Assignment: Implementing Internal & External Load Balancer in Azure

---

Prepared by:
Virat Pandey

Celebal Technology
Cloud Infrastructure & Security Internship

Research & Development Document

July, 2025

Table of Contents

Introduction

In cloud computing, **load balancing** is a fundamental concept used to enhance the availability, scalability, and performance of applications and services. Whether it's handling incoming web traffic or distributing internal requests between backend servers, load balancers ensure no single resource is overwhelmed while maintaining high reliability and uptime.

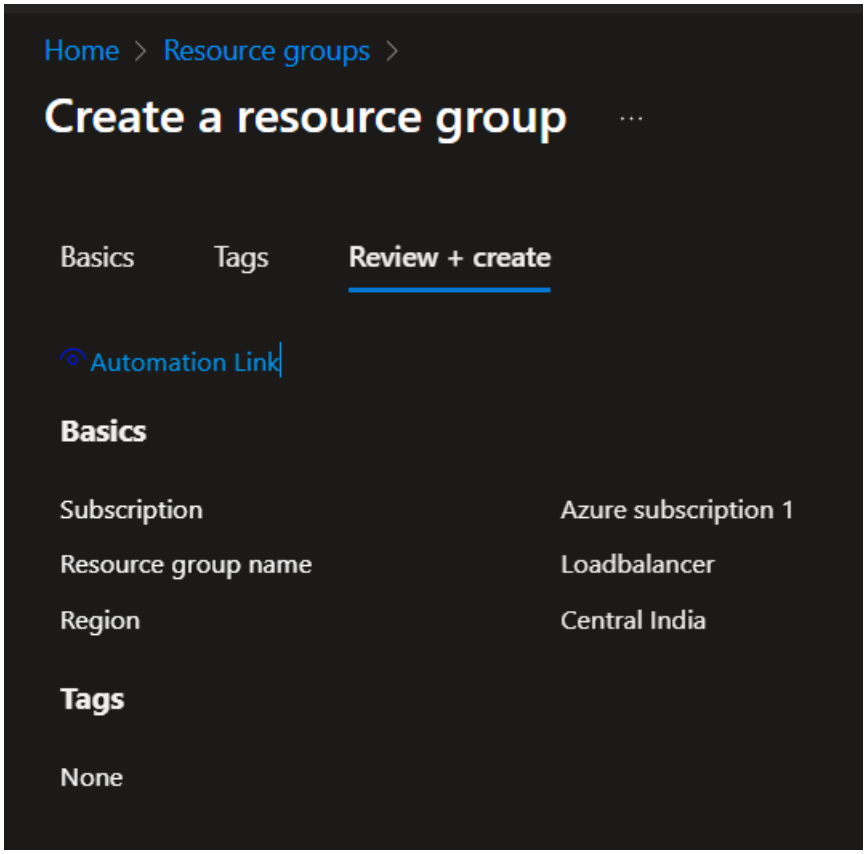
This project focuses on the **implementation and verification of both External and Internal Load Balancers in Microsoft Azure**. The **External Load Balancer (ELB)** is used to distribute incoming public internet traffic across multiple virtual machines, thereby improving fault tolerance and ensuring consistent performance. On the other hand, the **Internal Load Balancer (ILB)** serves traffic only within the Azure Virtual Network (VNet), allowing private communication and load distribution between services that are not exposed to the internet.

The assignment includes deploying and configuring two virtual machines (VMs) behind an External Load Balancer and then repeating a similar process for an Internal Load Balancer. The VMs serve as IIS web servers, providing a static web page to simulate real-world service scenarios. Proper configuration of **Network Security Groups (NSGs)** and inbound/outbound rules is also covered to ensure secure and controlled access to the VMs.

This hands-on project not only demonstrates technical skills in infrastructure provisioning on Azure but also builds a foundational understanding of how enterprise-grade applications are deployed and managed using load balancing strategies. By the end of this exercise, one gains insight into key Azure services such as Load Balancers, Public IPs, Virtual Networks, NSGs, Frontend IP Configurations, Health Probes, and Backend Pools—forming the essential components of a high-availability architecture.

Resource Group Setup

A new Resource Group named **Loadbalancer** was created in the **Central India** region to manage all resources related to this project.

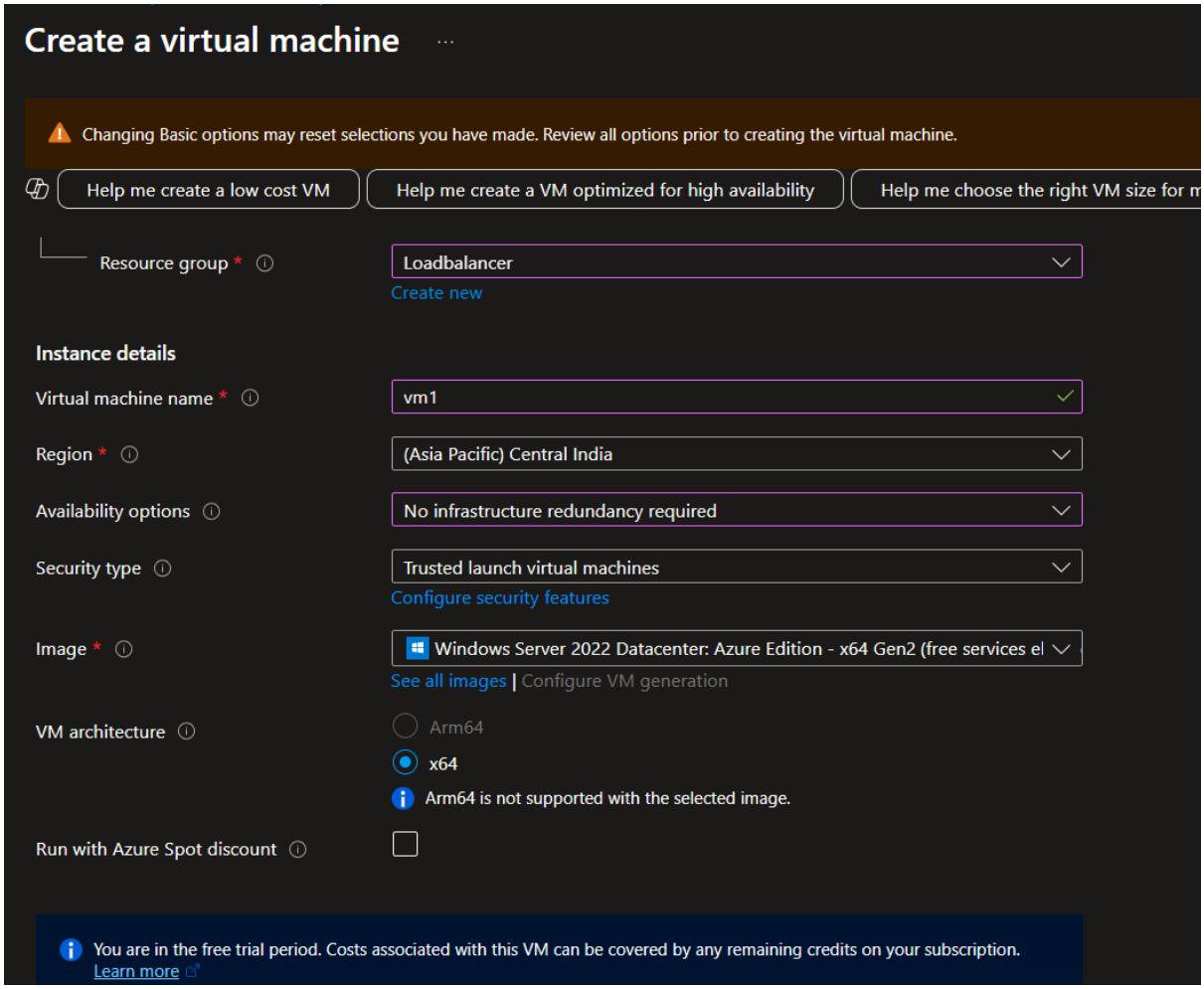


Virtual Machine Creation

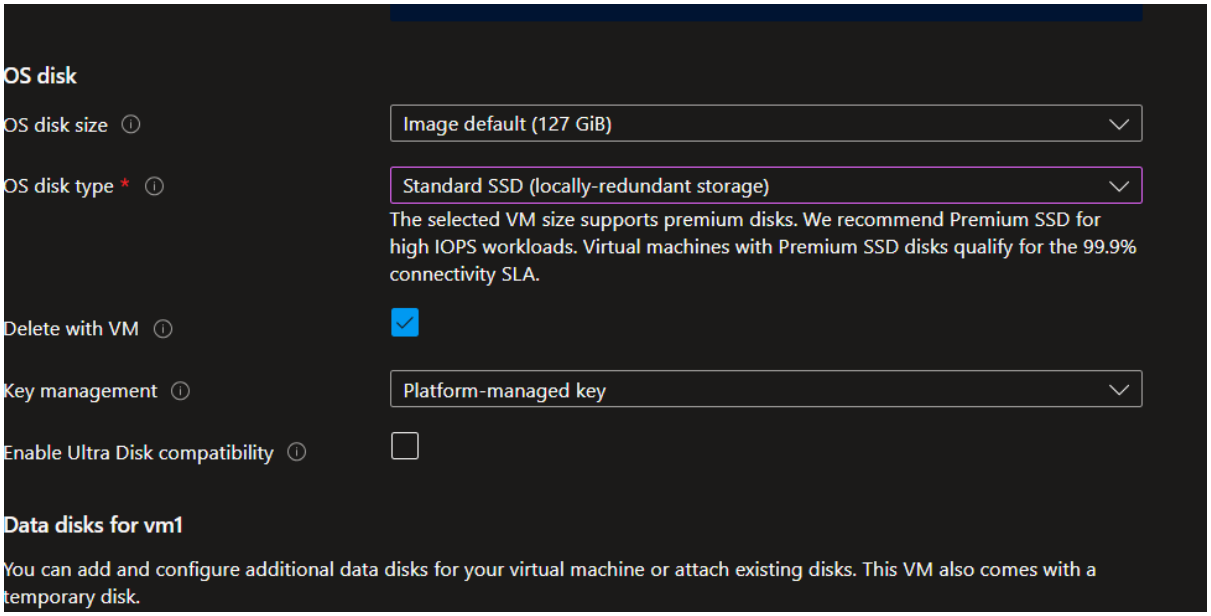
Two Windows Server VMs (vm1 and vm2) were created under the same VNet and subnet with the following config:

- OS: Windows Server 2022
- Size: Standard B1s
- Inbound Port: RDP
- IIS installed manually after RDP
- No load balancing selected during VM setup

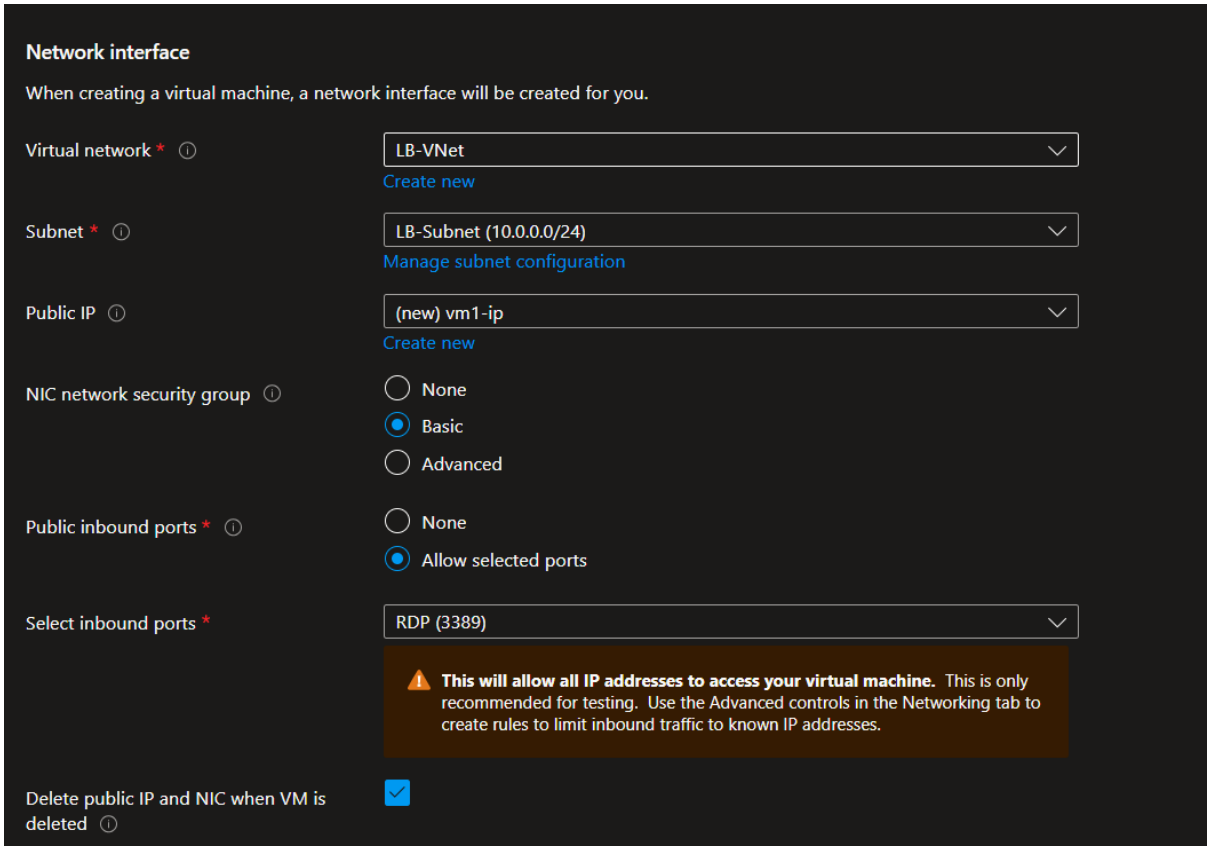
Basics Tab while creating a Virtual machine (vm1)



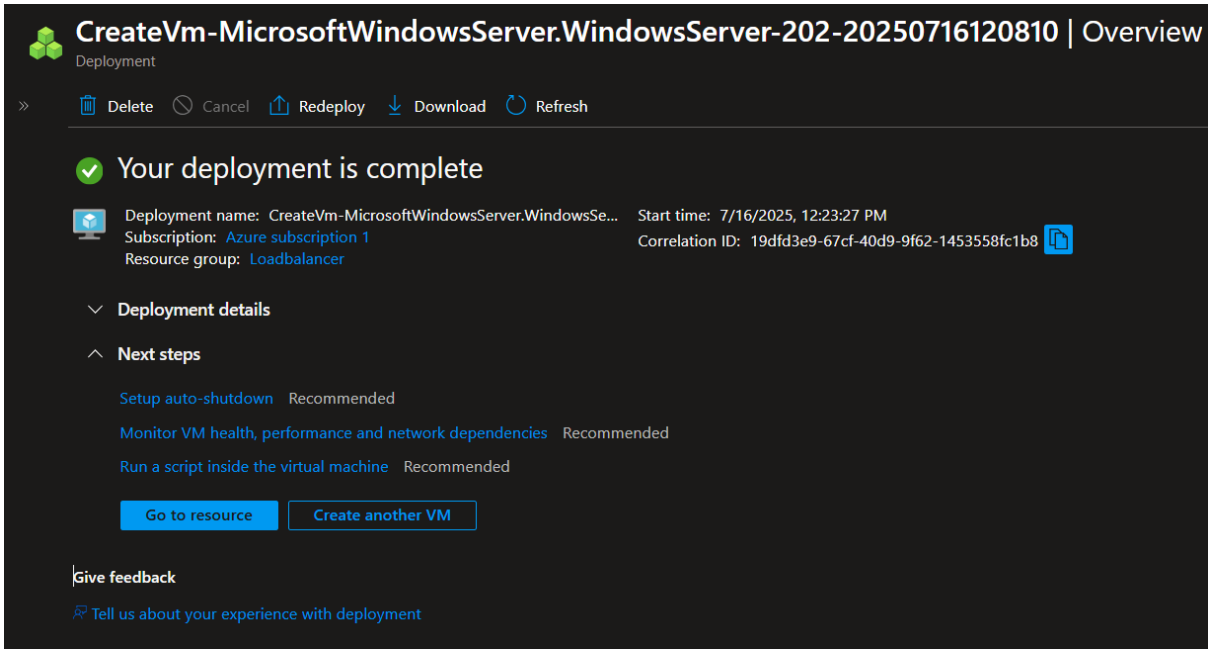
Disk Config tab:



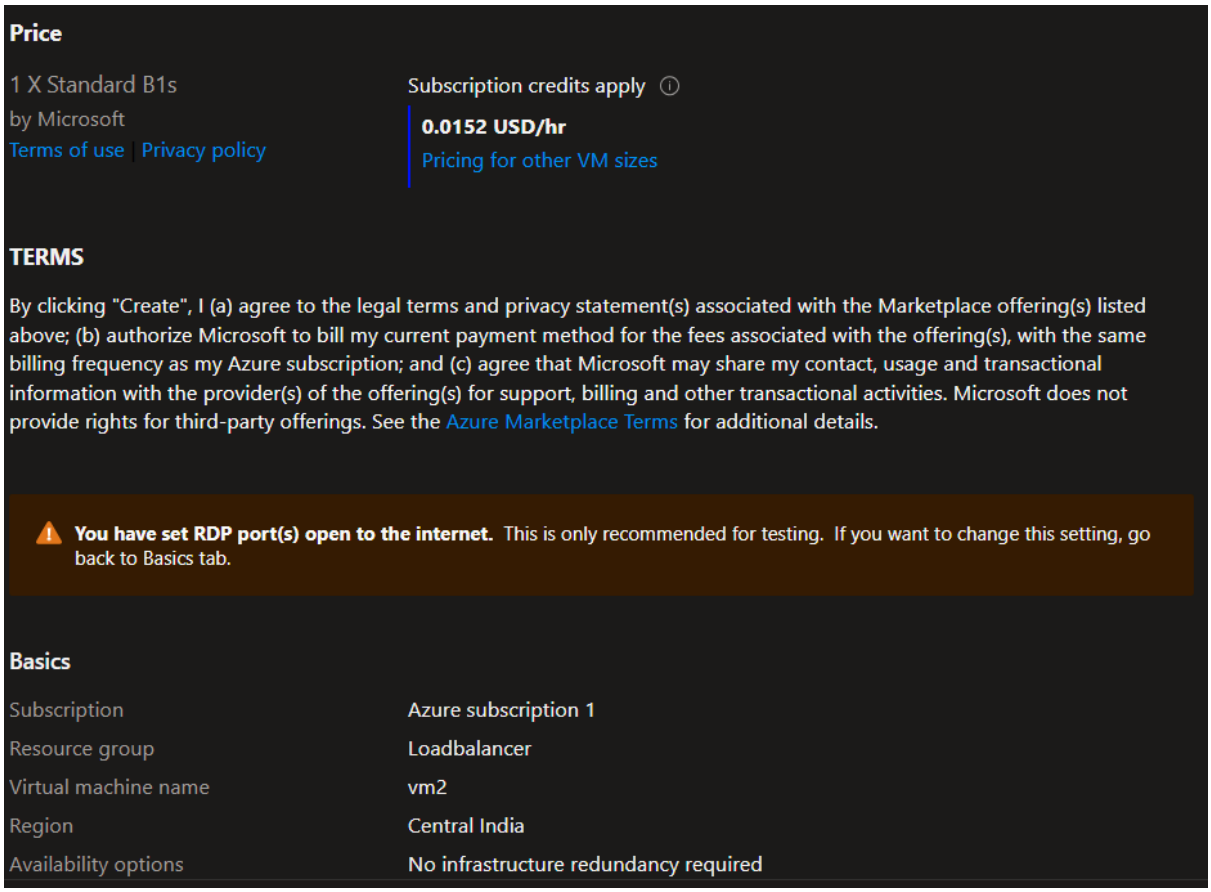
Networking config tab:

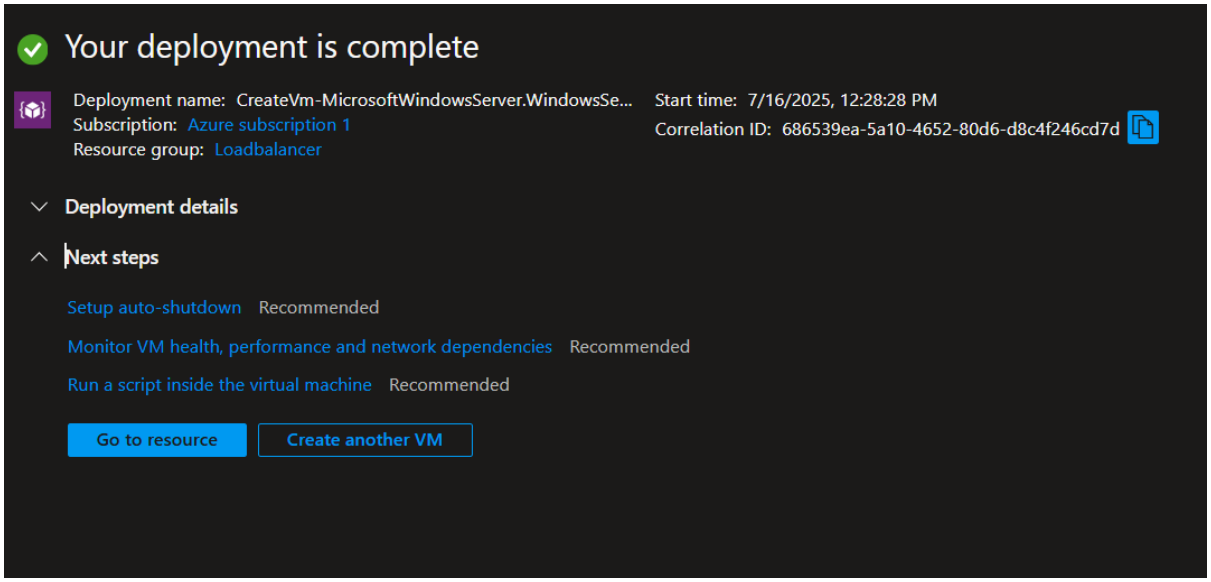


Virtual Machine Created: Overview:



here we have successfully created and deployed the virtual machine named – VM1 similarly with the same configurations we will replicate this vm and name it VM2. Here is a image of that as well:

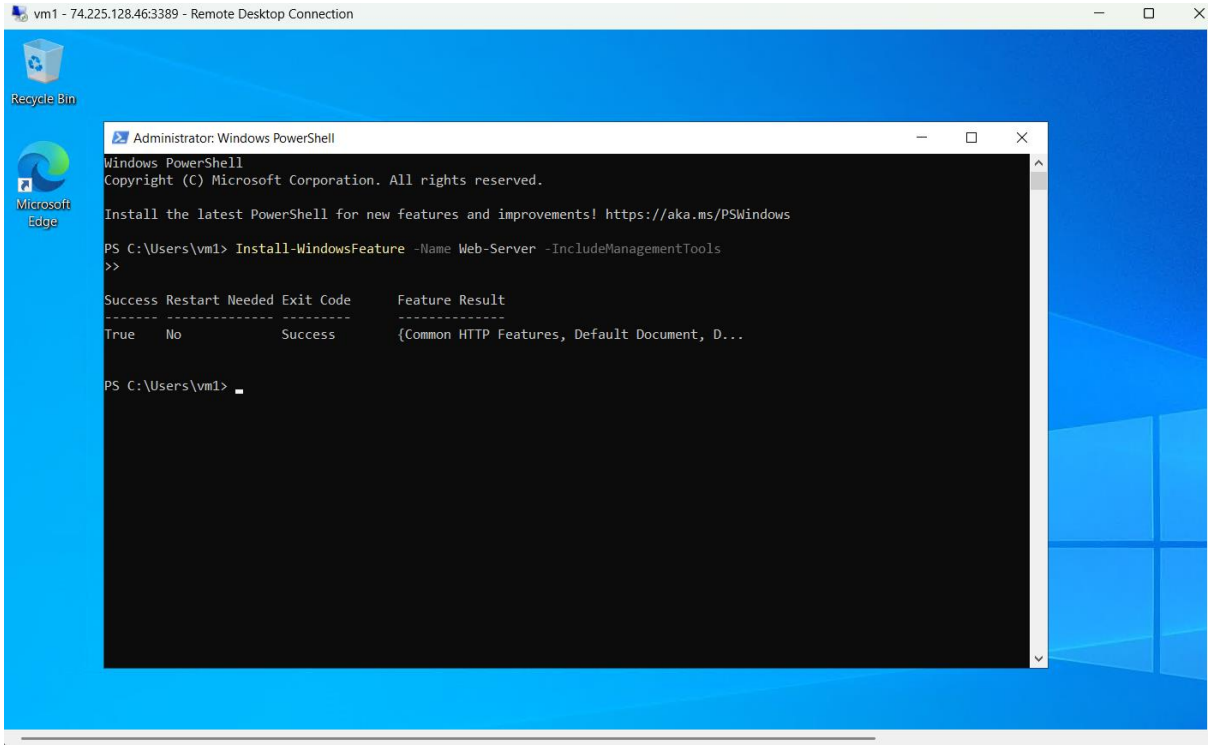




Here we are completed with Virtual machine creation while creating this vm we have initially not assigned any Load balancer to this virtual machine after that we RDP into the vms and installed the IIS server on it

So here is the IIS installed page :

first we installed he iis into the virtual machine while powershell



Here we have installed the IIS on this Virtual Machine:

We did the same for virtual machine 2 as well.

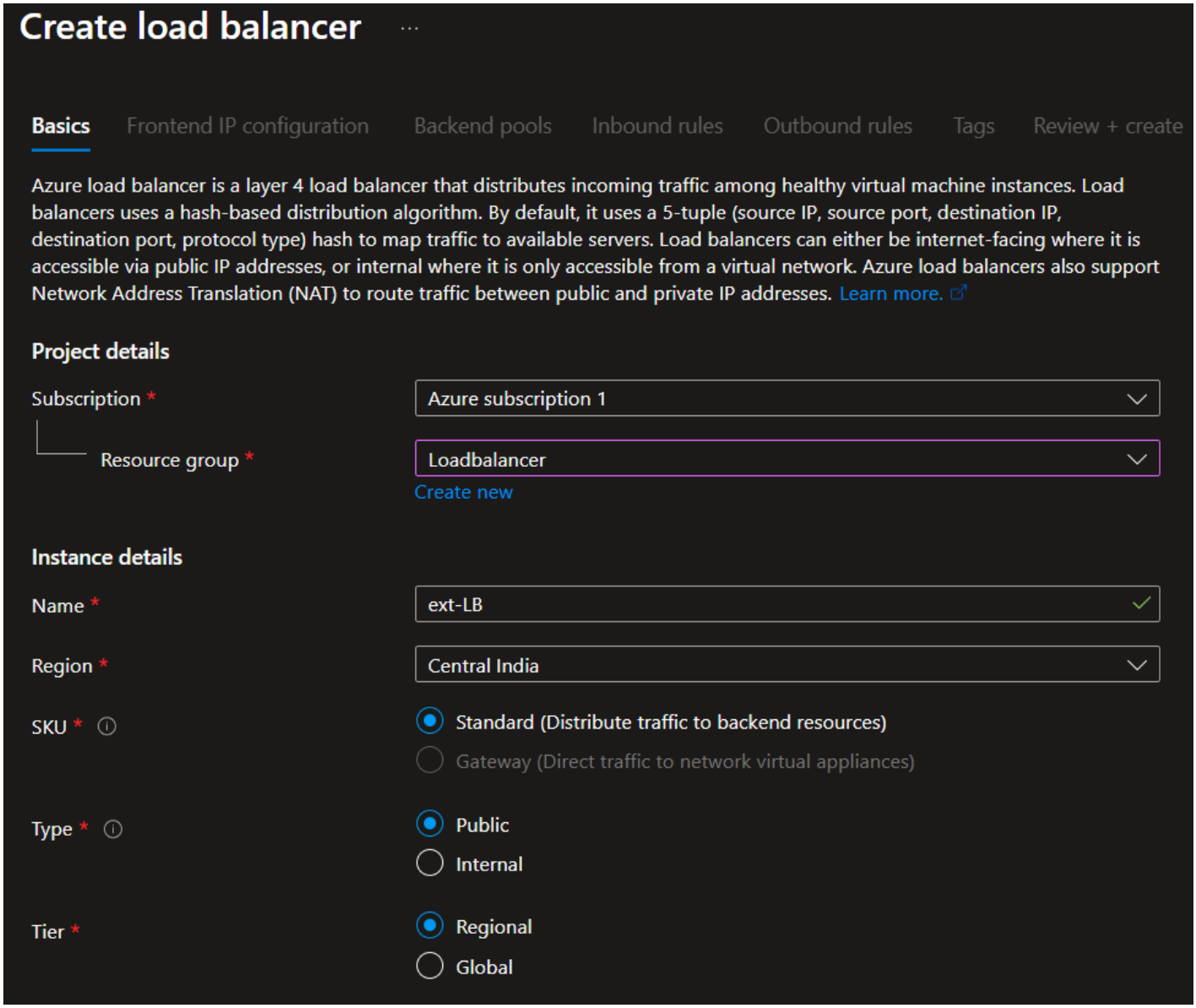
External Load Balancer Configuration:

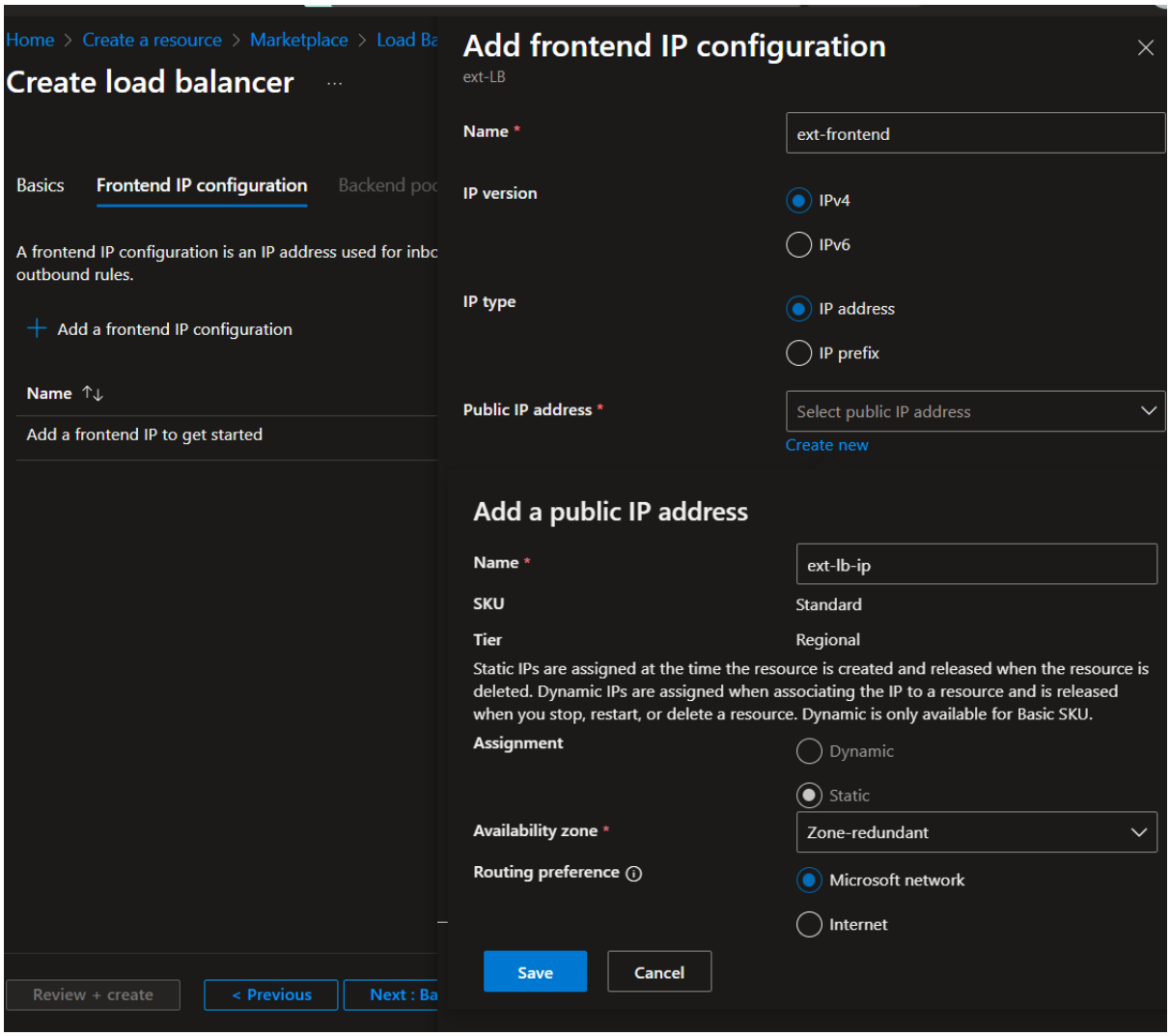
A Public Load Balancer named ext-lb was created with:

- Frontend IP: Public IP (Static)
- Backend Pool: Added vm1 and vm2
- Health Probe: TCP, Port 80
- Load Balancing Rule: Port 80, TCP

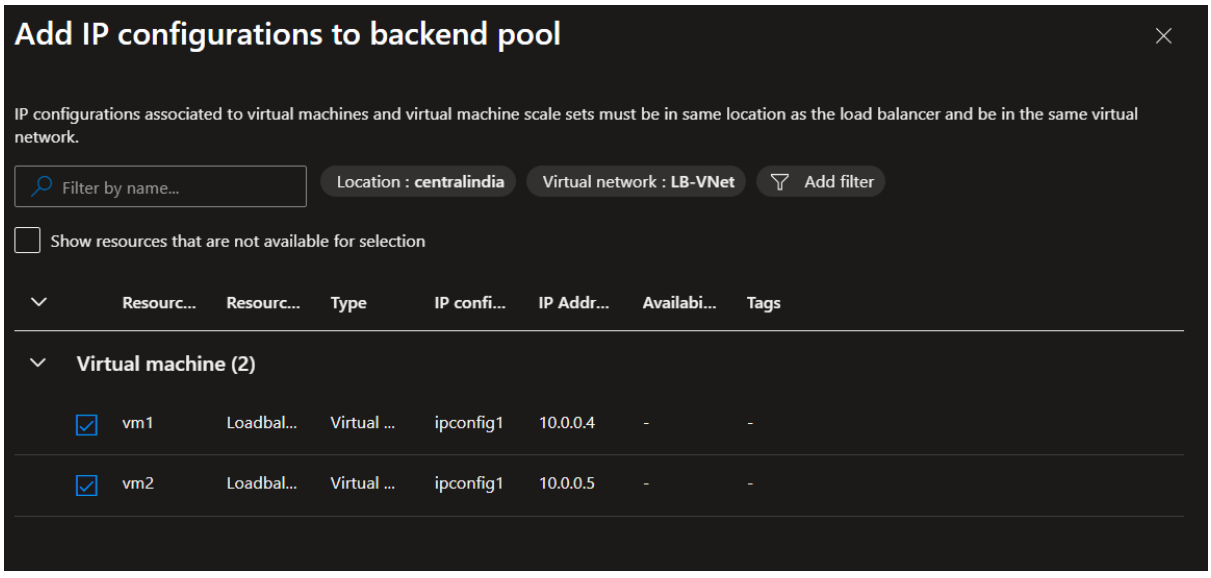
First we will start by creating a load balancer:

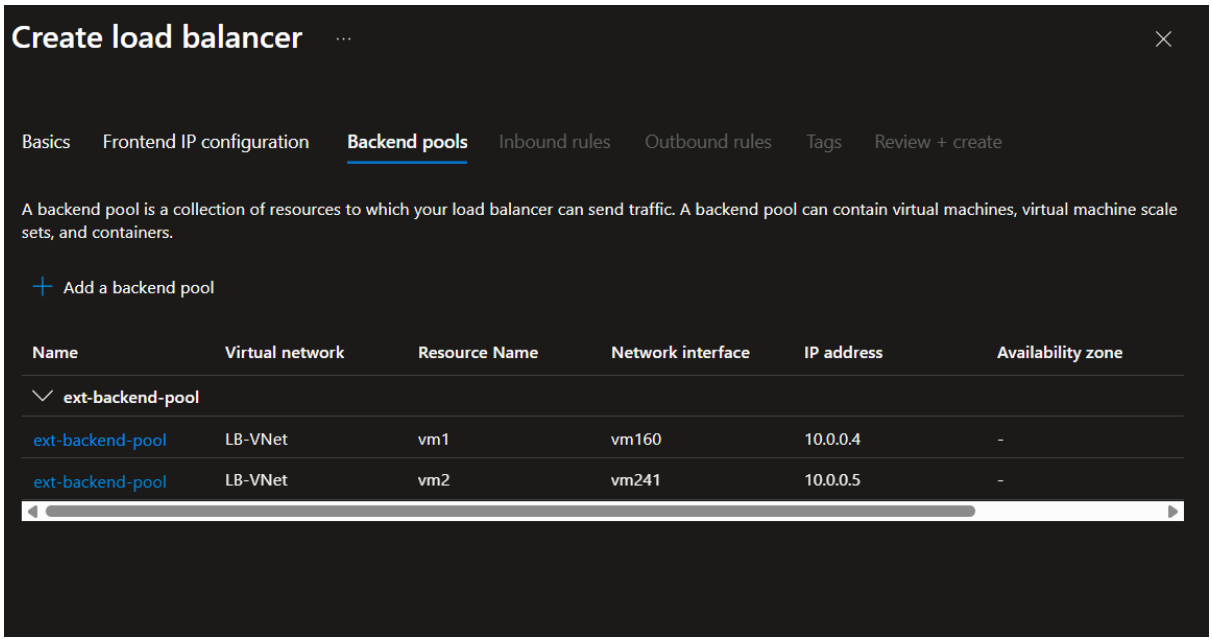
Adding frontend IP configurations::





Next we move onto Backendpool IP configuration :





Now we have done the setup for the frontend IP and the Backendpool next we will move onto

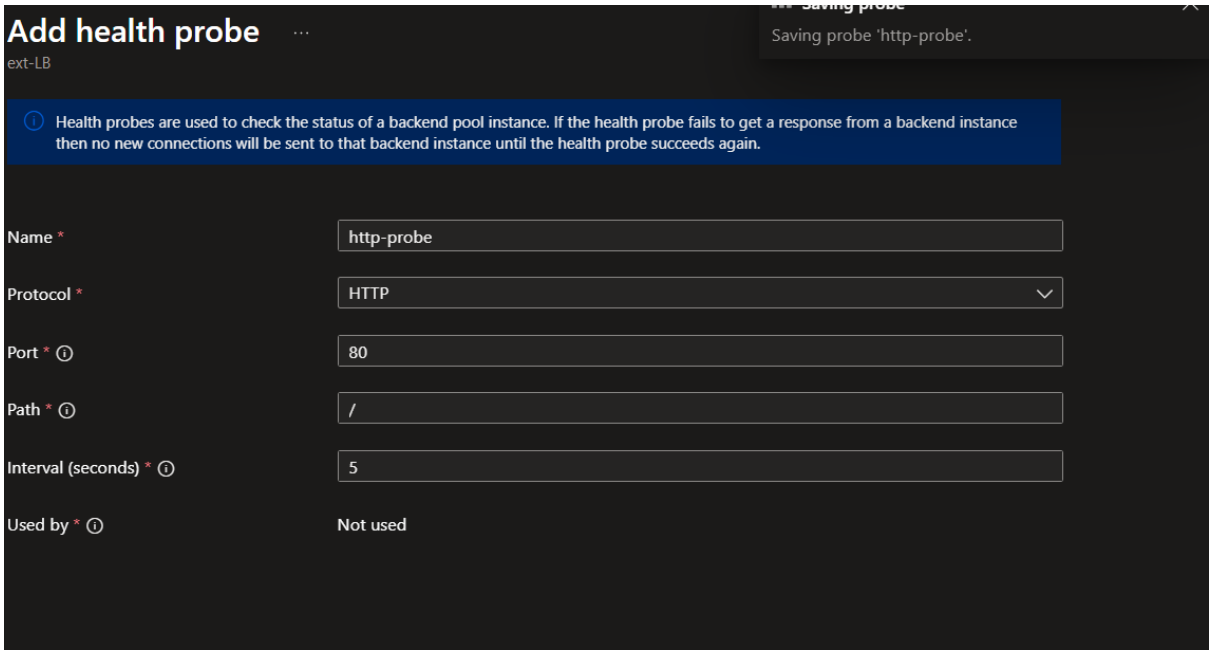
HEALTH PROBE:

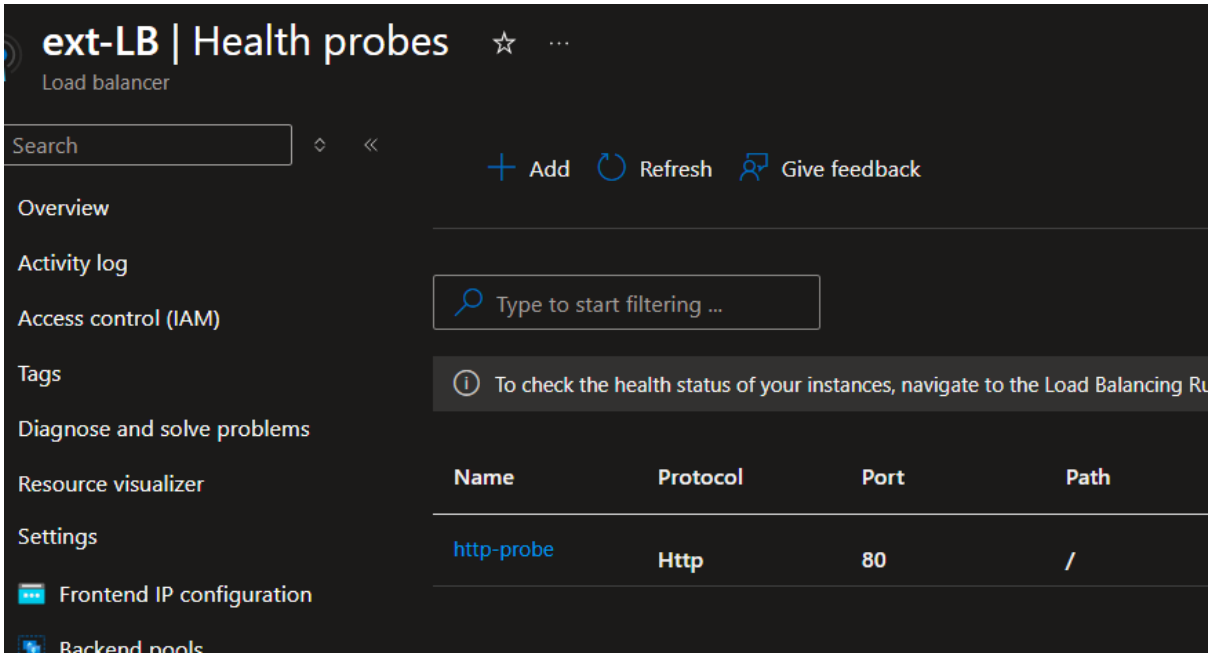
While setting up both the External and Internal Load Balancers in Azure, we configured a Health Probe to continuously monitor the health of virtual machines in the backend pool. This probe plays a crucial role in ensuring high availability and seamless traffic distribution. For this assignment, we used the following Health Probe settings:

- Protocol: HTTP
- Port: 80
- Path: /
- Interval: Default
- Unhealthy Threshold: Default

The health probe sends periodic HTTP requests to port 80 of each VM to check if the IIS server is running and responding. If a VM fails to respond to a number of consecutive probe attempts, it is considered unhealthy and temporarily removed from the load balancer's rotation. Once the VM recovers and starts responding again, it is automatically reinstated.

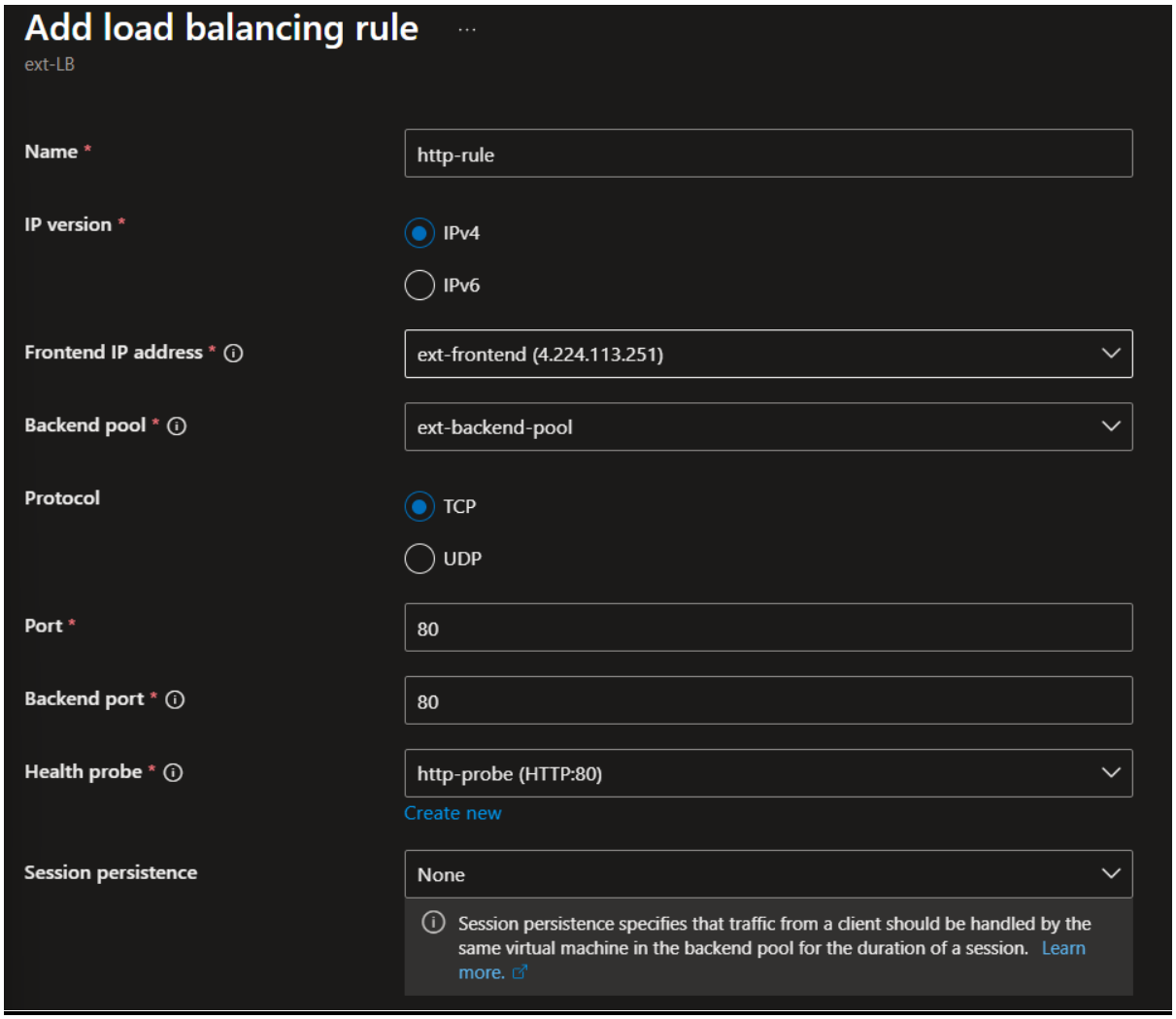
This dynamic health monitoring ensures that only healthy and responsive VMs receive traffic, enhancing the reliability and performance of the application served through the load balancer.





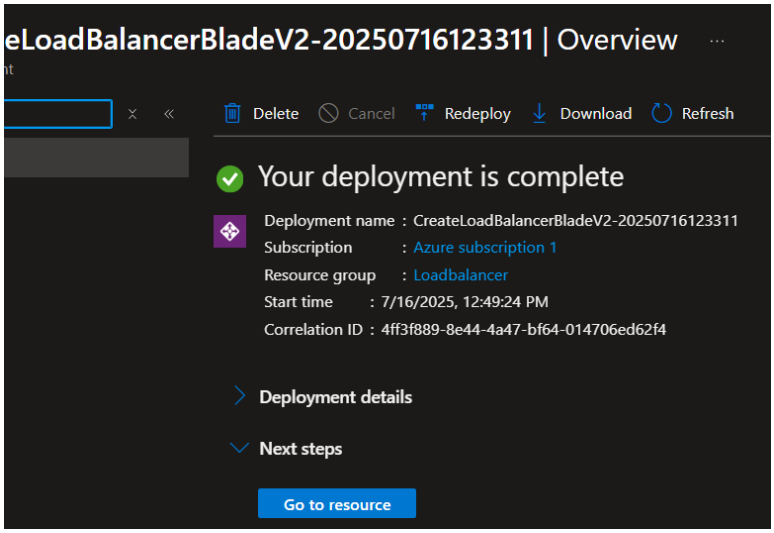
Here we have added the screenshots for the Health probe application.
Now next we will move onto Loadbalancing rule:

Load balancing Rules:



Here we have added the load balancing rule now we have done all the configurations for the load balancer now let us deploy it:

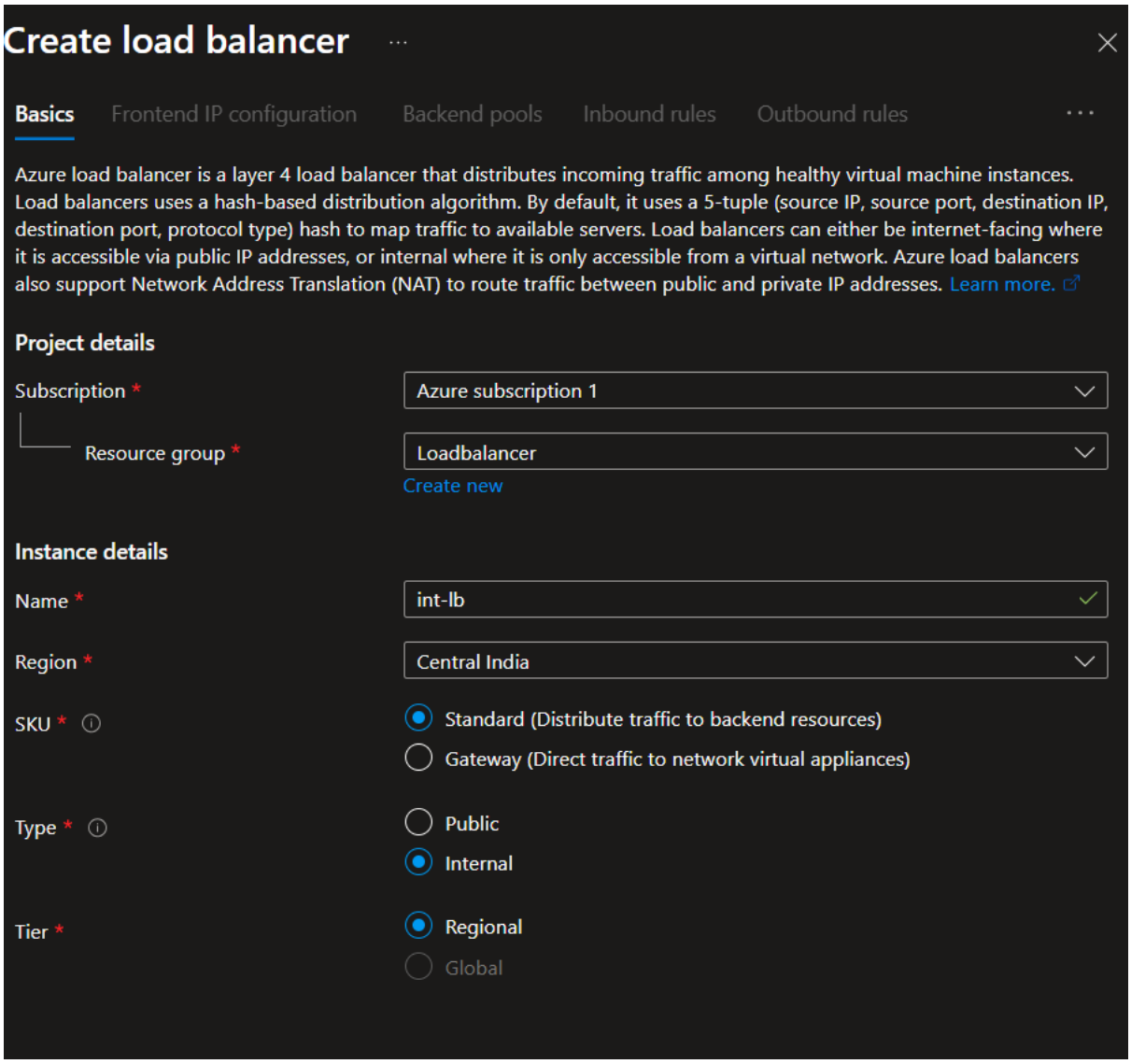
Load balancer overview:



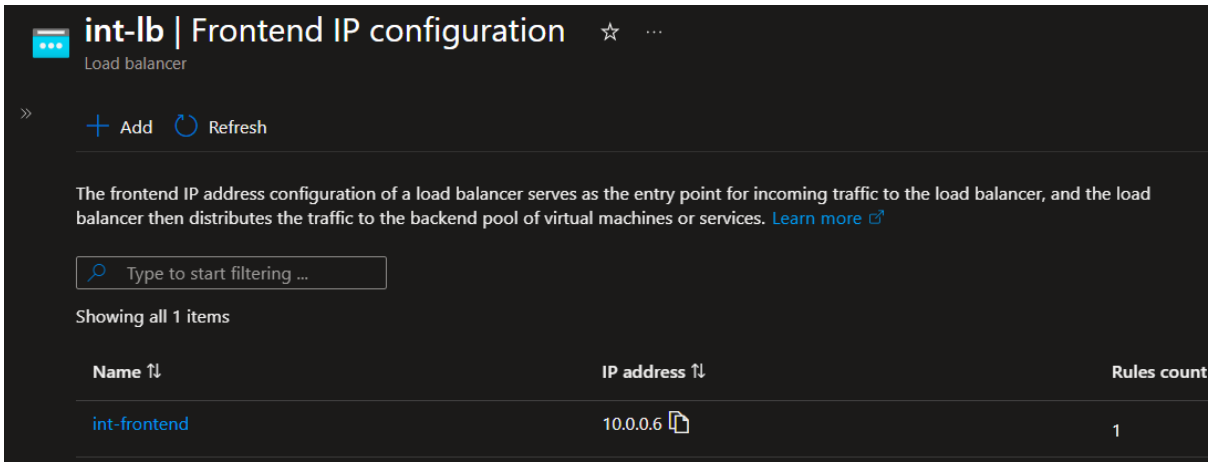
Internal Load Balancer Configuration

An Internal Load Balancer named int-lb was created in the same VNet, different subnet (lb-subnet) with:

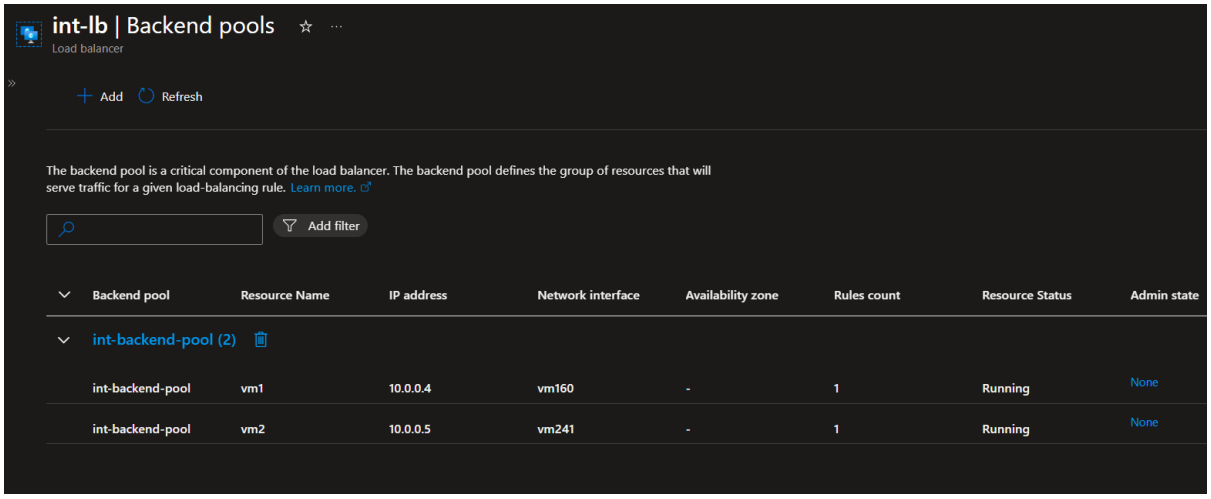
- Frontend IP: Private IP (Static)
- Backend Pool: vm1, vm2
- Health Probe: TCP, Port 80
- Load Balancing Rule: Same as external, port 80
- Verified via RDP → browser request to private IP



Here is the frontend ip configuration

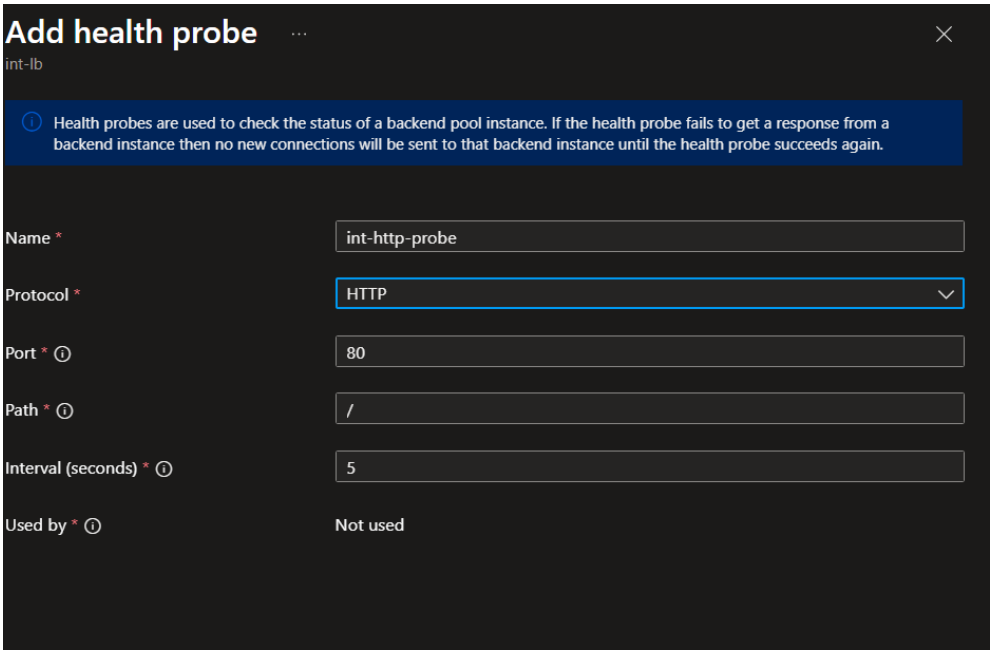


Here is the backendpool configuration:



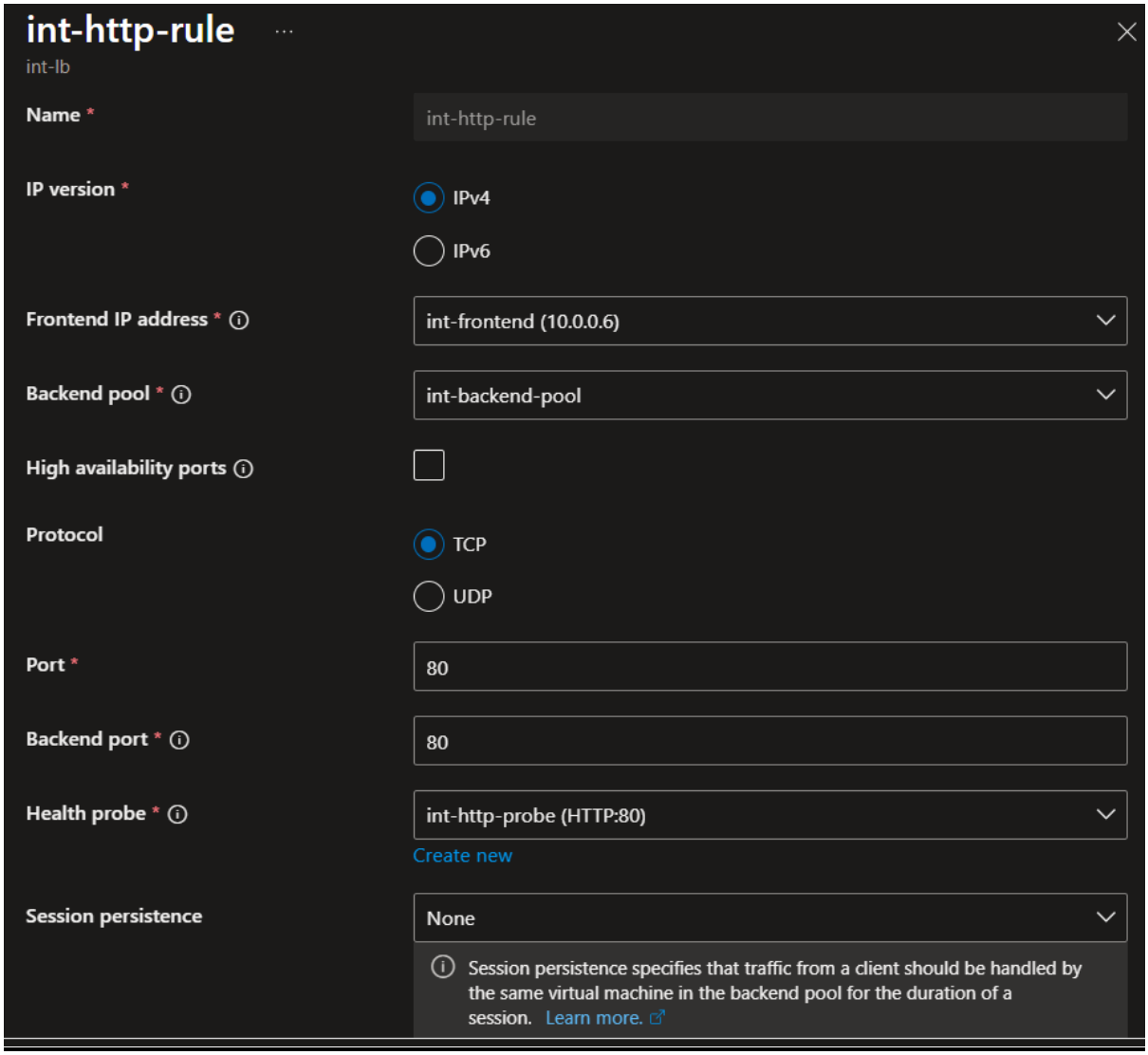
We have applied the rules on the Virtual machine 1 and virtual machine 2
Now we will move on to health probe for the internal load balancer

Health Probe:



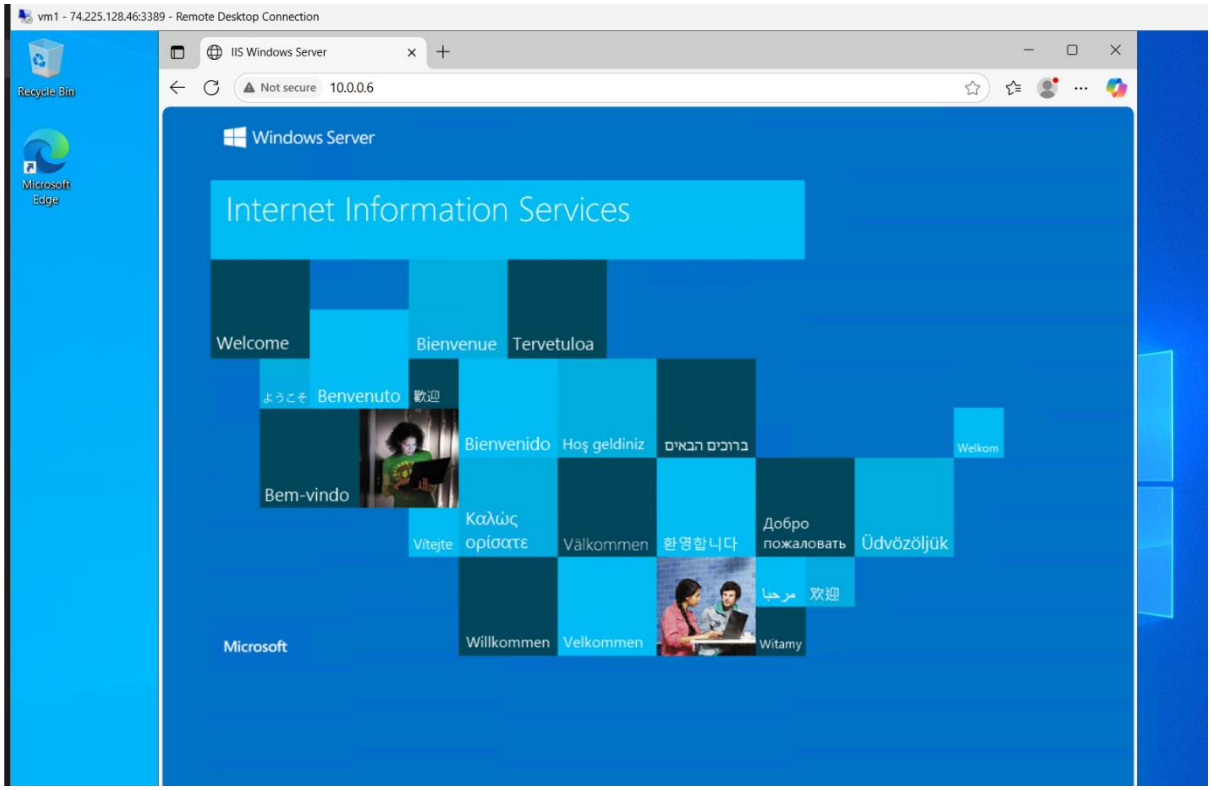
Similarly like in the external Loadbalancer we will now add the loadbalancing rule:

Load Balancing Rule:



This rule directs incoming HTTP traffic on **port 80** from the load balancer's public IP to the **backend VMs** hosting the IIS server. It works in tandem with the **health probe** to ensure that only **healthy virtual machines** receive traffic.
The configuration helps ensure **load distribution**, **high availability**, and **seamless performance** of the web application hosted across multiple instances.

Now lets verify this via RDP:

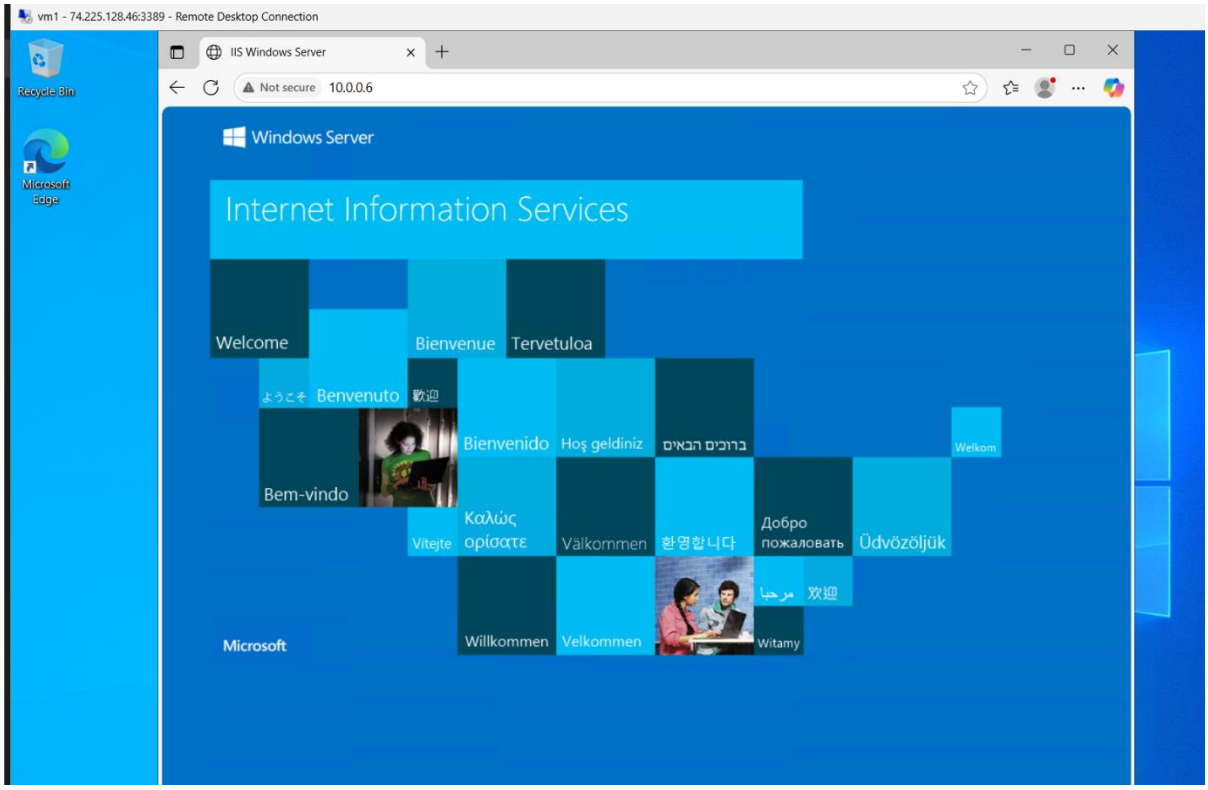


In this project, we configured the Load Balancing Rule with the following parameters:

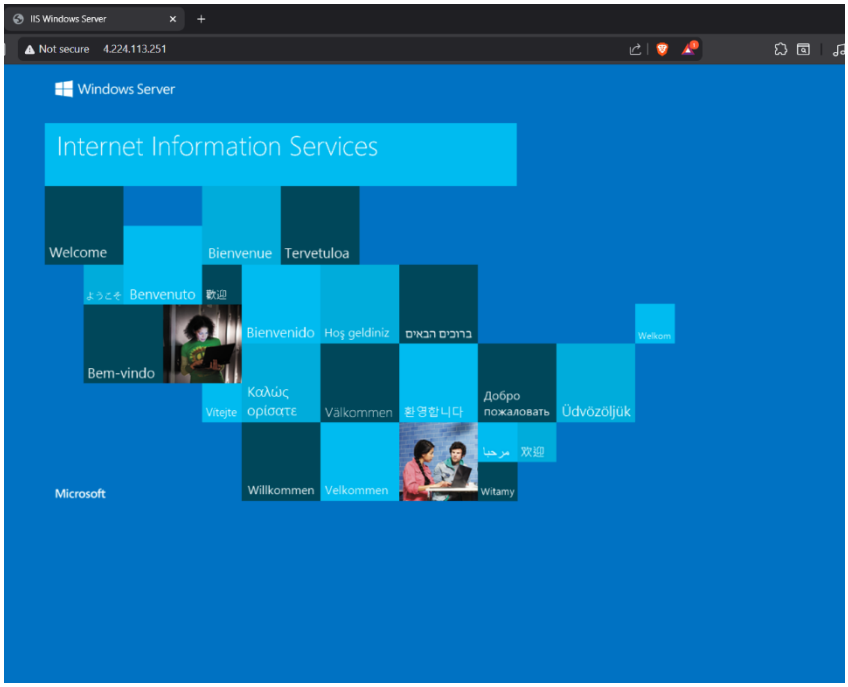
- **Frontend IP**: ext-lb-ip (for external load balancer)
- **Backend Pool**: Contains the two Windows VMs running IIS
- **Protocol**: TCP
- **Port**: 80 (frontend)
- **Backend Port**: 80
- **Floating IP**: Disabled
- **Session Persistence**: None
- **Idle Timeout**: Default

Testing Load Balancers:

- Public IP tested in browser showed the IIS page



- Internal Load Balancer private IP tested from within VM showed IIS page
  This validates both load balancers are functioning correctly.



Conclusion:

Successfully created and validated both Internal and External Load Balancers in Azure. This setup proves the capability of Azure LBs in distributing traffic efficiently between backend resources.