

# Configuration Manual

MSc Research Project  
MSc Cybersecurity

Vishak Anandha Kumar  
Student ID: x23206055

School of Computing  
National College of Ireland

Supervisor: Prof: Jawad Salahuddin

**National College of Ireland**  
**MSc Project Submission Sheet**



**School of Computing**

Vishak Anandha Kumar

**Student Name:** .....  
**Student ID:** 23206055 .....  
**Programme:** MSc Cybersecurity ..... **Year:** 2024 .....  
MSc Research Project .....  
**Module:** .....  
Jawad Salahuddin .....  
**Lecturer:** .....  
**Submission Due Date:** 12/12/2024 .....  
**Project Title:** Multi-cloud Infrastructure Provisioning with Auto Scaling .....  
819 ..... 07  
**Word Count:** ..... **Page Count:** .....

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:** Vishak Anandha Kumar .....  
12/12/2024 .....  
**Date:** .....

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
<b>Attach a Moodle submission receipt of the online project submission,</b> to each project (including multiple copies).	<input type="checkbox"/>
<b>You must ensure that you retain a HARD COPY of the project,</b> both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

<b>Office Use Only</b>	
Signature:	
Date:	
Penalty Applied (if applicable):	

# Configuration Manual

Vishak Anandha Kumar  
Student ID: x23206055

## 1. Introduction

This configuration manual contains step by step configuration screenshots of the tools which are included in the deployment of Multi cloud Auto scaling. It aims to provide detailed guidance on setting up and integrating resources across AWS and Azure for seamless, scalable, and secure workload management.

## 2. System Requirements

### Hardware Specifications

- Device: Acer Aspire 3 15
- Processor: AMD RYZEN 7000 series 5
- ROM: 512GB
- RAM: 8GB

### Software Specification

- Windows 11
- Terraform and Notepad
- Windows Powershell, AWS and Azure CLI
- Amazon Web Services
- Azure

## 3. Configuration

For auto-scaling group mlcl-asg, the desired capacity is set at 2 instances; the capability to scale the configuration allows an upper limit of 4 instances and a lower limit of 2 instances. It uses a launch template (mlcl-tmp) specifying an AMI and t2.micro instance type. The ASG is integrated with the target group (mlcl-tg) by directing traffic through load balancing to ensure high availability and resource-efficient distribution. This setup is essential for handling variable workloads dynamically while maintaining fault tolerance and scalability.

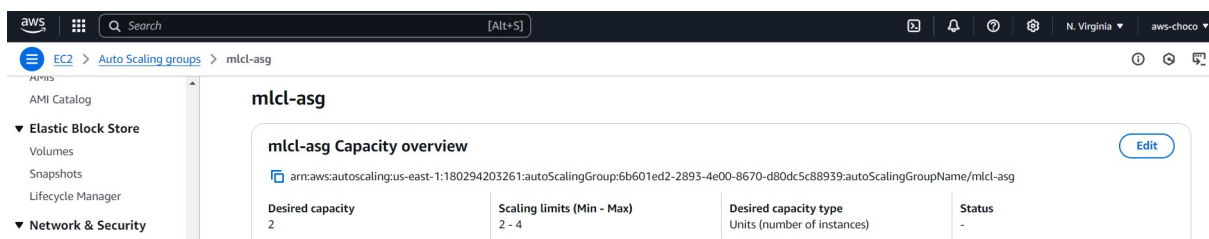


Fig 1




Details	Integrations - new	Automatic scaling	Instance management	Instance refresh
<b>Launch template</b>				
Launch template  lt-0329d7a319dab64ef mlcl-tmp	AMI ID  ami-0453ec754f44f9a4a	Instance type t2.micro		
Version Default	Security groups -	Security group IDs  sg-0a1a611d8f5330d5c		

Fig 1.1


Details	Integrations - new	Automatic scaling	Instance management
<b>Load balancing</b>			
Load balancer target groups  mlcl-tg	Classic Load Balancers -		

Fig 1.2

- Terraform Installed and associated with my local to run code for aws and azure using cli.

```

Windows PowerShell
Terraform v1.9.8
on windows_386
PS C:\Users\VISHAK> terraform init
Terraform initialized in an empty directory!

The directory has no Terraform configuration files. You may begin working
with Terraform immediately by creating Terraform configuration files.
PS C:\Users\VISHAK> notepad main.tf
PS C:\Users\VISHAK> notepad main.tf
PS C:\Users\VISHAK> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.74.0...
- Installed hashicorp/aws v5.74.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
PS C:\Users\VISHAK>

```

Fig 2

- IAM has created just to get the access and secret key to connect azure cli with terraform

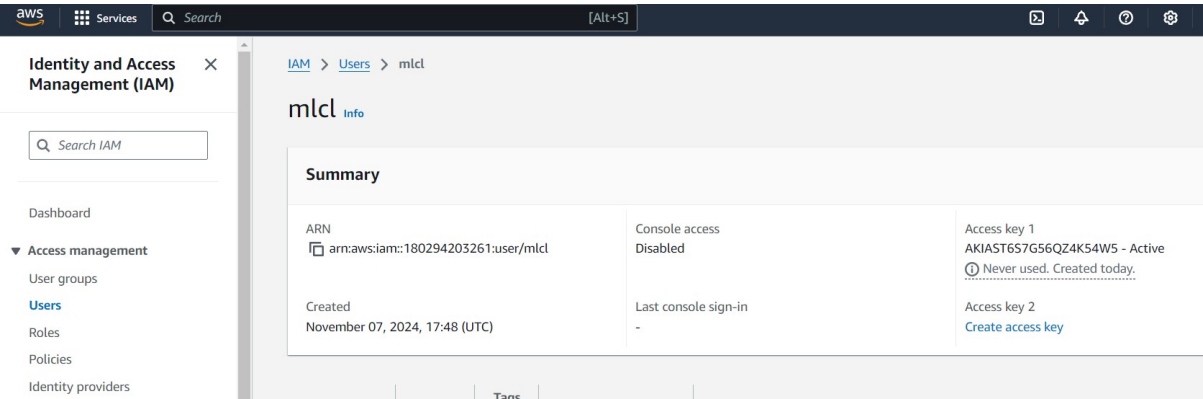


Fig 3

- Both aws and azure associated with terraform, meanwhile azure also get associated using subscription ID

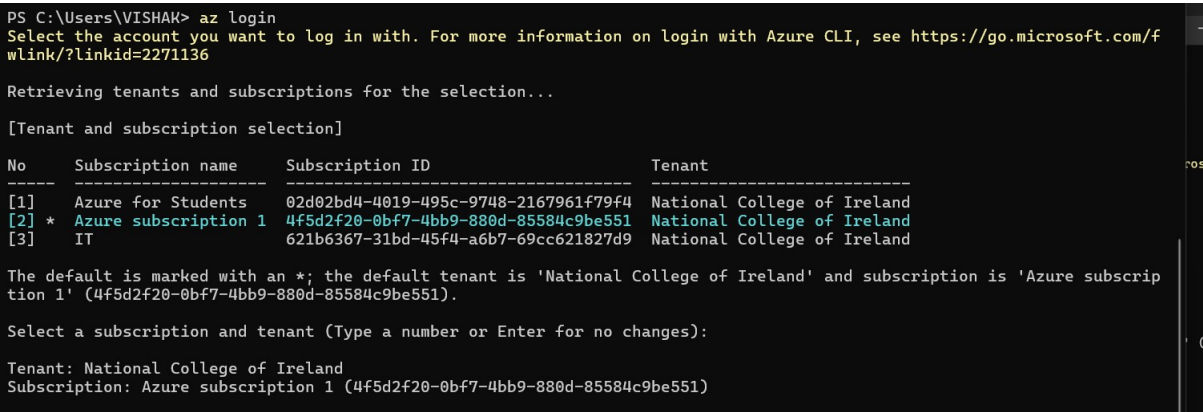


Fig 4

- VPC has been created

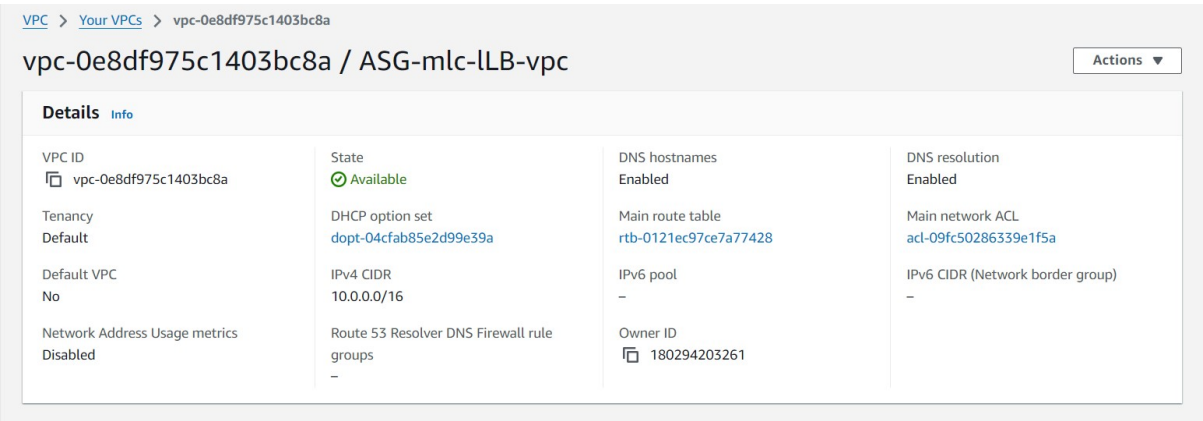


Fig 5

- In this we can see that required features associated with VPC

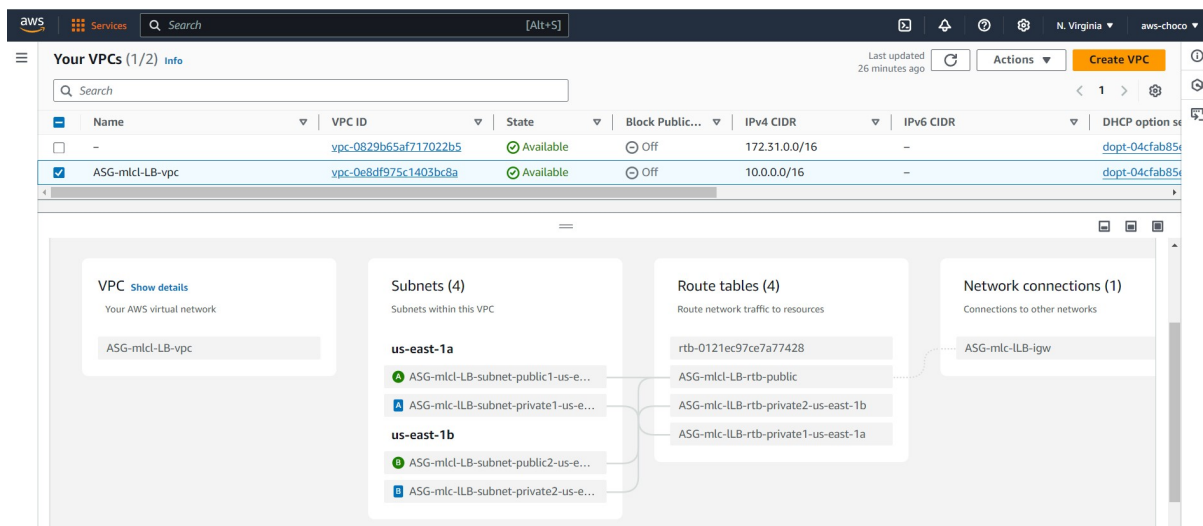


Fig 6

- The subnets are distributed across multiple availability zones (AZs) within the same region (us-east-1a and us-east-1b) to ensure fault tolerance. If one AZ becomes unavailable, resources in the other AZ can continue functioning without interruption, ensuring business continuity.

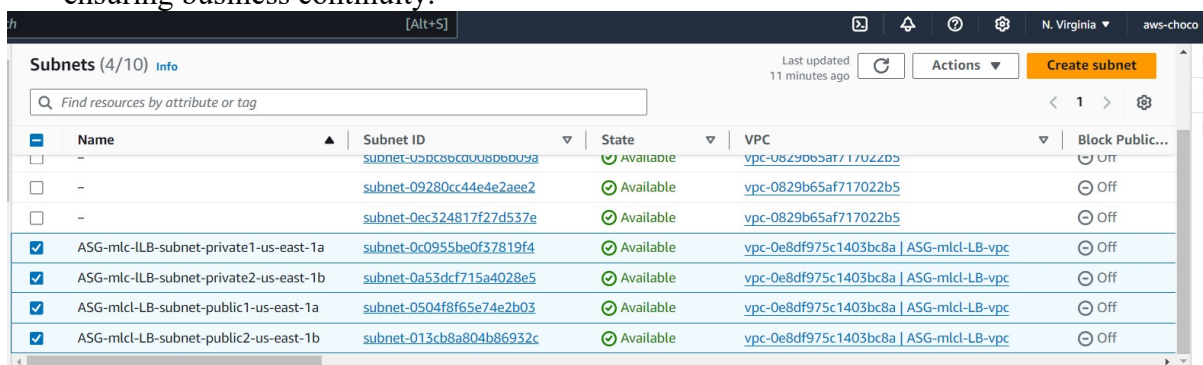


Fig 7

- VPC associated with subnets has been attached to Internet gateway

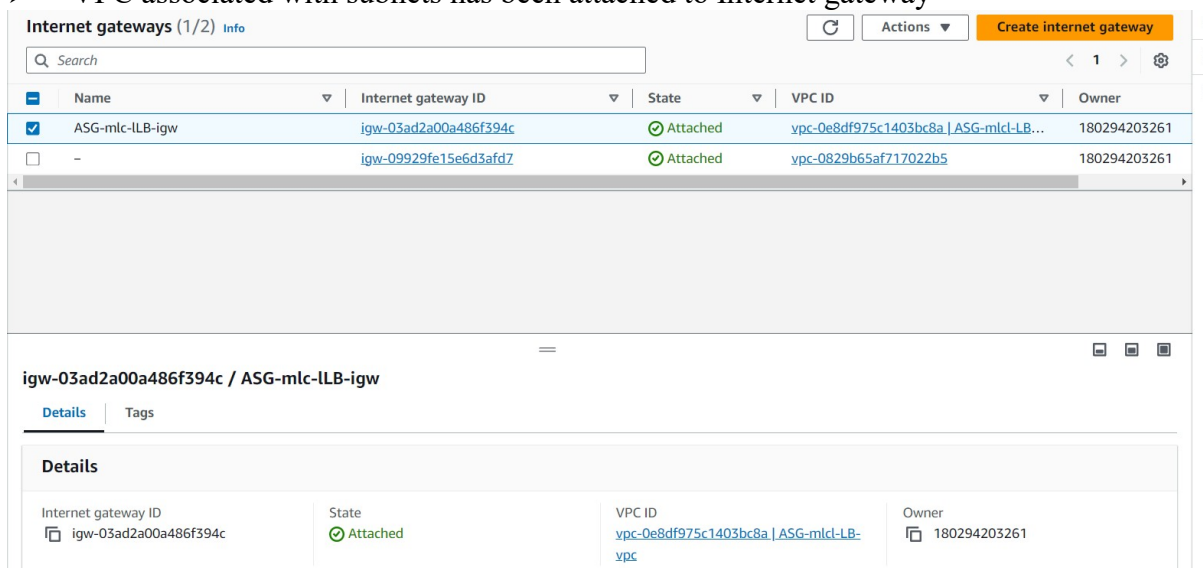
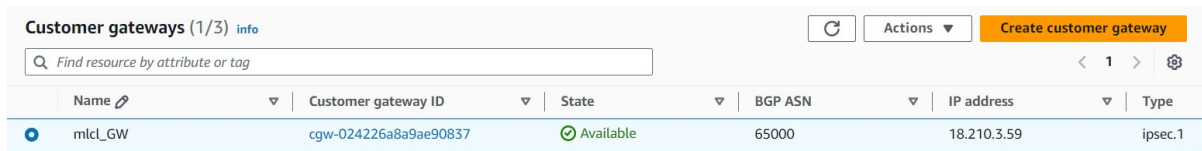


Fig 8

- The customer gateway here is the part of VPN which is to connect Azure with AWS.



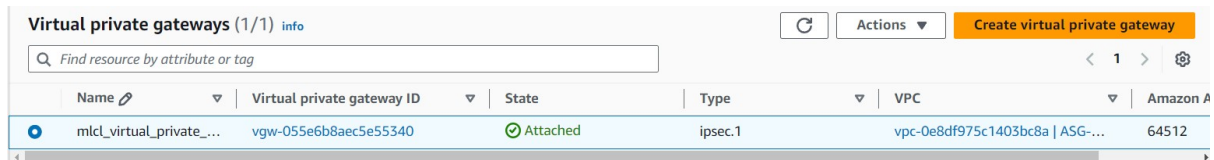
**Customer gateways (1/3)** info

Find resource by attribute or tag

Name	Customer gateway ID	State	BGP ASN	IP address	Type
mlcl_GW	cgw-024226a8a9ae90837	Available	65000	18.210.3.59	ipsec.1

Fig 9

- The virtual private gateway creates secure connection between AWS VPC and Azure VN, so the VPC ID is attached here.



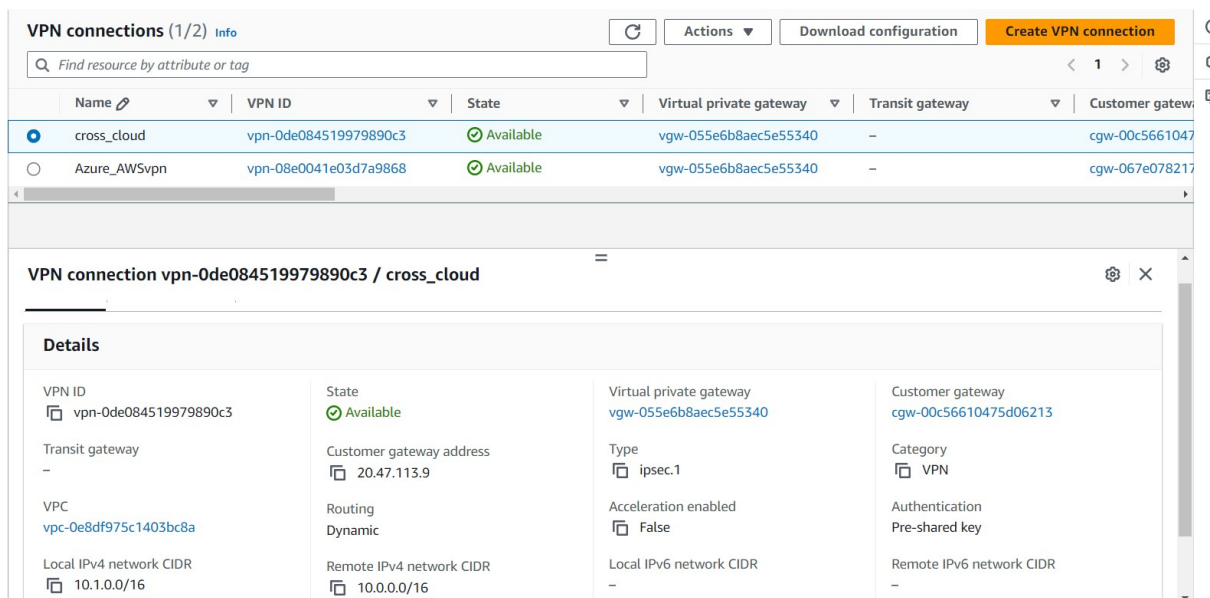
**Virtual private gateways (1/1)** info

Find resource by attribute or tag

Name	Virtual private gateway ID	State	Type	VPC	Amazon A
mlcl_virtual_private_...	vgw-055e6b8aec5e55340	Attached	ipsec.1	vpc-0e8df975c1403bc8a	ASG-... 64512

Fig 10

- Finally VPN connection established in AWS



**VPN connections (1/2)** info

Find resource by attribute or tag

Name	VPN ID	State	Virtual private gateway	Transit gateway	Customer gateway
cross_cloud	vpn-0de084519979890c3	Available	vgw-055e6b8aec5e55340	-	cgw-00c5661047
Azure_AWSvpn	vpn-08e0041e03d7a9868	Available	vgw-055e6b8aec5e55340	-	cgw-067e078217

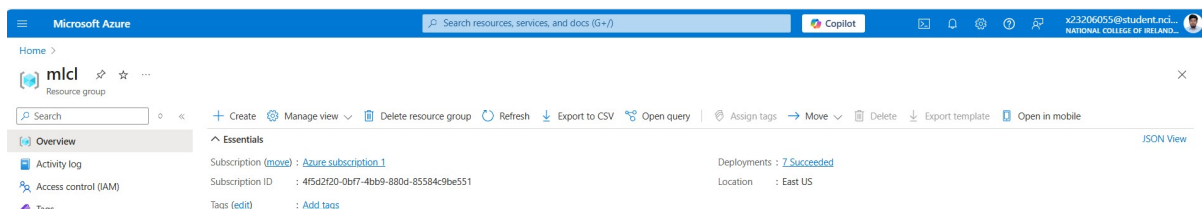
  

**VPN connection vpn-0de084519979890c3 / cross\_cloud**

Details			
VPN ID vpn-0de084519979890c3	State Available	Virtual private gateway vgw-055e6b8aec5e55340	Customer gateway cgw-00c56610475d06213
Transit gateway -	Customer gateway address 20.47.113.9	Type ipsec.1	Category VPN
VPC vpc-0e8df975c1403bc8a	Routing Dynamic	Acceleration enabled False	Authentication Pre-shared key
Local IPv4 network CIDR 10.1.0.0/16	Remote IPv4 network CIDR 10.0.0.0/16	Local IPv6 network CIDR -	Remote IPv6 network CIDR -

Fig 11

- In Azure, resource group has been created and necessary resources were also deployed.



**Microsoft Azure**

Search resources, services, and docs (G+)

Home

mlcl Resource group

Search

+ Create Manage view Delete resource group Refresh Export to CSV Open query Assign tags Move Delete Export template Open in mobile

**Overview**

Essentials

Subscription (move): Azure subscription 1

Subscription ID: 4f5d2f20-0bf7-4bb9-880d-85584c9be551

Deployments: 7 Succeeded

Location: East US

Tags (edit): Add tags

JSON View

Fig 12

- Establish an Azure virtual network (mlclVN) with the right address space, for example, 10.1.0.0/16, to avoid overlapping IP address space with the AWS VPC and connect the Azure VNet to the VPN Gateway to communicate securely.

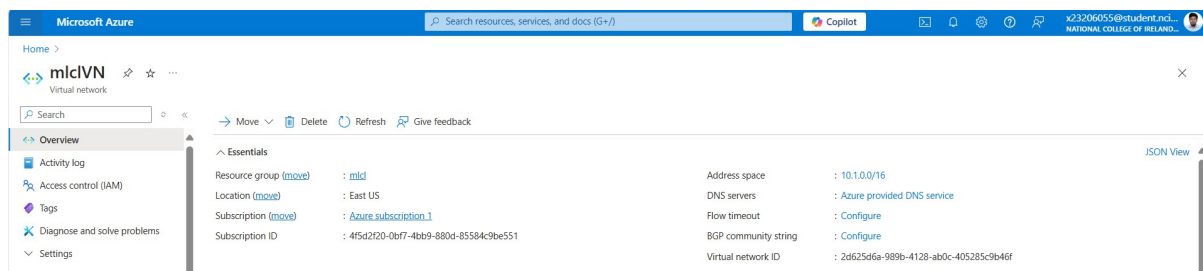


Fig 13

- Create a route-based VPN Gateway (VpnGw2AZ) in Azure and connect it to the AWS VGW. Configure the BGP settings and Azure route table to manage traffic between Azure and AWS seamlessly.

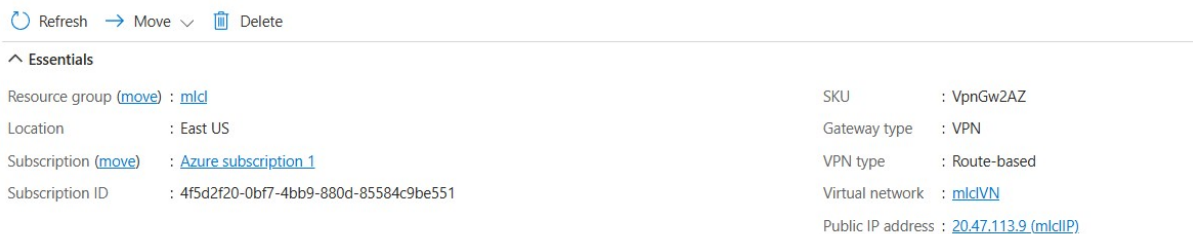


Fig 14

- Local Network Gateway is the Azure endpoint managing the VPN connection to the AWS Virtual Private Gateway. Ensure proper pairing and routing to bridge the networks so Azure can communicate across cloud with AWS.

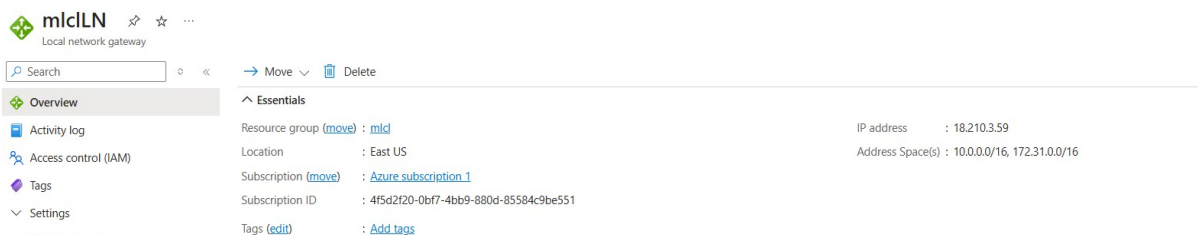


Fig 15

- Allows UDP traffic on port 4500 to Azure virtual network (10.1.0.0/16) from AWS CIDR block (10.0.0.0/16). Enables traffic from Azure virtual network (10.1.0.0/16) to AWS CIDR block.

Home > m1c1NSG Network security group

Search

Move Delete Refresh Give feedback

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Inbound security rules

Outbound security rules

Network interfaces

Subnets

Properties

Locks

Monitoring

Automation

Help

Essentials

Resource group (move) : m1c1

Location : East US

Subscription (move) : Azure subscription 1

Subscription ID : 4f5d2f20-0bf7-4bb9-880d-85584c9be551

Tags (edit) : Add tags

Custom security rules : 1 inbound, 1 outbound

Associated with : 1 subnets, 0 network interfaces

Filter by name

Port == all Protocol == all Source == all Destination == all Action == all

Priority	Name	Port	Protocol	Source	Destination	Action
<b>Inbound Security Rules</b>						
100	AllowCidrBlockCustom4500n...	4500	UDP	10.0.0.0/16	10.1.0.0/16	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBou...	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny
<b>Outbound Security Rules</b>						
110	AllowCidrBlockCustom4500n...	4500	UDP	10.1.0.0/16	10.0.0.0/16	Allow
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowInternetOutBound	Any	Any	Any	Internet	Allow
65500	DenyAllOutBound	Any	Any	Any	Any	Deny

Fig 16

- A route will forward any traffic to AWS networks (10.0.0.0/16) through the Virtual Network Gateway. So, Azure resources in that subnet can now communicate with AWS resources over the VPN connection.

Home > m1c1RouteTable Route table

Search

Move Delete Refresh Give feedback

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Configuration

Routes

Subnets

Properties

Locks

Monitoring

Automation

Help

Essentials

Resource group (move) : m1c1

Location : East US

Subscription (move) : Azure subscription 1

Subscription ID : 4f5d2f20-0bf7-4bb9-880d-85584c9be551

Tags (edit) : Add tags

Associations : 1 subnet associations

Routes

Search routes

Name	Address prefix	Next hop type
az_aws_route	10.0.0.0/16	Virtual network gateway

Subnets

Search subnets

Name	Address range	Virtual network
GatewaySubnet	10.1.1.0/24	m1c1VN

Fig 17

## Reference

N/A.