34d4-0360-408c-80f4-ca0f4899fcc5/resourceGroups/az104-05-rg1  
  
PS /home/velibor>
```

7. From the Cloud Shell pane, run the following to create the three virtual networks and deploy virtual machines into them by using the template and parameter files you uploaded:

New-AzResourceGroupDeployment `

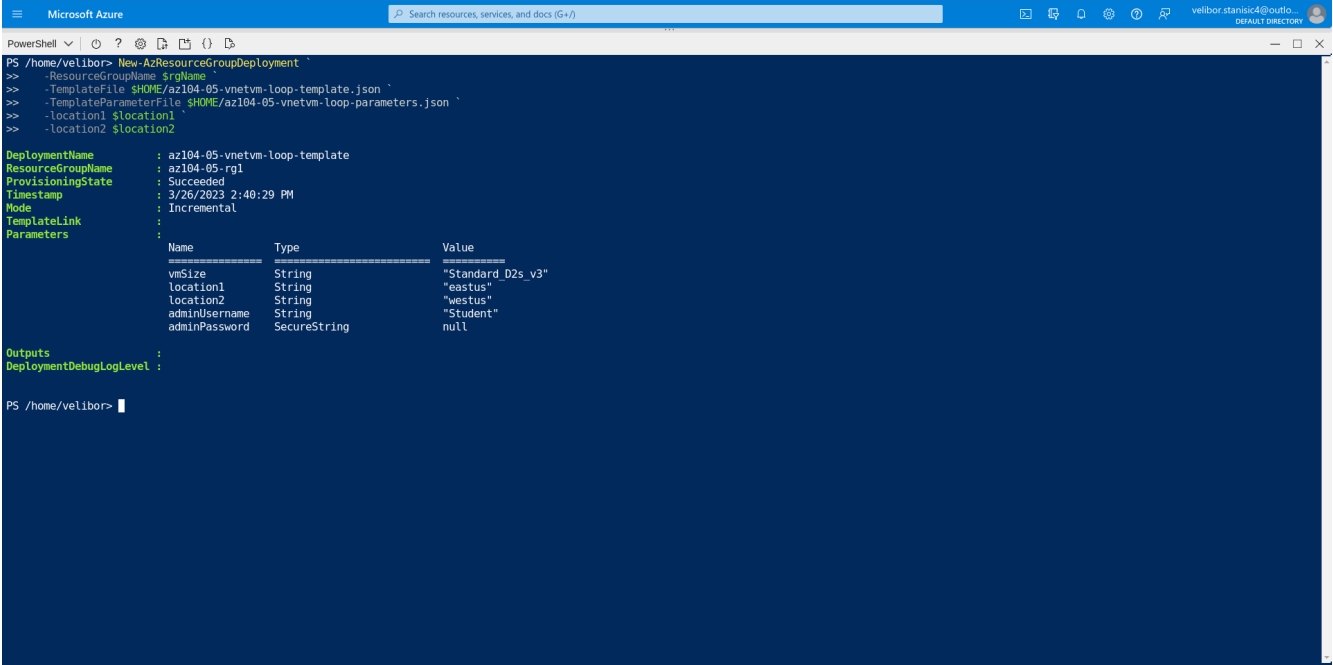
-ResourceGroupName \$rgName `

-TemplateFile \$HOME/az104-05-vnetvm-loop-template.json `

-TemplateParameterFile \$HOME/az104-05-vnetvm-loop-parameters.json `

-location1 \$location1 `

-location2 \$location2



```
Microsoft Azure
Search resources, services, and docs (G+I)
veibor.stanisic@outl...
DEFAULT DIRECTORY

PowerShell
PS /home/veibor> New-AzResourceGroupDeployment `
>> -ResourceGroupName $rgName `
>> -TemplateFile $HOME/az104-05-vnetvm-loop-template.json `
>> -TemplateParameterFile $HOME/az104-05-vnetvm-loop-parameters.json `
>> -location1 $location1 `
>> -location2 $location2

DeploymentName      : az104-05-vnetvm-loop-template
ResourceGroupName   : az104-05-rg1
ProvisioningState    : Succeeded
Timestamp           : 3/26/2023 2:40:29 PM
Mode                : Incremental
TemplateLink        :
Parameters
  Name      Type      Value
  -----
  vmSize    String    "Standard_D2s_v3"
  location1 String    "eastus"
  location2 String    "westus"
  adminUsername String    "Student"
  adminPassword SecureString null

Outputs
DeploymentDebugLogLevel :

PS /home/veibor>
```

Task 2: Configure local and global virtual network peering

In this task, you will configure local and global peering between the virtual networks you deployed in the previous tasks.

2. Review the virtual networks you created in the previous task and verify that the first two are located in the same Azure region and the third one in a different Azure region.

Microsoft Azure

Search resources, services, and docs (G+I)

Home >

Virtual networks

Default Directory

Create

Manage view

Refresh

Export to CSV

Open query

Assign tags

Filter for any field...

Subscription equals all

Resource group equals all

Location equals all

Add filter

No grouping

List view

Showing 1 to 3 of 3 records.

<input type="checkbox"/>	Name ↑↓	Resource group ↑↓	Location ↑↓	Subscription ↑↓	
<input type="checkbox"/>	az104-05-vnet0	az104-05-rg1	East US	Azure Pass - Sponsorship	...
<input type="checkbox"/>	az104-05-vnet1	az104-05-rg1	East US	Azure Pass - Sponsorship	...
<input type="checkbox"/>	az104-05-vnet2	az104-05-rg1	West US	Azure Pass - Sponsorship	...

< PreviousPage 1 of 1Next >

Give feedback

4. On the **az104-05-vnet0** virtual network blade, in the **Settings** section, click **Peerings** and then click **+ Add**.

5. Add a peering with the following settings (leave others with their default values) and click **Add**:

6. On the **az104-05-vnet0** virtual network blade, in the **Settings** section, click **Peerings** and then click **+ Add**.

7. Add a peering with the following settings (leave others with their default values) and click **Add**:

9. On the **az104-05-vnet1** virtual network blade, in the **Settings** section, click **Peerings** and then click **+ Add**.

10. Add a peering with the following settings (leave others with their default values) and click **Add**:

The screenshot shows the Microsoft Azure portal interface for the virtual network **az104-05-vnet0**. The left-hand navigation pane is expanded to the **Settings** section, with **Peerings** selected. The main content area displays a table of peerings. The table has columns for **Name**, **Peering status**, **Peer**, and **Gateway transit**. There are two peerings listed, both with a status of **Connected**.

Name	Peering status	Peer	Gateway transit
az104-05-vnet0_to_az104-05-vnet1	Connected	az104-05-vnet1	Disabled
az104-05-vnet0_to_az104-05-vnet2	Connected	az104-05-vnet2	Disabled

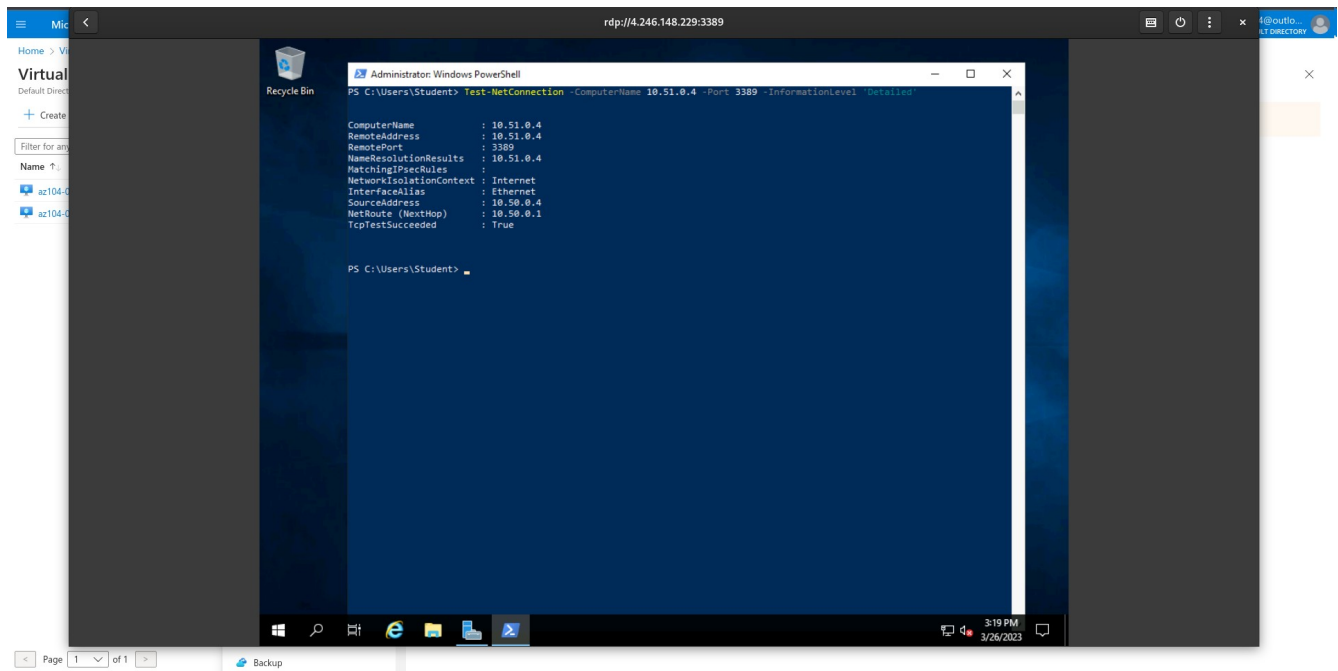
The screenshot shows the Microsoft Azure portal interface for the virtual network **az104-05-vnet1**. The left-hand navigation pane is expanded to the **Settings** section, with **Peerings** selected. The main content area displays a table of peerings. The table has columns for **Name**, **Peering status**, **Peer**, and **Gateway transit**. There are two peerings listed, both with a status of **Connected**.

Name	Peering status	Peer	Gateway transit
az104-05-vnet1_to_az104-05-vnet0	Connected	az104-05-vnet0	Disabled
az104-05-vnet1_to_az104-05-vnet2	Connected	az104-05-vnet2	Disabled

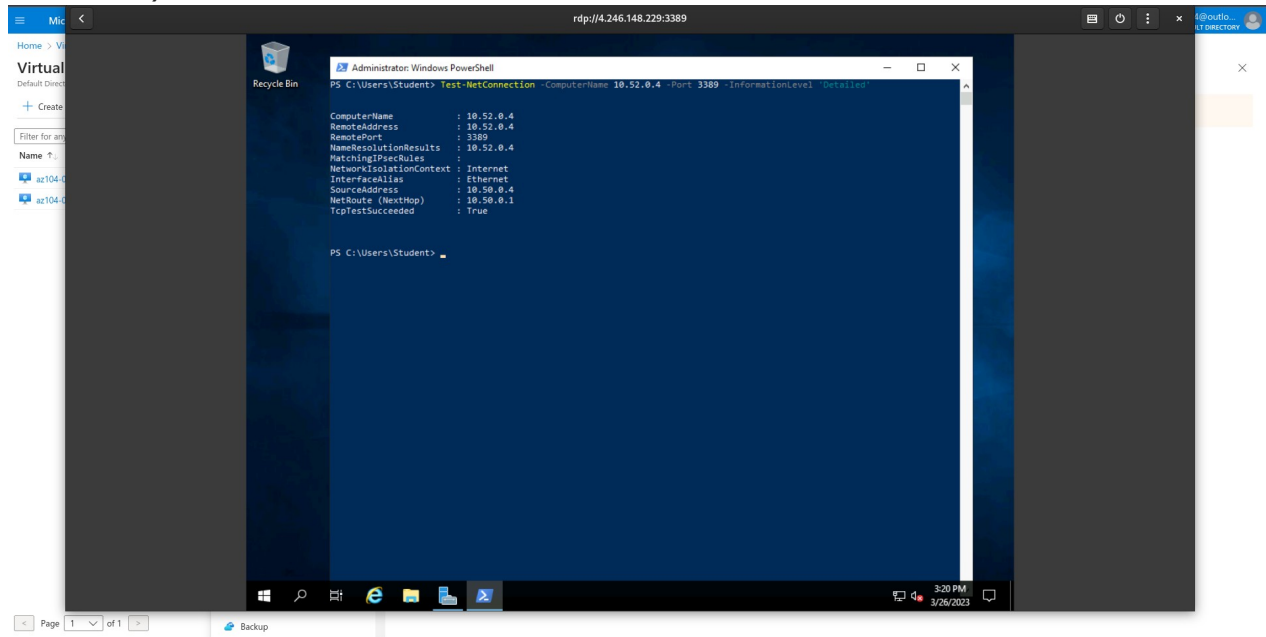
Task 3: Test intersite connectivity

In this task, you will test connectivity between virtual machines on the three virtual networks that you connected via local and global peering in the previous task.

6. In the Windows PowerShell console window, run the following to test connectivity to az104-05-vm1 (which has the private IP address of 10.51.0.4) over TCP port 3389:



8. In the Windows PowerShell console window, run the following to test connectivity to az104-05-vm2 (which has the private IP address of 10.52.0.4):



14. In the Windows PowerShell console window, run the following to test connectivity to az104-05-vm2 (which has the private IP address of 10.52.0.4) over TCP port 3389:

