Load Balancer

# Introduction

This project demonstrates the external and internal load balancer in Microsoft Azure, which helps to distribute traffic across multiple virtual machines. The external load balancer was installed to process public internet traffic and to balance it between two windows-based virtual machines (VM1 and VM2), that were configured with a web server to reply back with different messages, and this will help us verify that load balancing is working as intended. On the other side, the internal load balancer was meant to handle private traffic within the virtual network, where another VM (VM3) was installed and that accessed the backend VMs (VM1 and VM2) via the IP address of the internal load balancer. This setup provided a secure, scalable and highly available traffic management inside as well as outside the network.

# Architecture Overview

In this project, a 3-tier architecture is implemented with the help of the Azure Virtual Machines and Load Balancers.

* VM1 (WindowsII - IIS) and VM2 (Linux - Apache) represent backend web servers, where each of them demonstrates a simple HTML page.
* VM3 which is Linux-based and it is used to access internal load balancers within the virtual network.
* An External Load Balancer (ELB) distributes HTTP traffic on the internet and then it directs this traffic to VM1 and VM2 using health probes.
* A load balancer is an Internal Load Balancer (ILB) which distributes the traffic between VM1 and VM2 privately and can only be accessed within the VNet (tested via VM3).
* HTTP and load balancer probe access were configured so that proper NSG rules and routing could be enabled.

# Virtual Network and Subnet Configuration

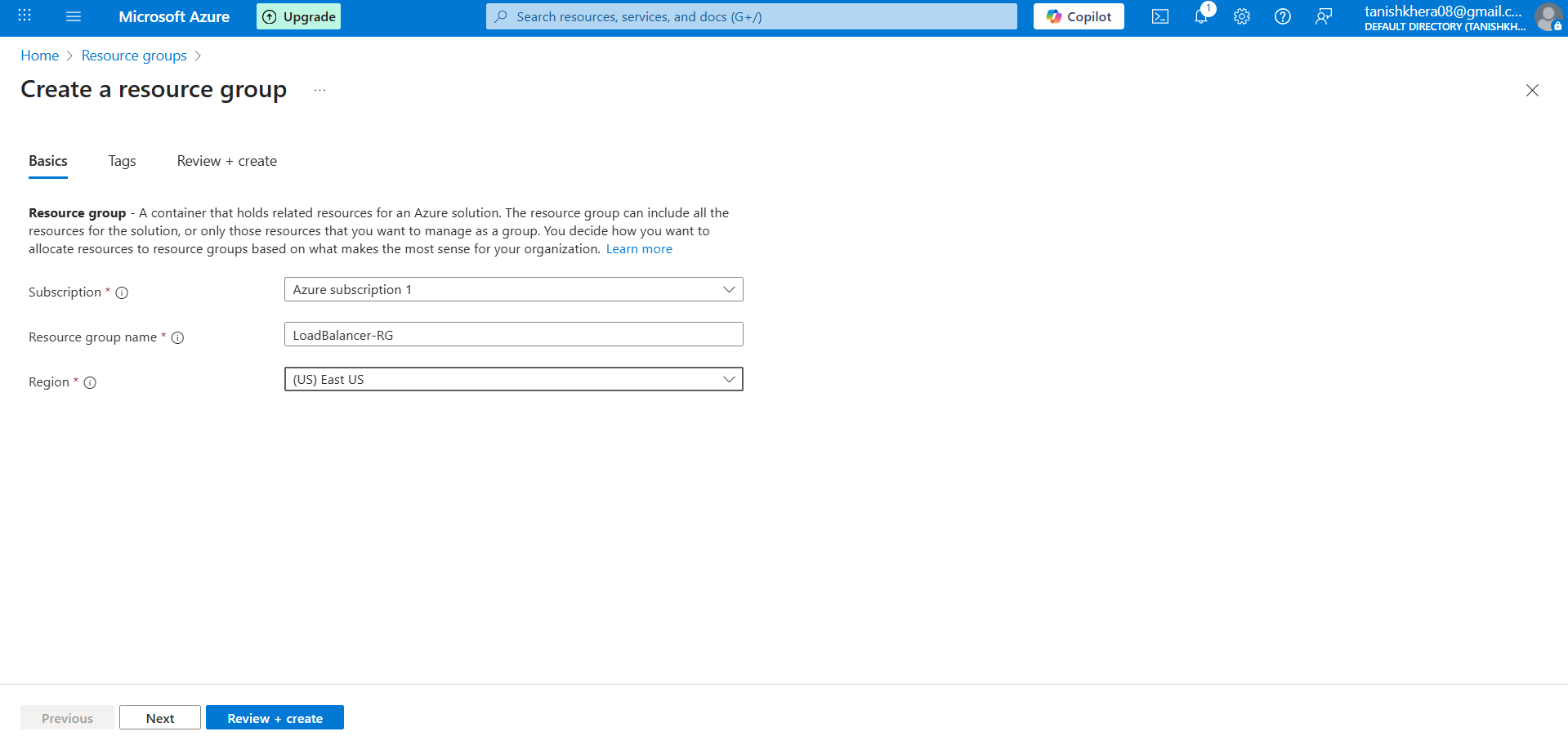
In order to assist load balancing configuration in Azure:

* One Resource Group was established in order to ensure that all components are managed.
* A Virtual Network (VNet) was created which has only one subnet, named Web Subnet (10.0.1.0/24), therefore it is hosting all the VMs.
* VM1/VM 2 are backend servers.
* Testing of the internal load balancer takes place with the help of VM3.

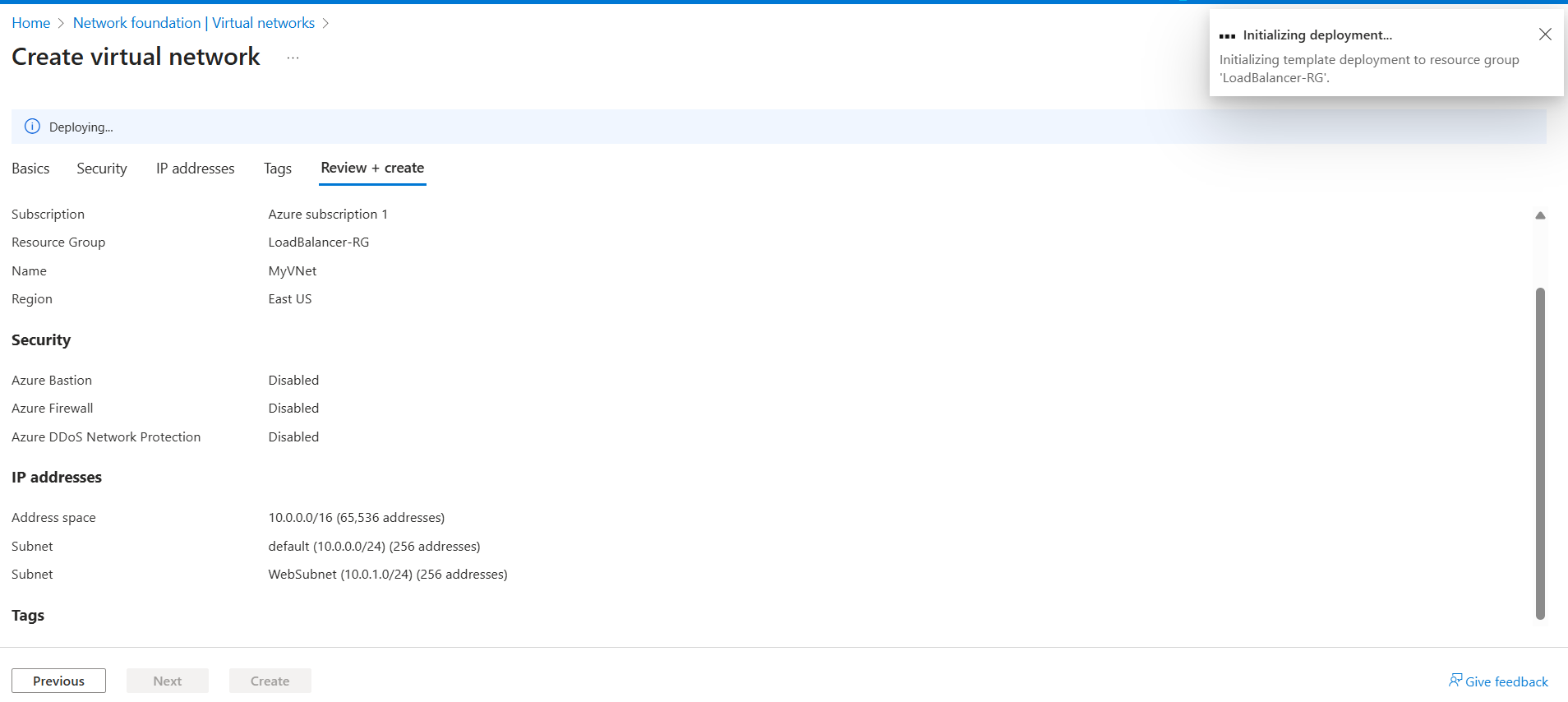
A Public IP was also provisioned to be connected to the External Load Balancer to have access through the internet.

This small configuration is based on internal as well as external load balancing in a single subnet environment.

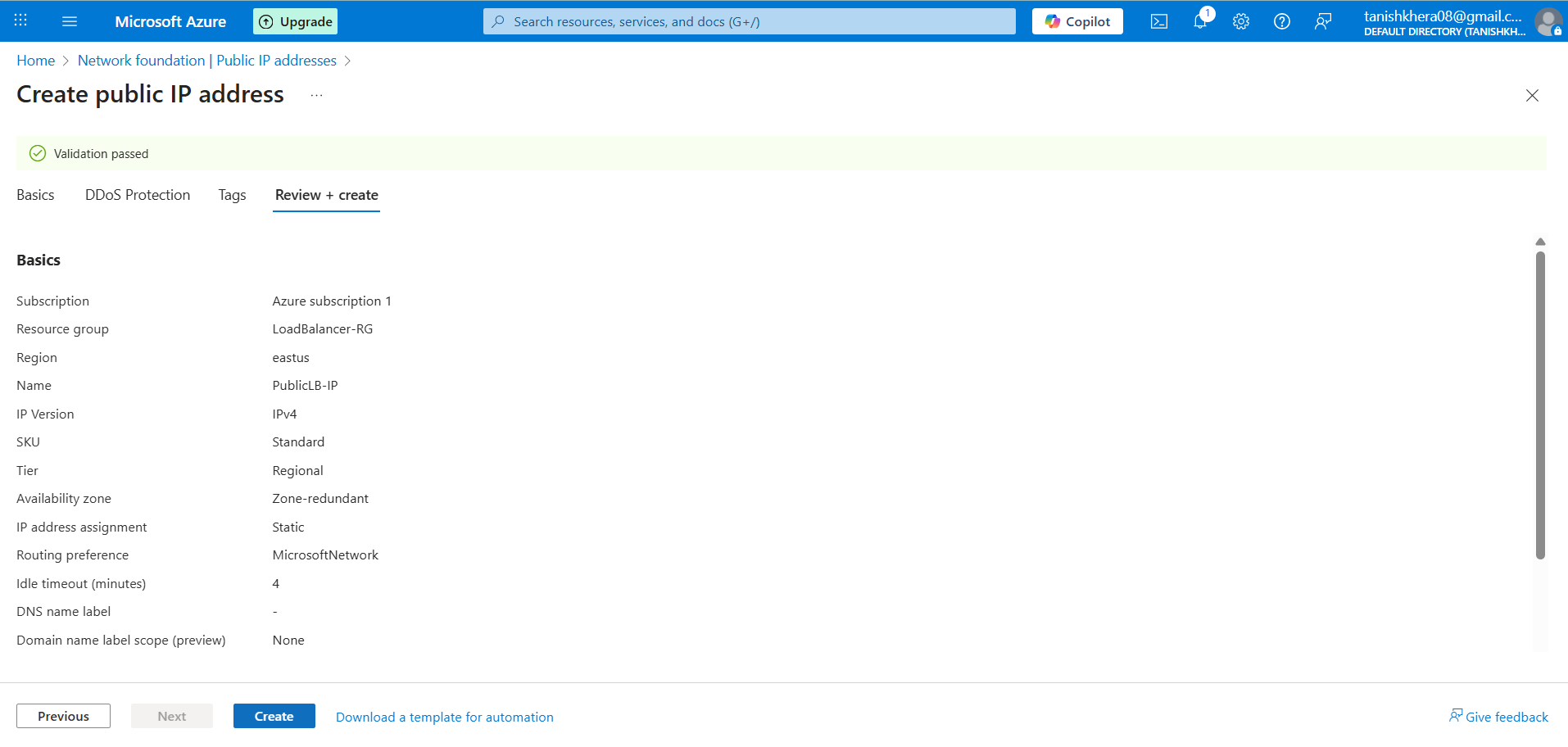
## Resource Group



Vnet and a Subnet



Public IP



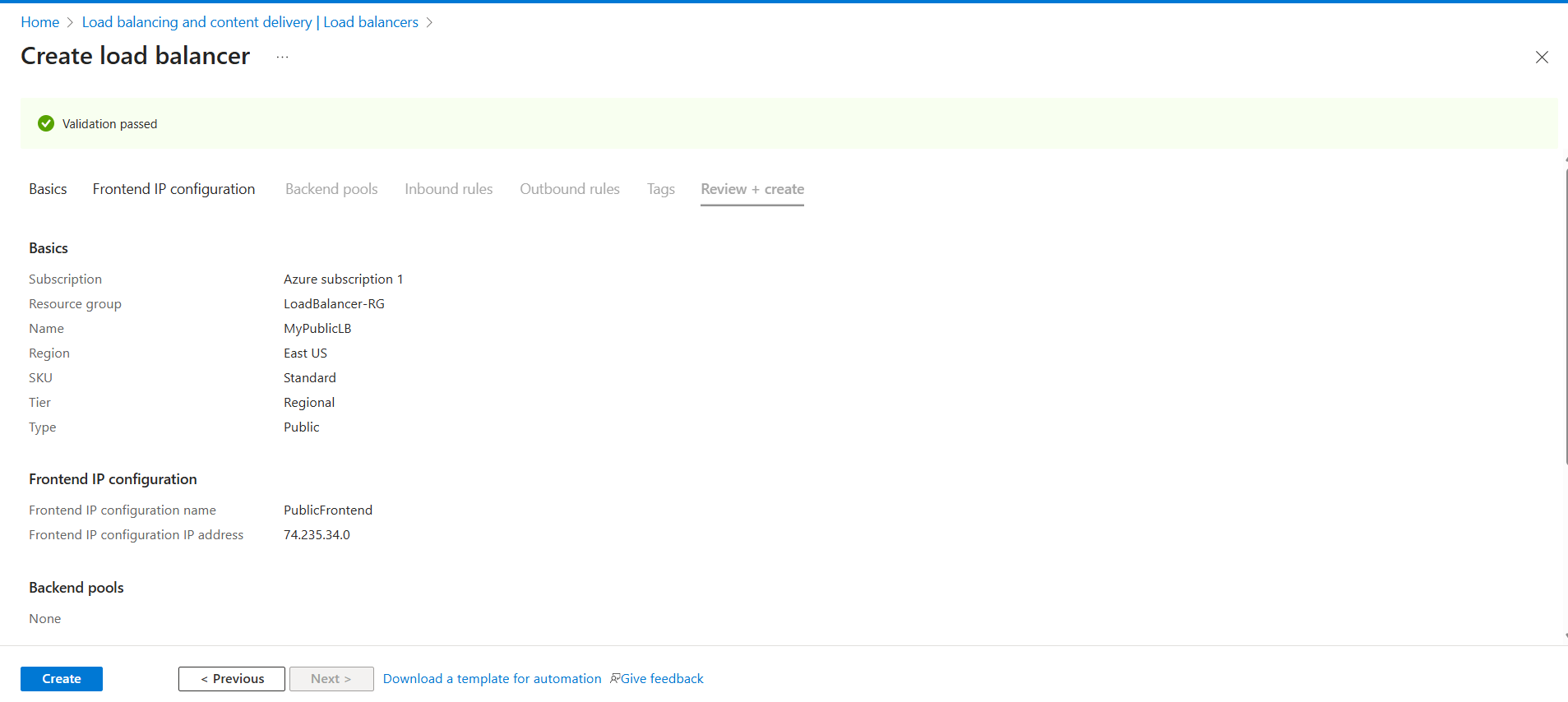
# External Load Balancer Configuration

External Load Balancer was set up to distribute traffic flow from the internet to back end virtual machines. It is configured as follows:

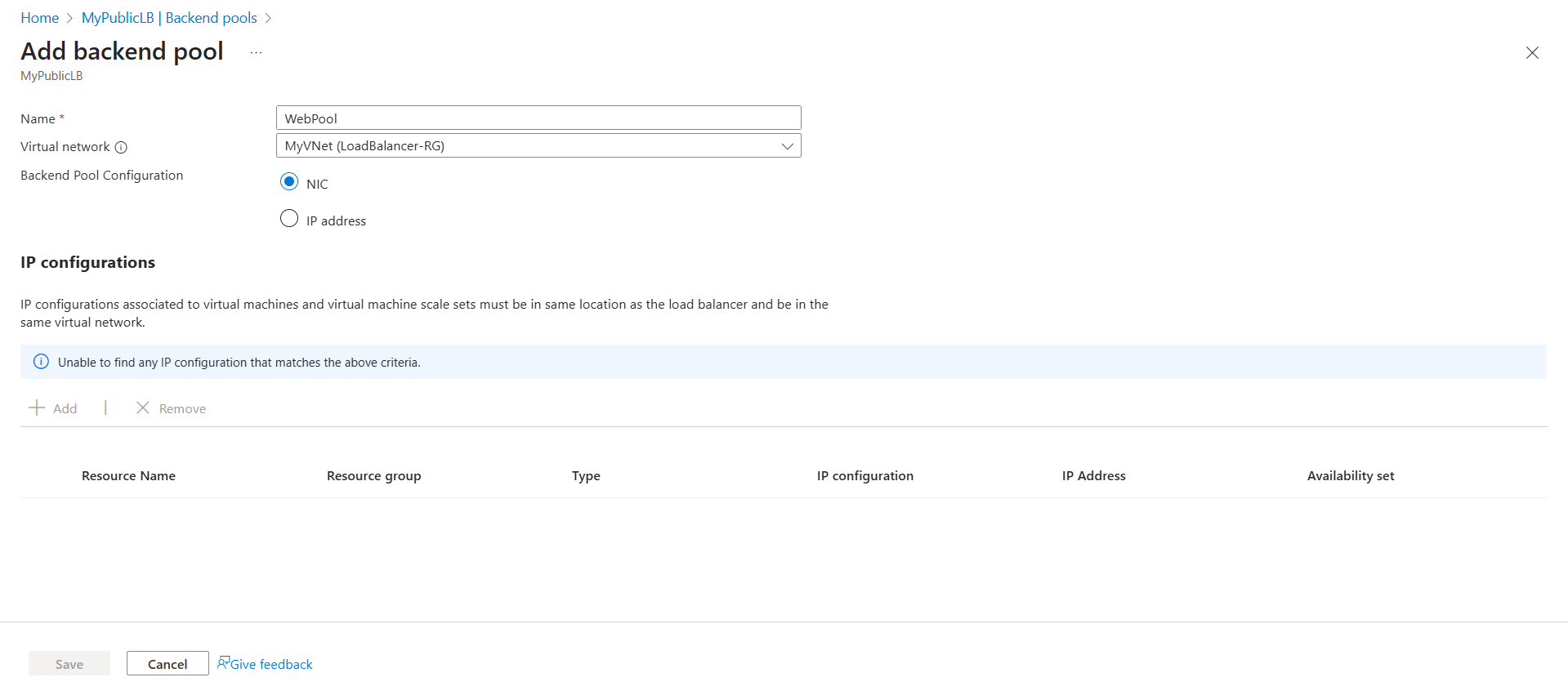
* A zone-redundant Public IP address was designed to resemble high availability within the zones.
* The above public IP was used to define Frontend IP configuration.
* A Backend Pool was established and this pool had VM1 and VM2, and both are hosted on Zone 2.
* A Health Probe was configured so that it checks the availability of the backend VMs on port 80.
* A Load Balancing Rule was established to route HTTP traffic (port 80) between the frontend to the backend pool in an evenly distributed way between VM1 and VM2.

Such arrangement allows external clients to access services on the backend VMs using the public IP endpoint.

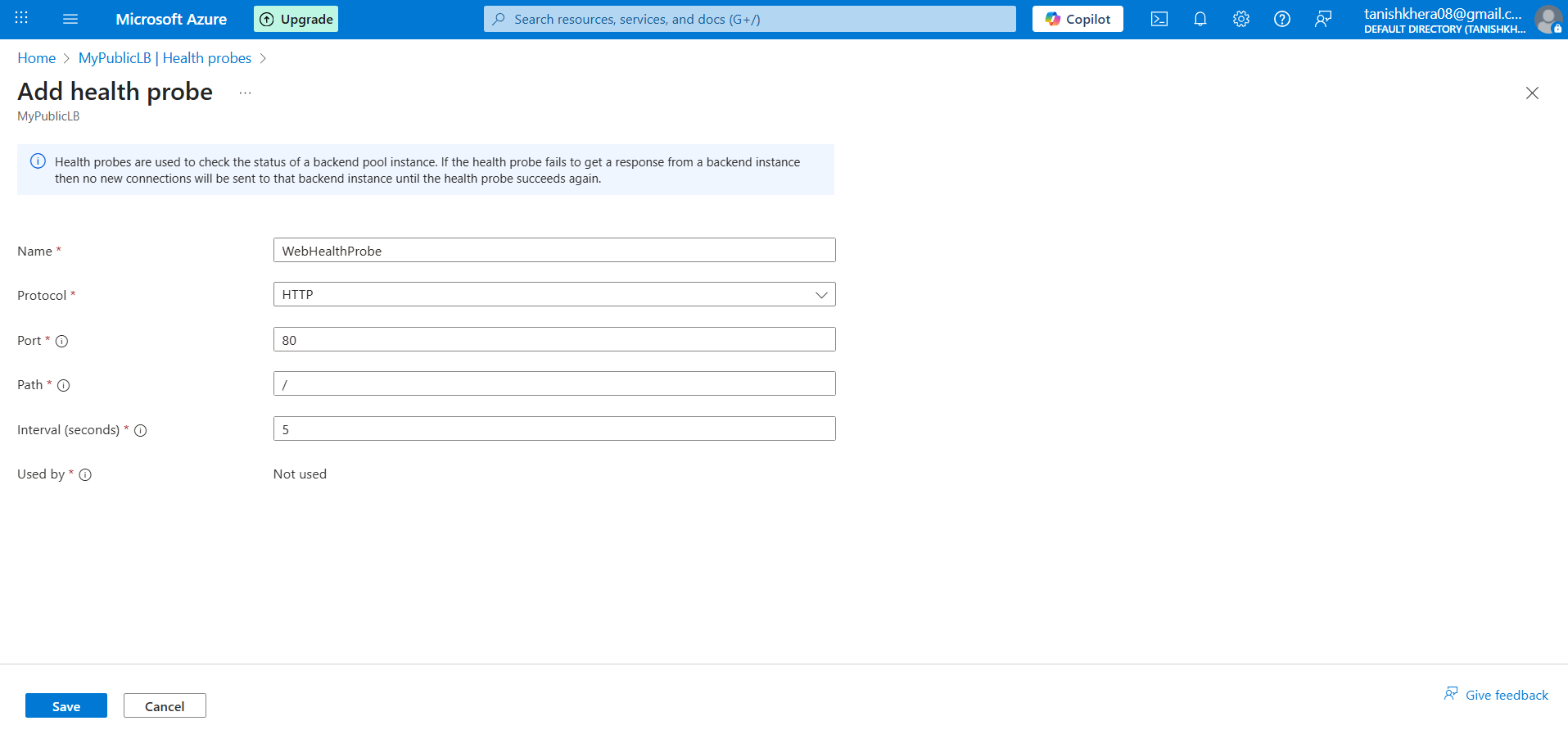
External Load Balancer



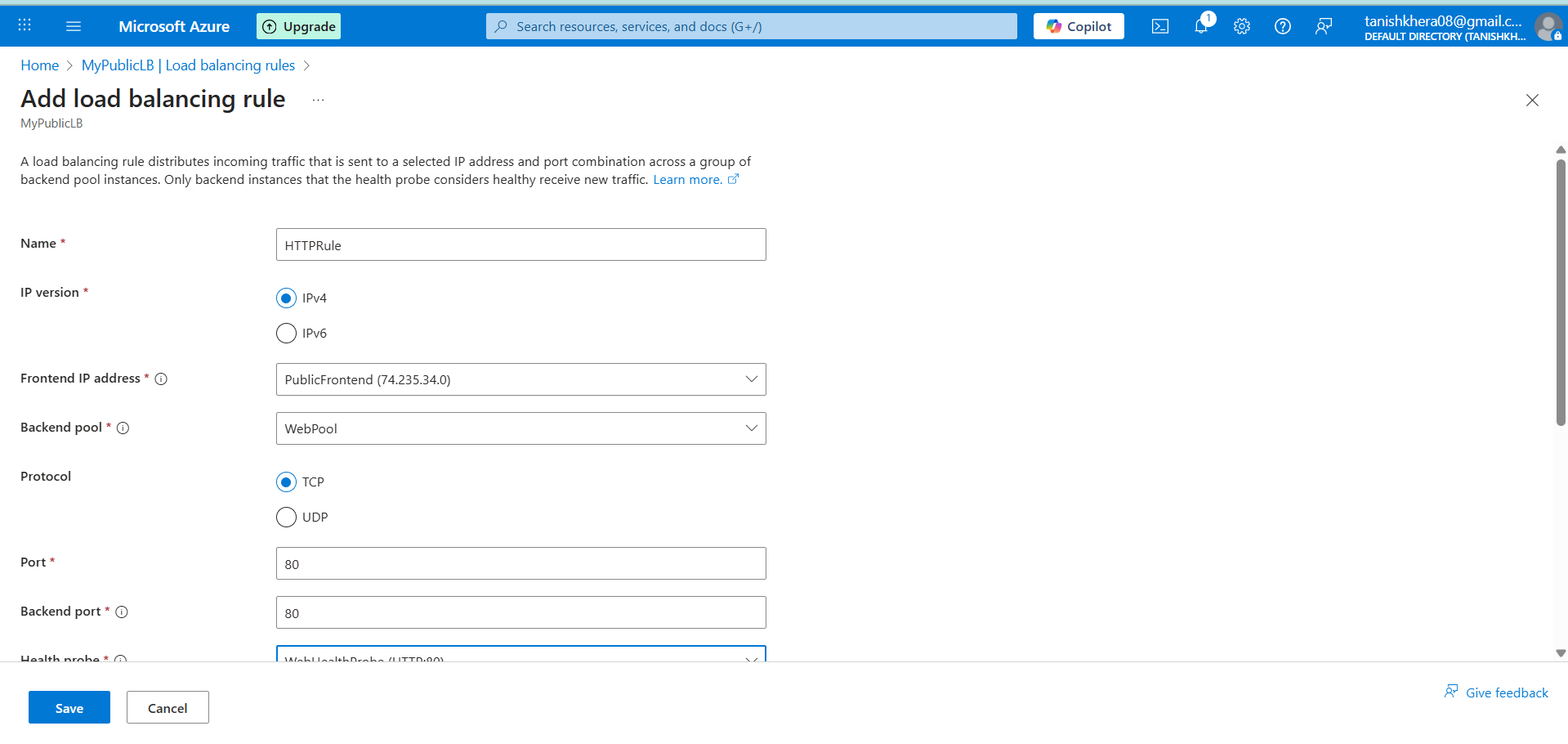
Backend Pool in Load Balancer



Adding Health Probe



Load Balancing Rule to forward traffic



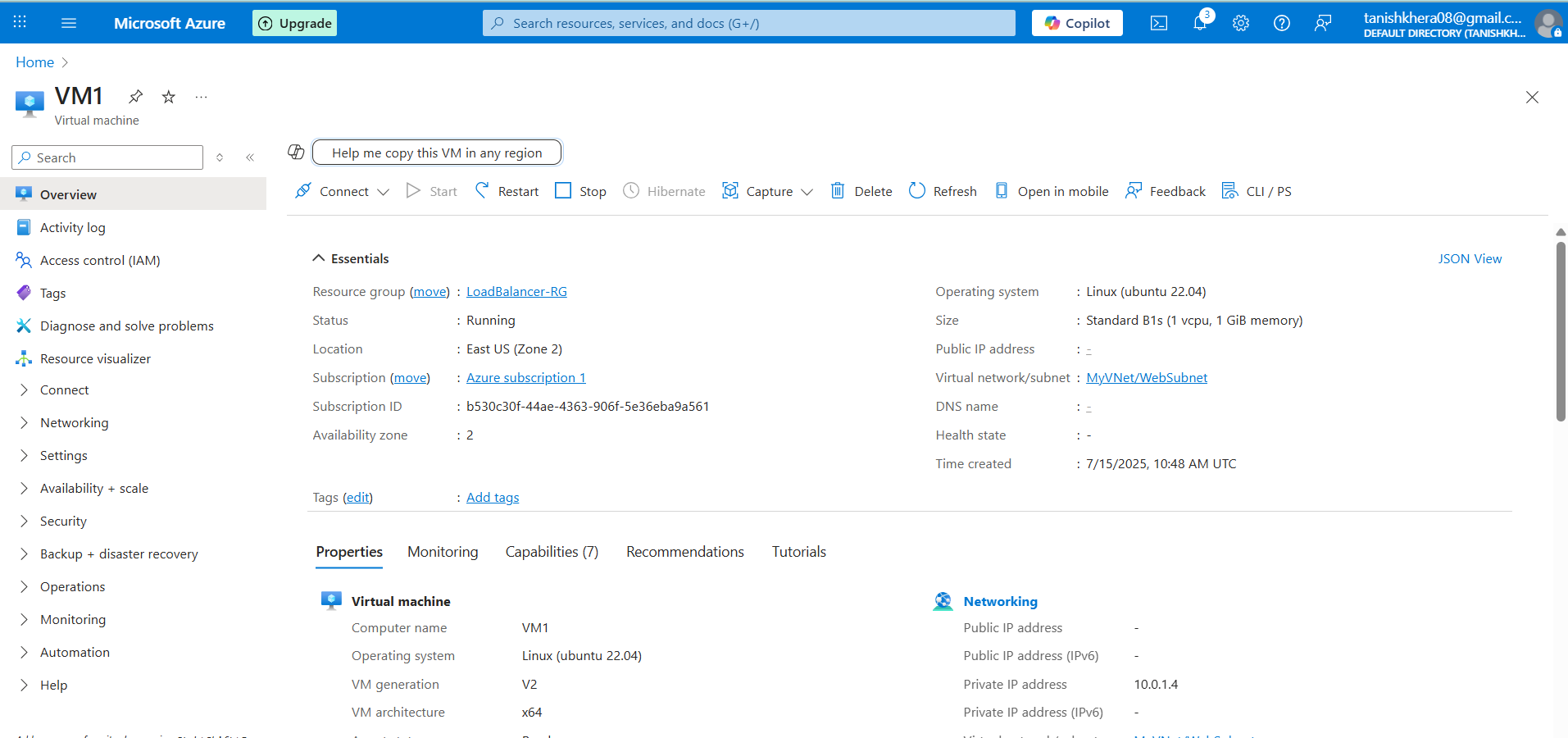
# Virtual Machine

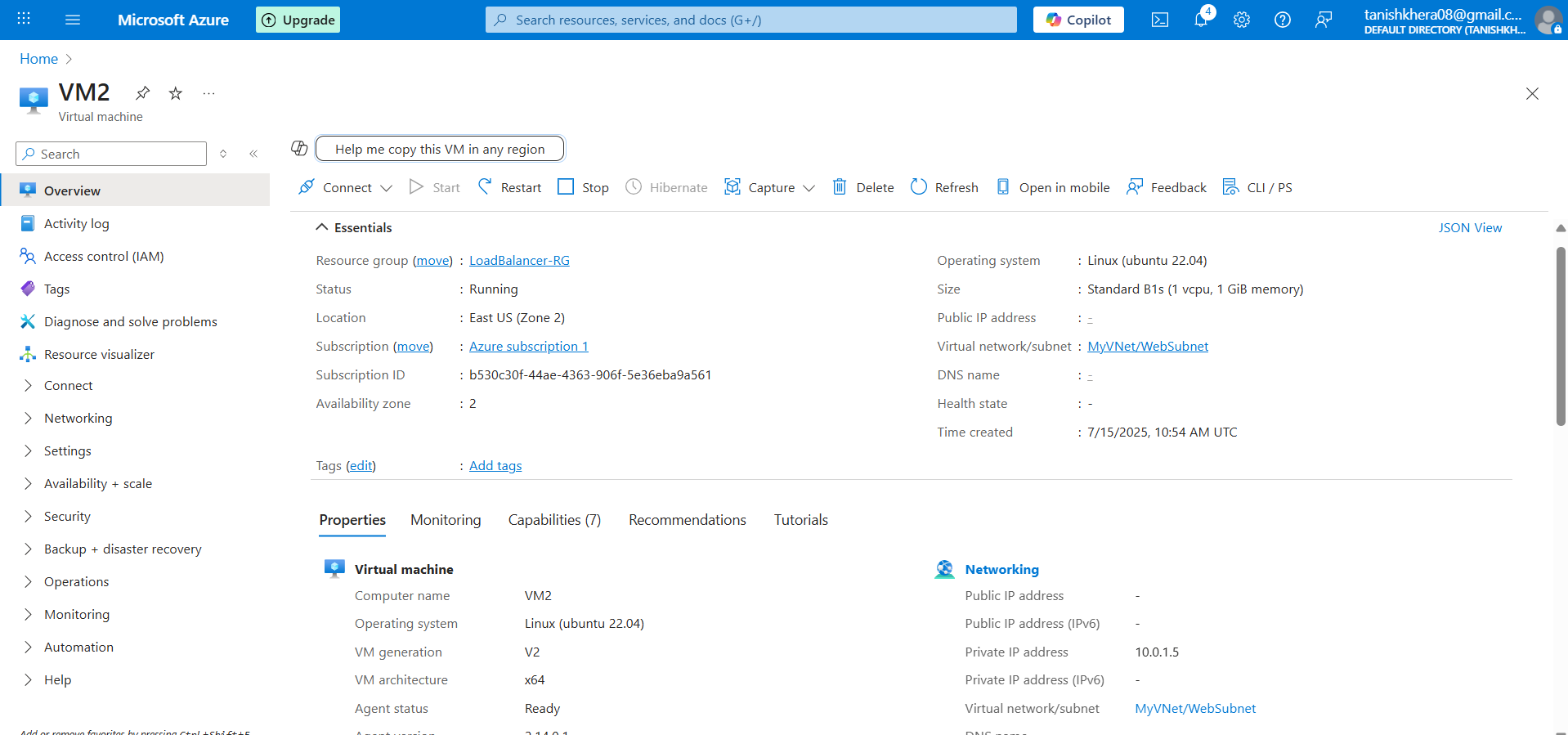
There were two backend VM which were provisioned to process the traffic:

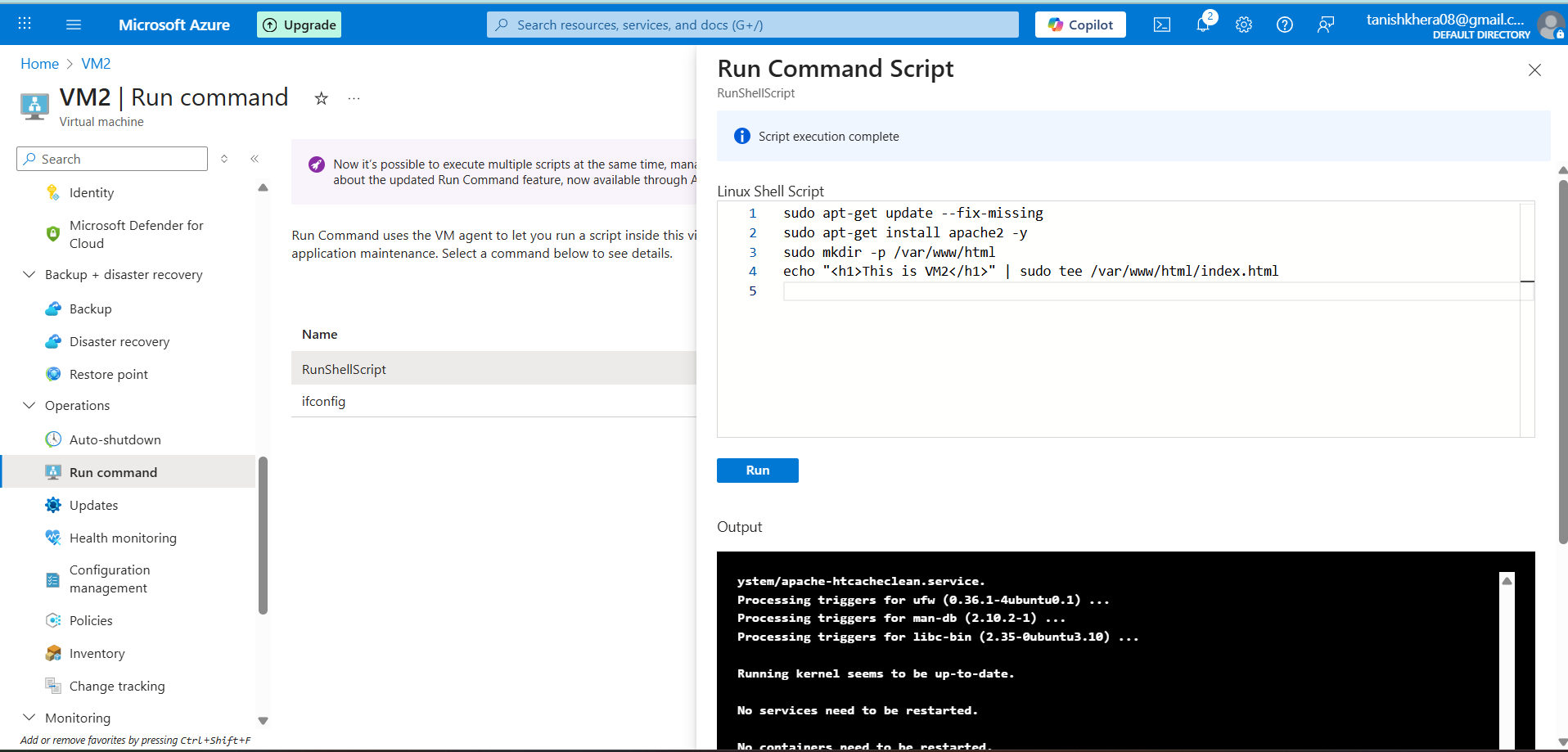
* VM1 is deployed in the Web Subnet, therefore configured with windows server. Basic HTML page for verification was used by installing IIS (Internet Information Services).
* In the same subnet, VM2 was deployed with a Linux based OS (Ubuntu). Apache HTTP Server was installed here and it is configured to give a custom response.

In the same availability zone, both the VMs were deployed and their NIC was attached to the backend pool of the external load balancer.

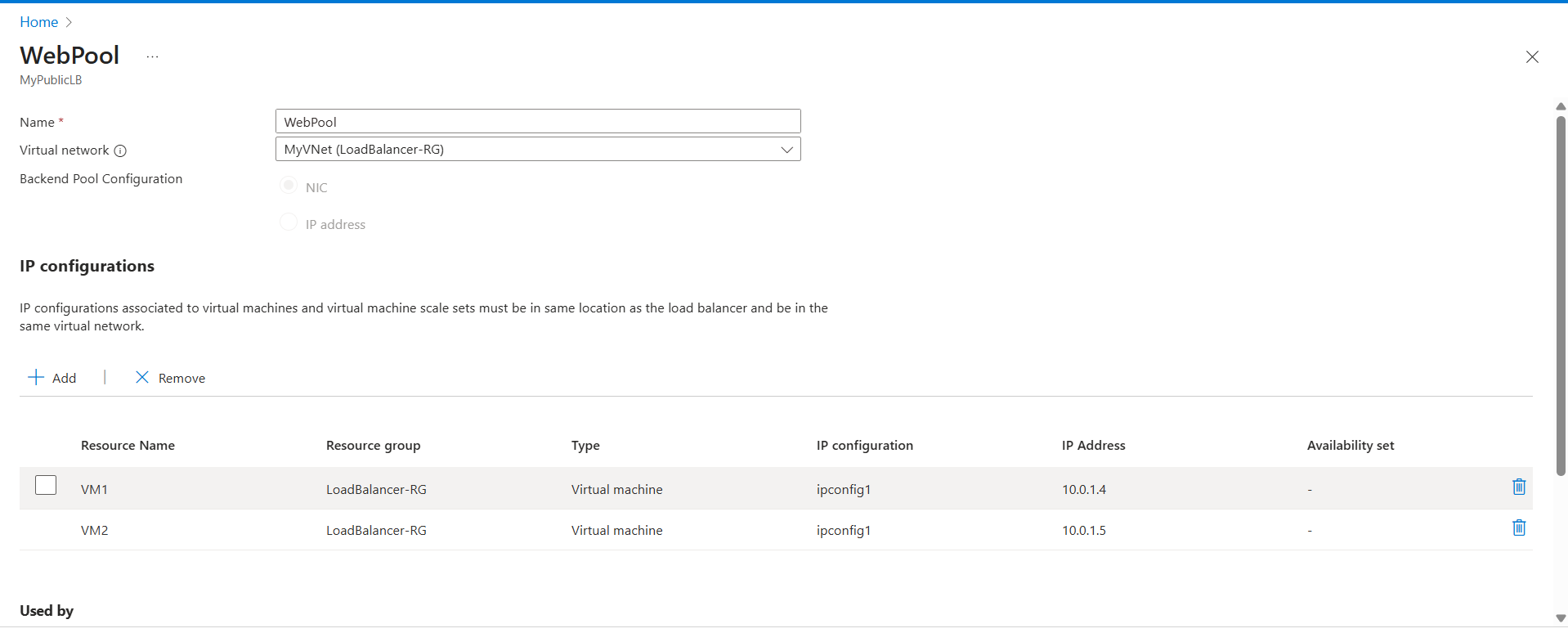
Virtual Machine 1 and 2



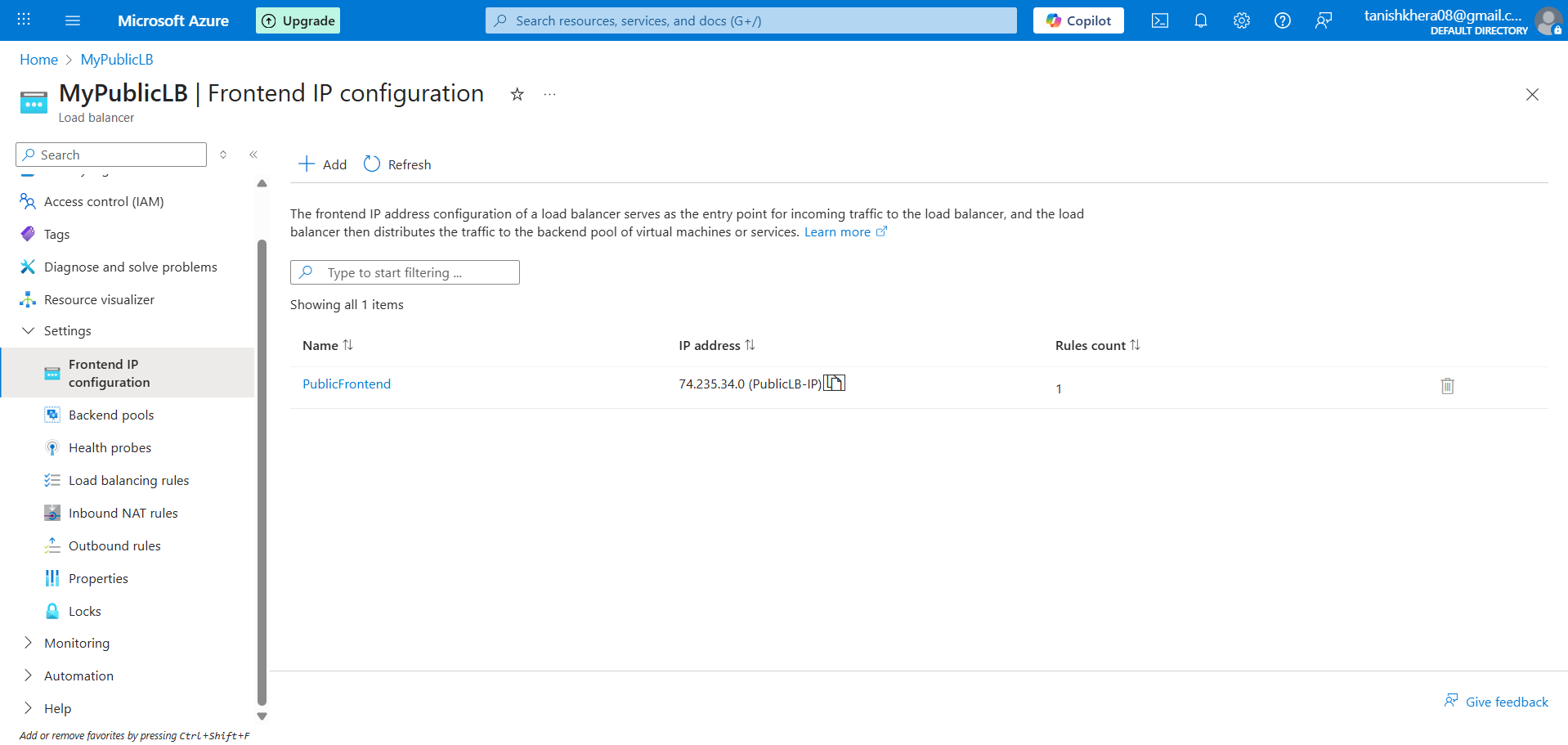


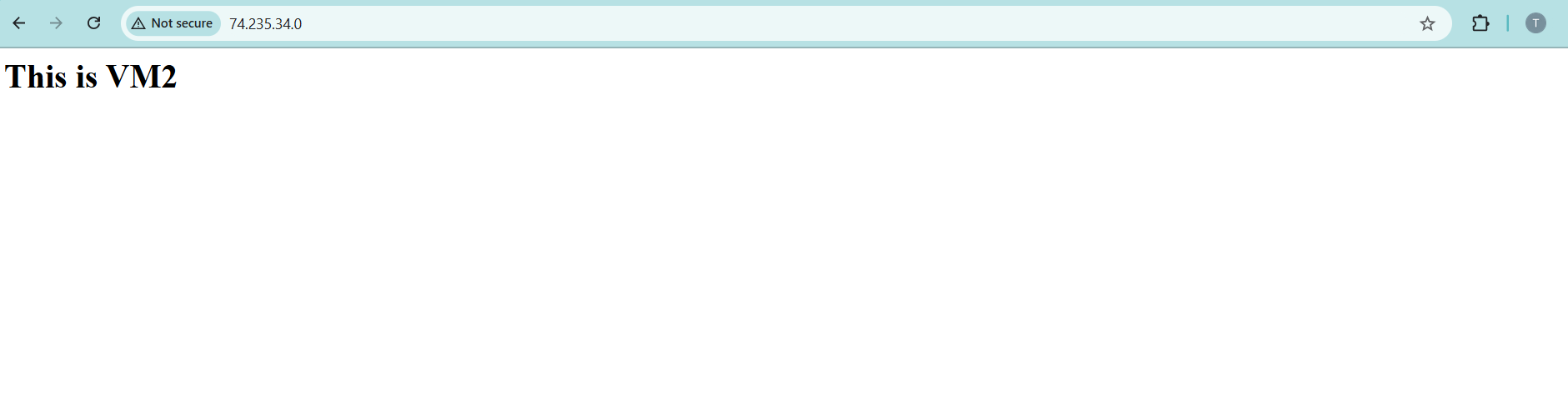


Attached VM NIC to Backend Pool



Testing Public IP of External Load Balancer





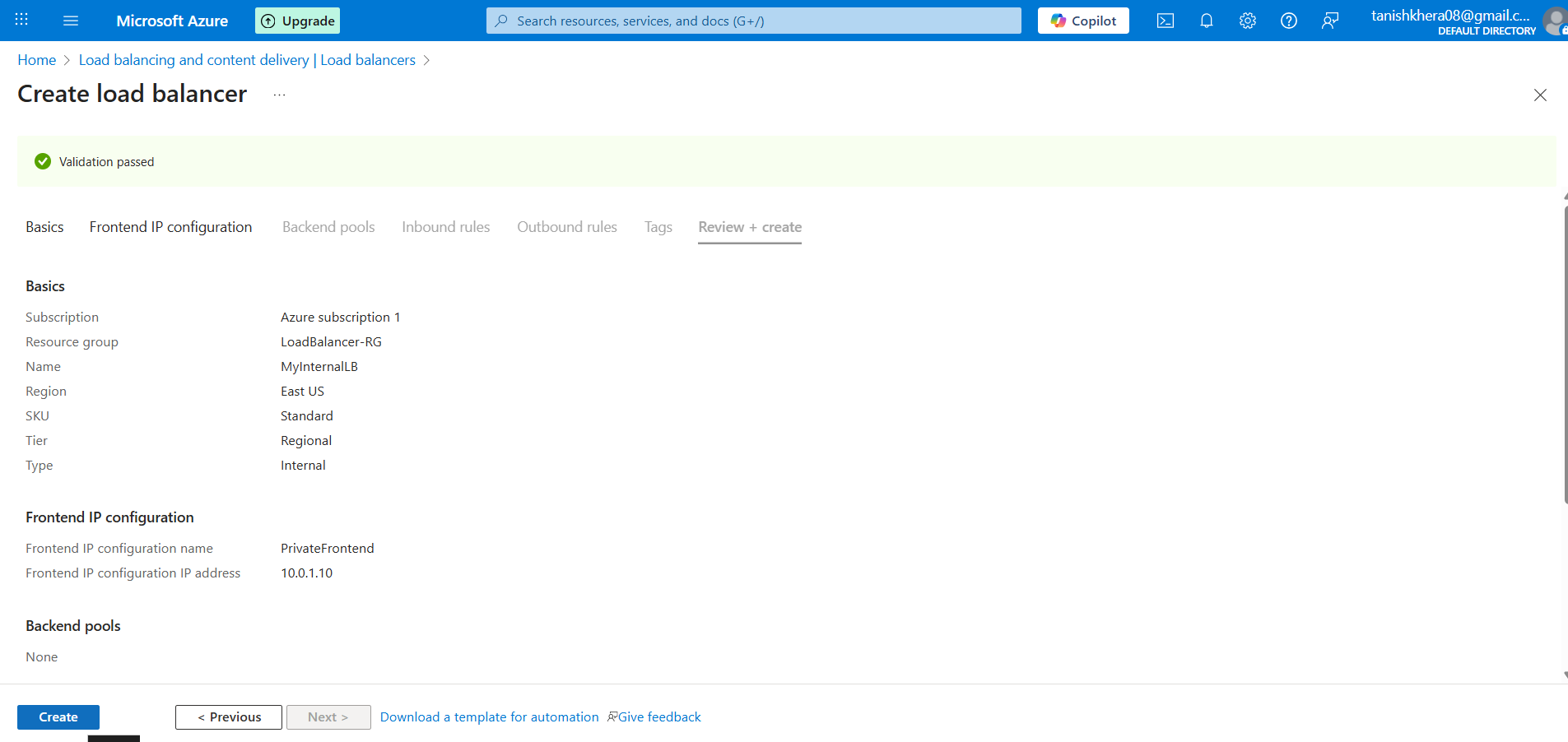
**We were successfully able to get the response when load balancer IP was accessed in the browser: This is VM2 which ensures the functionality of load balancer.**

# Internal Load Balancer

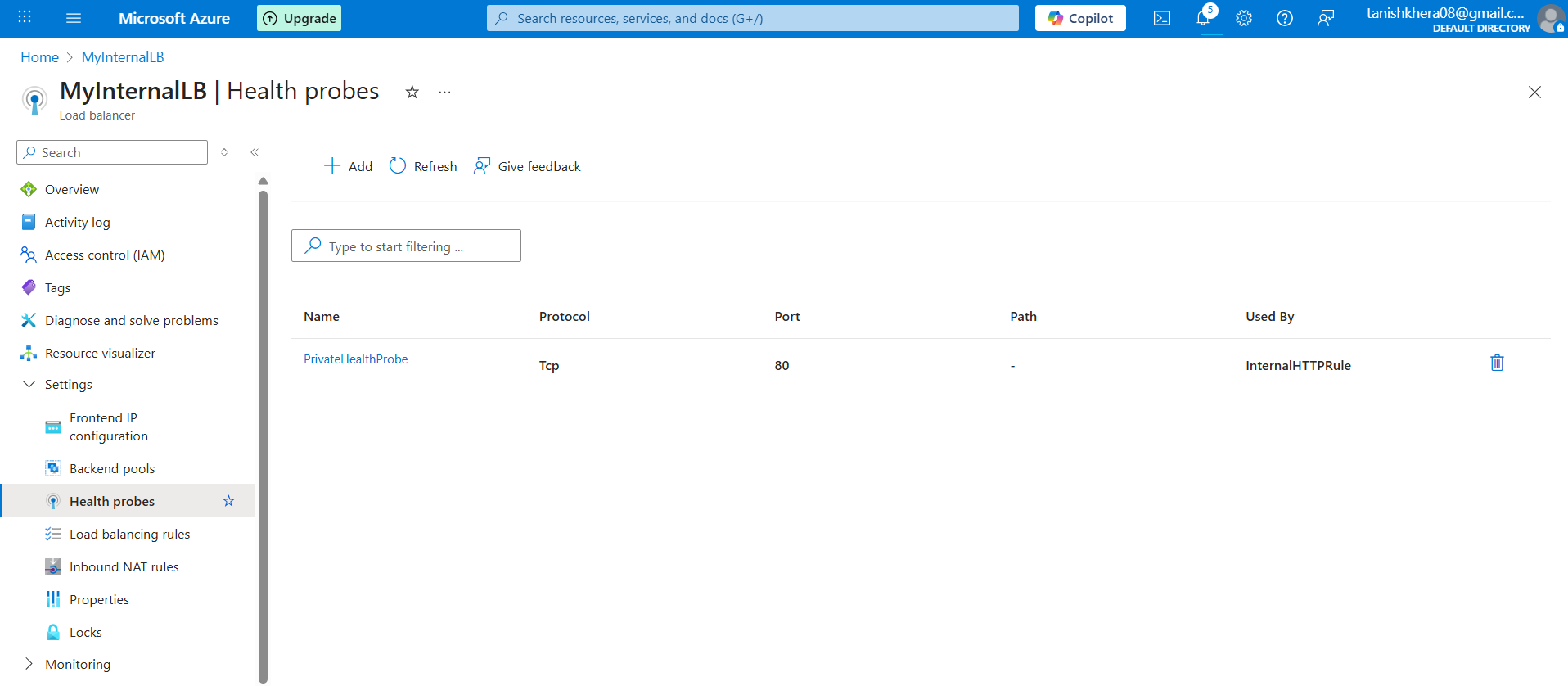
In order to facilitate the use of traffic distribution in a private network, an Internal Load Balancer was installed with the following elements:

* There is also allocation of private IP address for internal load balancer within the Web Subnet(10.0.1.0/24).
* A Backend Pool was made including the NIC of VM1 and VM2 and configured to act as an application server internally.
* A Health Probe on port 80 was defined that checks the availability of backend virtual machines and also monitors their health.
* To redirect the internal HTTP web traffic (port 80) from frontend IP to the backend pool of servers, a Load Balancing Rule was created.

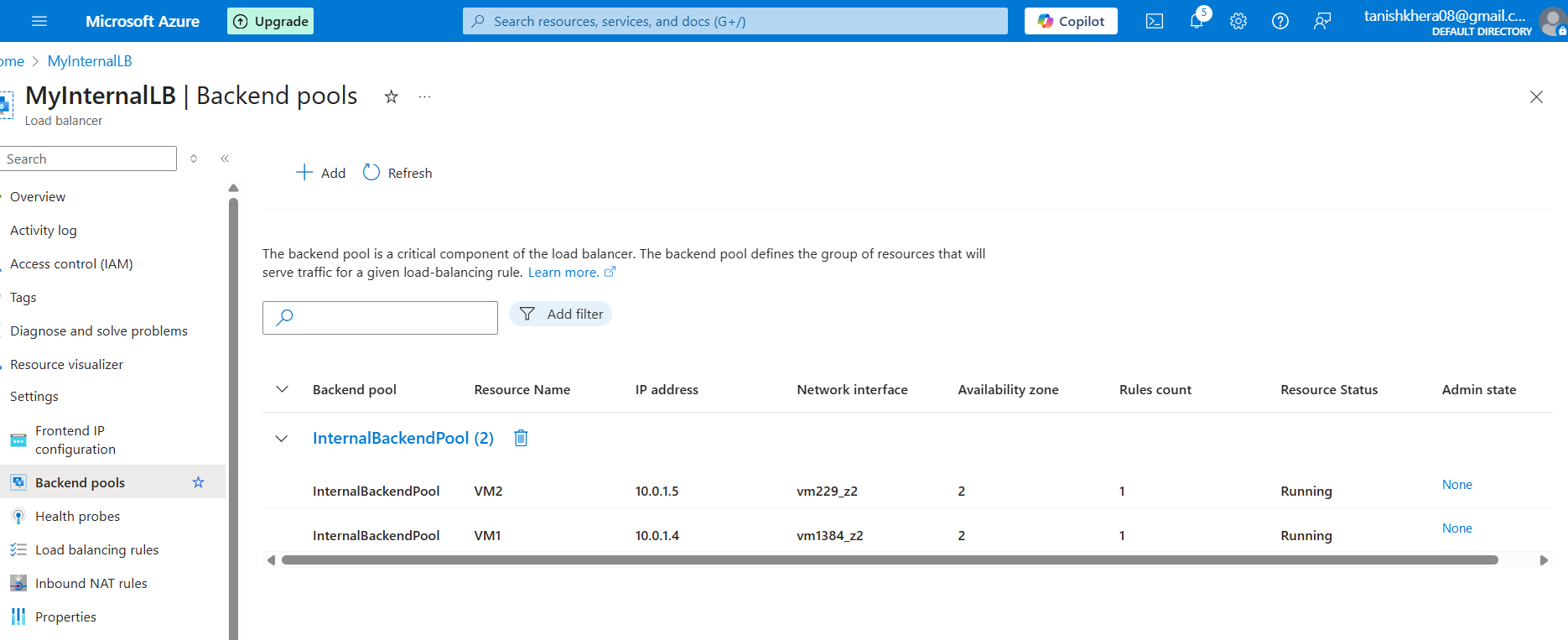
Internal Load Balancer and its IP Configuration



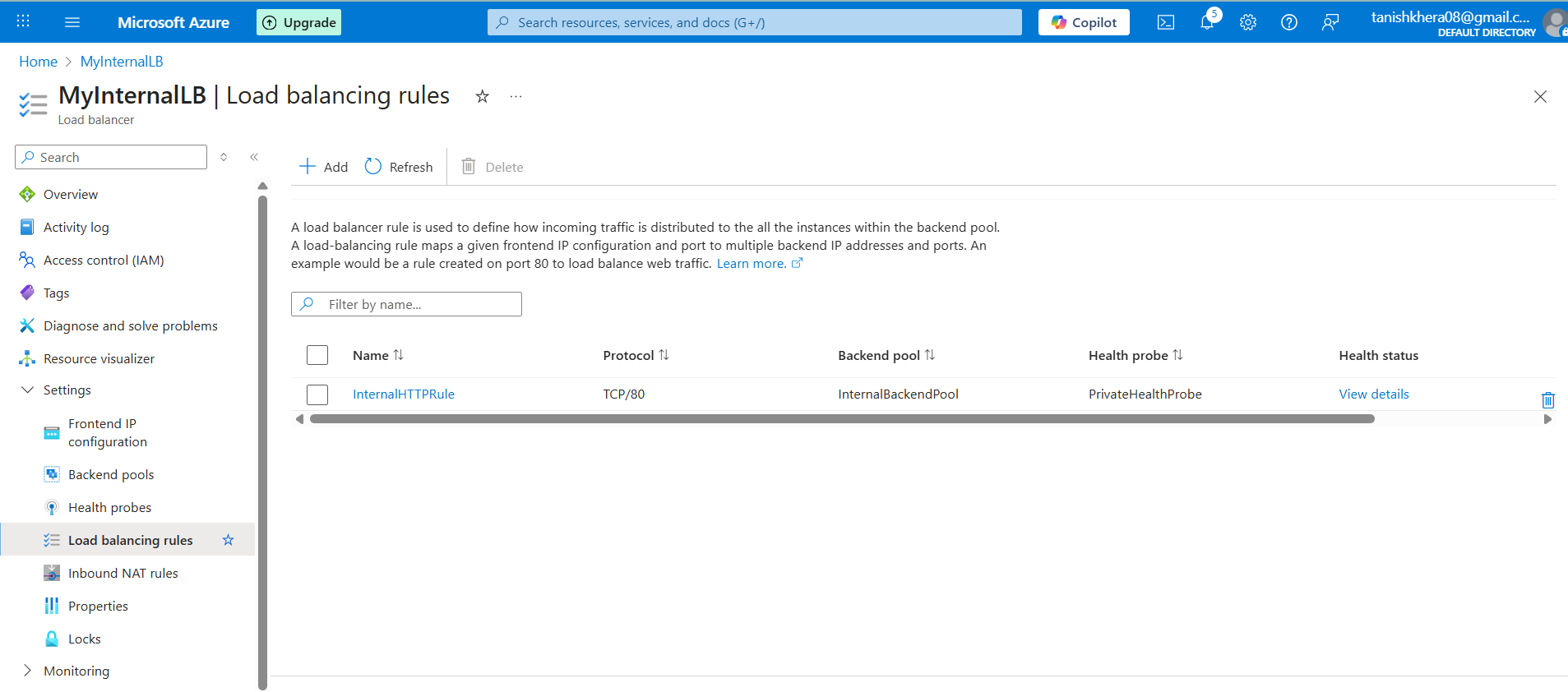
Adding Health Probe



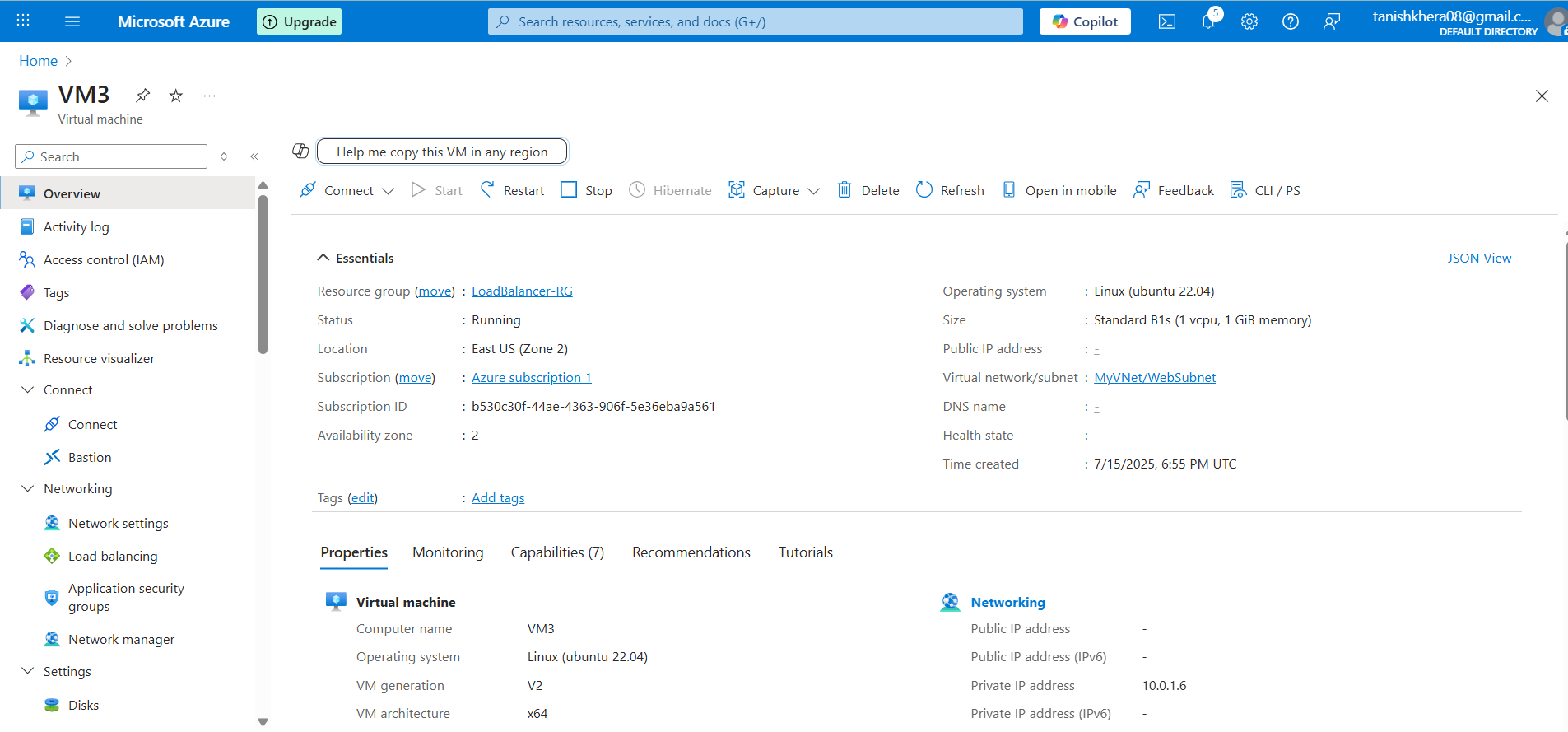
Backend Pool



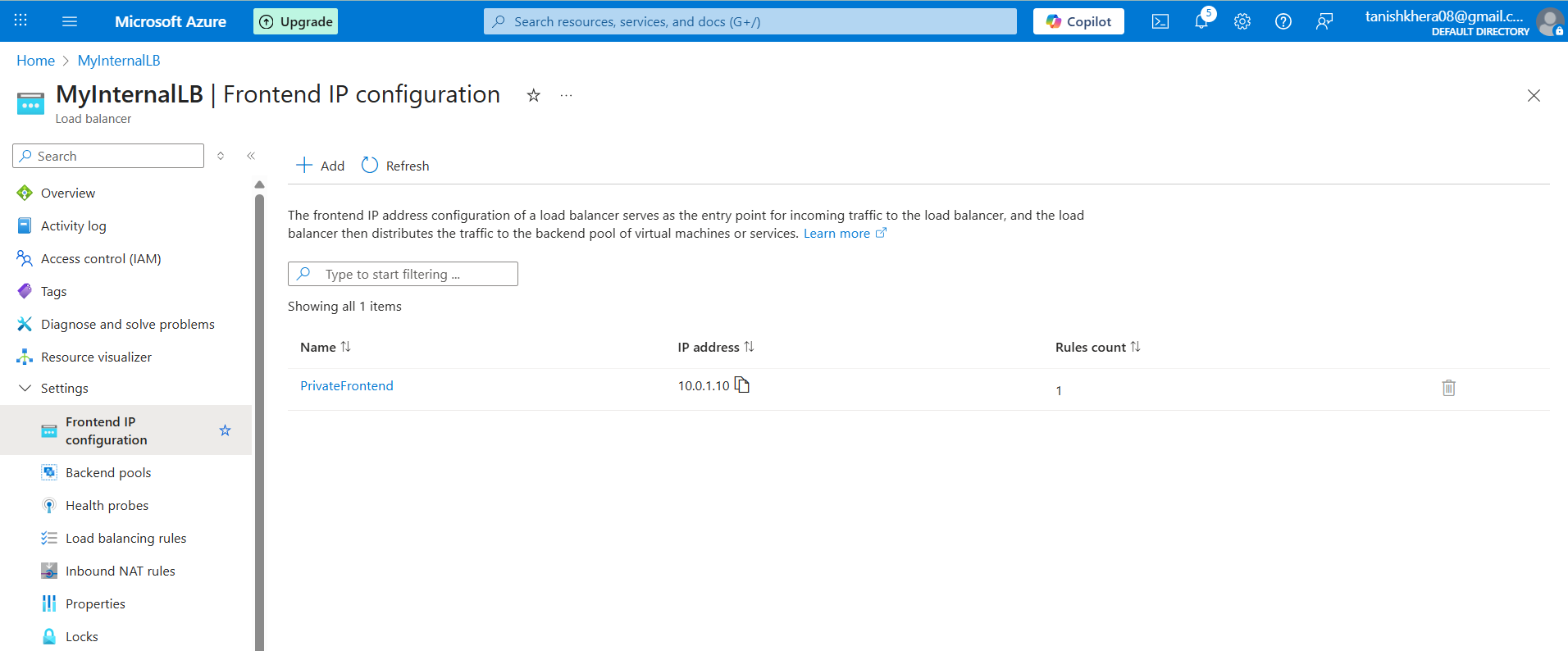
Load Balancing Rule for Internal Load Balancer



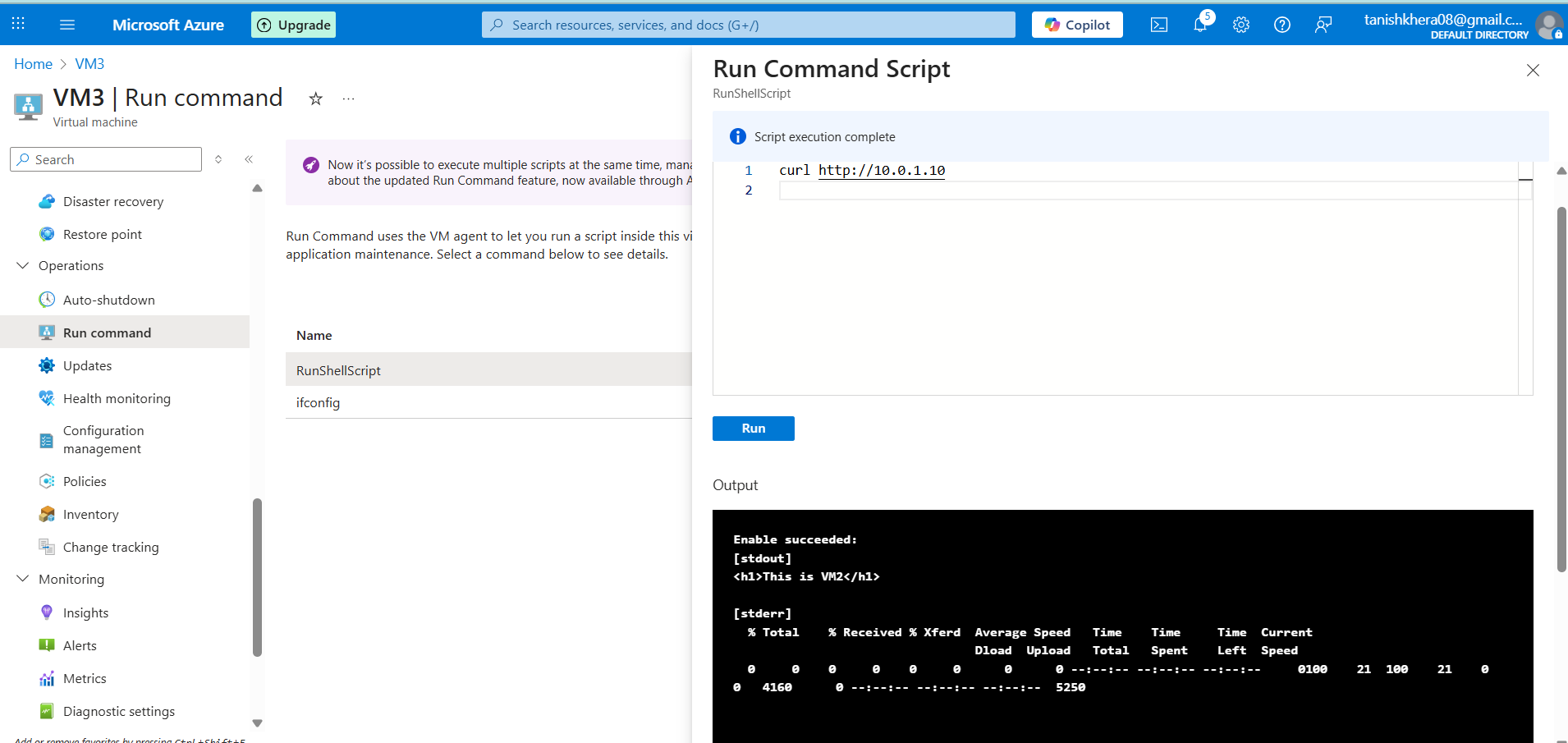
Virtual Machine 3



Frontend Private IP



Testing and Verification



**To prove the validity of the internal load balancer, VM3 (installed into the same virtual network) was utilized to link to the private front end IP of the load balancer in a web browser. Distribution of the traffic was successful as the test displayed traffic served by VM2 i.e. <h1>This is VM2</h1>.**

# Conclusion

The use of the External and Internal Load Balancers in Azure proved effective distribution of the network traffic across virtual machines. The external load balancer provided high availability and balanced the HTTP requests between VM1 and VM2 with zone-redundant public IP. Internal Load Balancer offered intra network traffic distribution of which the validation was through VM3 as a test client. It is a good use case demonstrating the powerful load balancing features of Azure that can deliver scalable, fault-tolerant and highly available application designs.