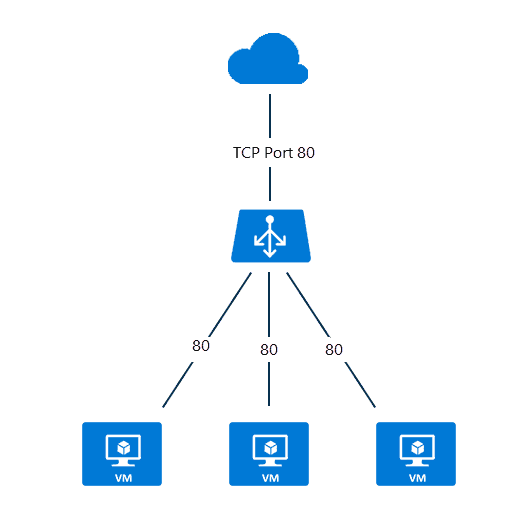
**Azure Load Balancers**

**Naman Moolri CEQ-525**

**Introduction**

**Azure Load Balancer** allows you to distribute traffic to your backend virtual machines. An Azure load balancer provides high availability for your application. The Azure load balancer is a fully managed service itself.



**Types of Load Balancers**

A **public load balancer** can provide outbound connections for virtual machines (VMs) inside your virtual network. These connections are accomplished by translating their private IP addresses to public IP addresses. Public Load Balancers are used to load balance internet traffic to your VMs.

An **internal (or private) load balancer** is used where private IPs are needed at the frontend only. Internal load balancers are used to load balance traffic inside a virtual network. A load balancer frontend can be accessed from an on-premises network in a hybrid scenario.

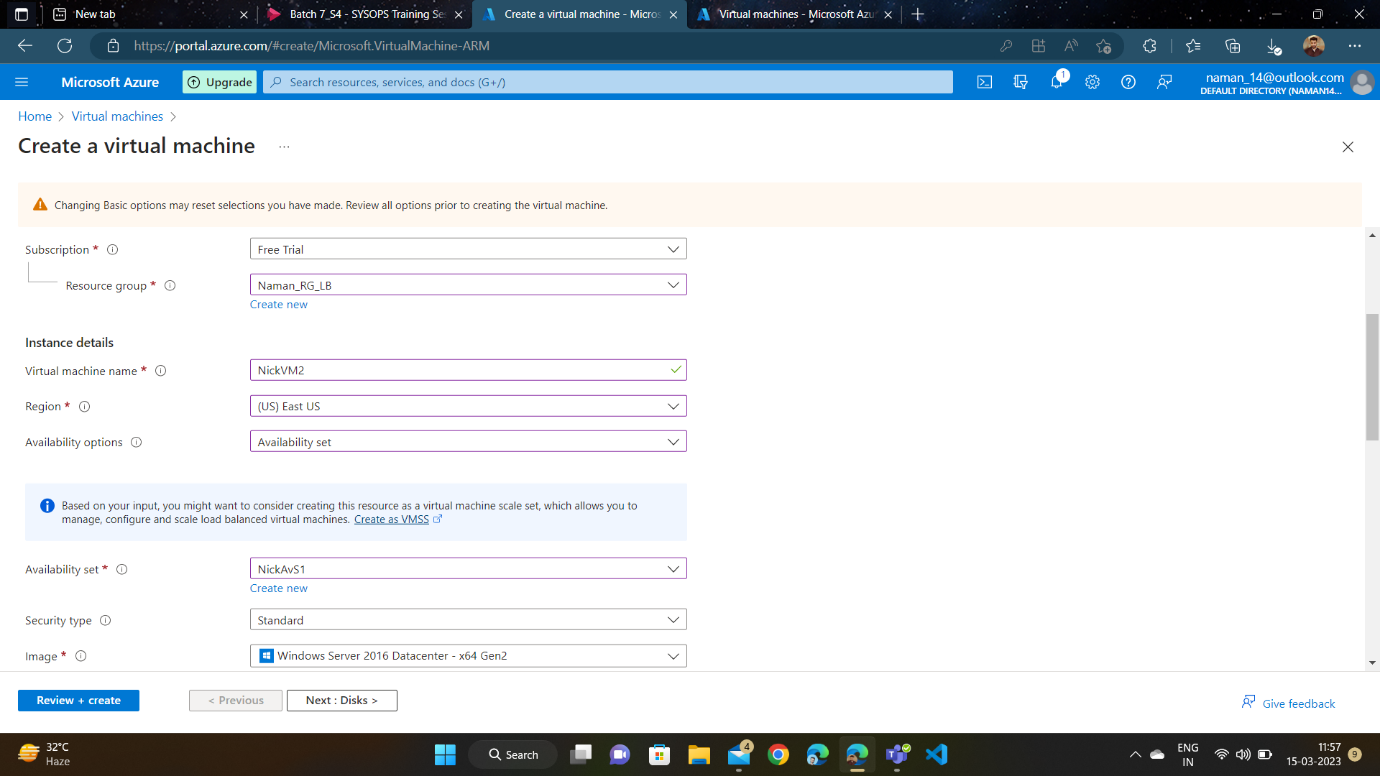
**Why do we need it?**

With Standard Load Balancer, you can scale your applications and create highly available services. Load balancer supports both inbound and outbound scenarios. A load balancer provides low latency and high throughput and scales up to millions of flows for all TCP and UDP applications.  
Some of the key scenarios that you can accomplish using Standard Load Balancer include:

* Load balance internal and external traffic to Azure virtual machines.
* Increase availability by distributing resources within and across zones.
* Use health probes to monitor load-balanced resources.
* Employ port forwarding to access virtual machines in a virtual network by public IP address and port.
* Standard Load Balancer provides multi-dimensional metrics through Azure Monitor. These metrics can be filtered, grouped, and broken out for a given dimension. They provide current and historic insights into the performance and health of your service. Resource Health is also supported.
* Load balance services on multiple ports, multiple IP addresses, or both.

**Steps for making an Azure Load Balancer**

1. **Creating two virtual machines using Availability Set, B2s Storage, and adding RDP, HTTP and HTTPs to inbound rules.**

****

1. **Two VMs are created.**

**A screenshot of a computer

Description automatically generated**

1. **Configuring the Server Manager by adding roles and features and then installing IIS on it for both VMs.**

**Graphical user interface, application

Description automatically generated**

1. **Configuration is done for both VMs.**

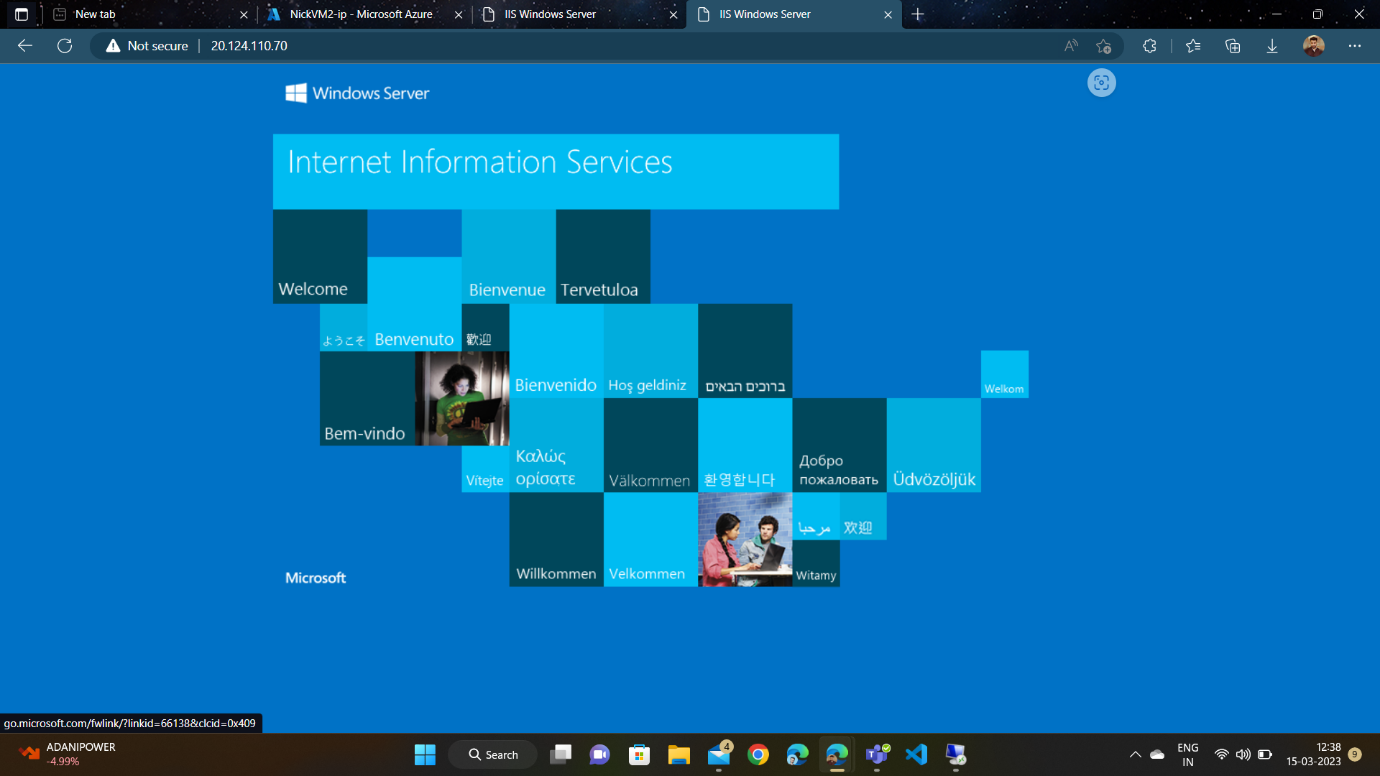
**Graphical user interface, application

Description automatically generated**

1. **Disabling the firewalls for both VMs.**

****

1. **Accessing our VMs through public IPs.**

****

1. **In order to differentiate between our VMs, we’ll make changes in the HTML file of both the VMs.**

**Graphical user interface, application

Description automatically generated**

1. **Adding the headers to the body in order to differentiate between two VMs as those headers will display on our VM interface.**

**A screenshot of a computer

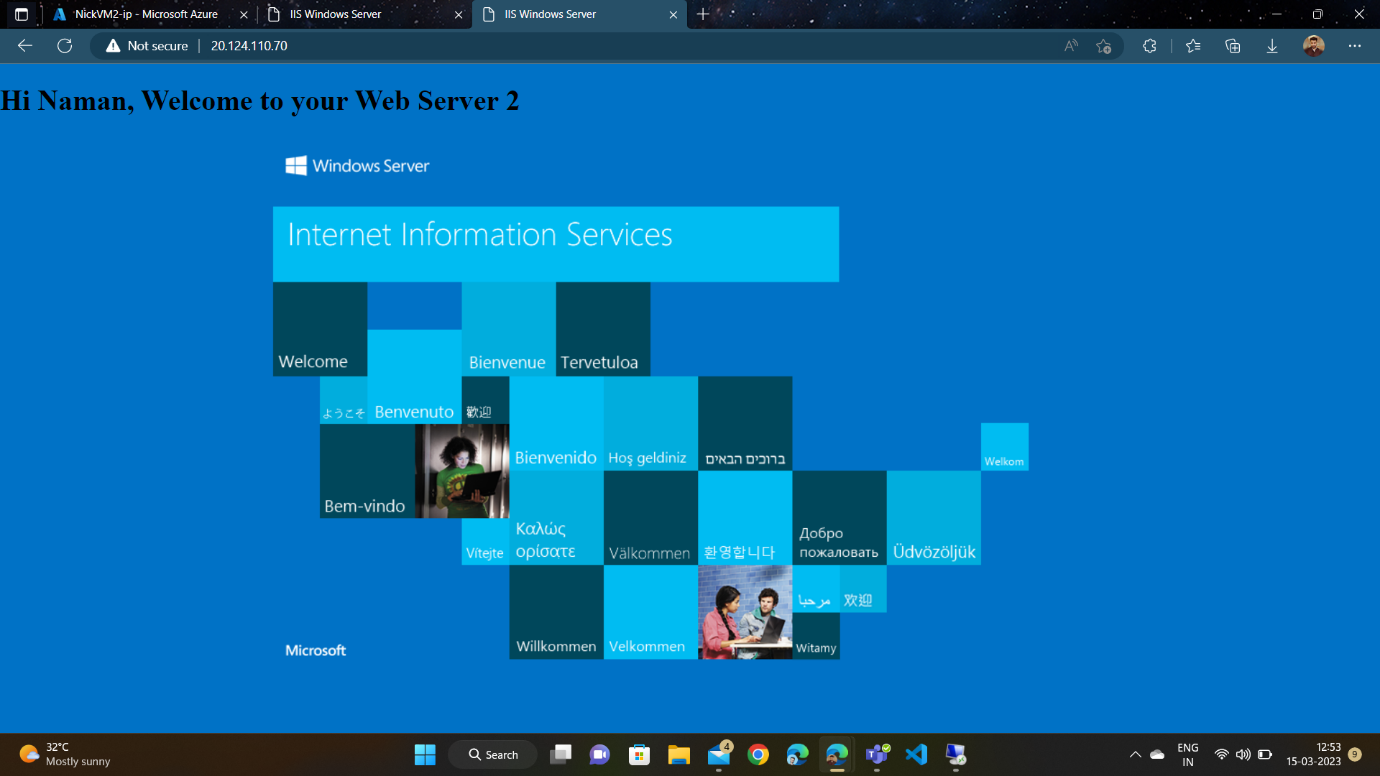
Description automatically generated**

1. **Here’s our VM1.**

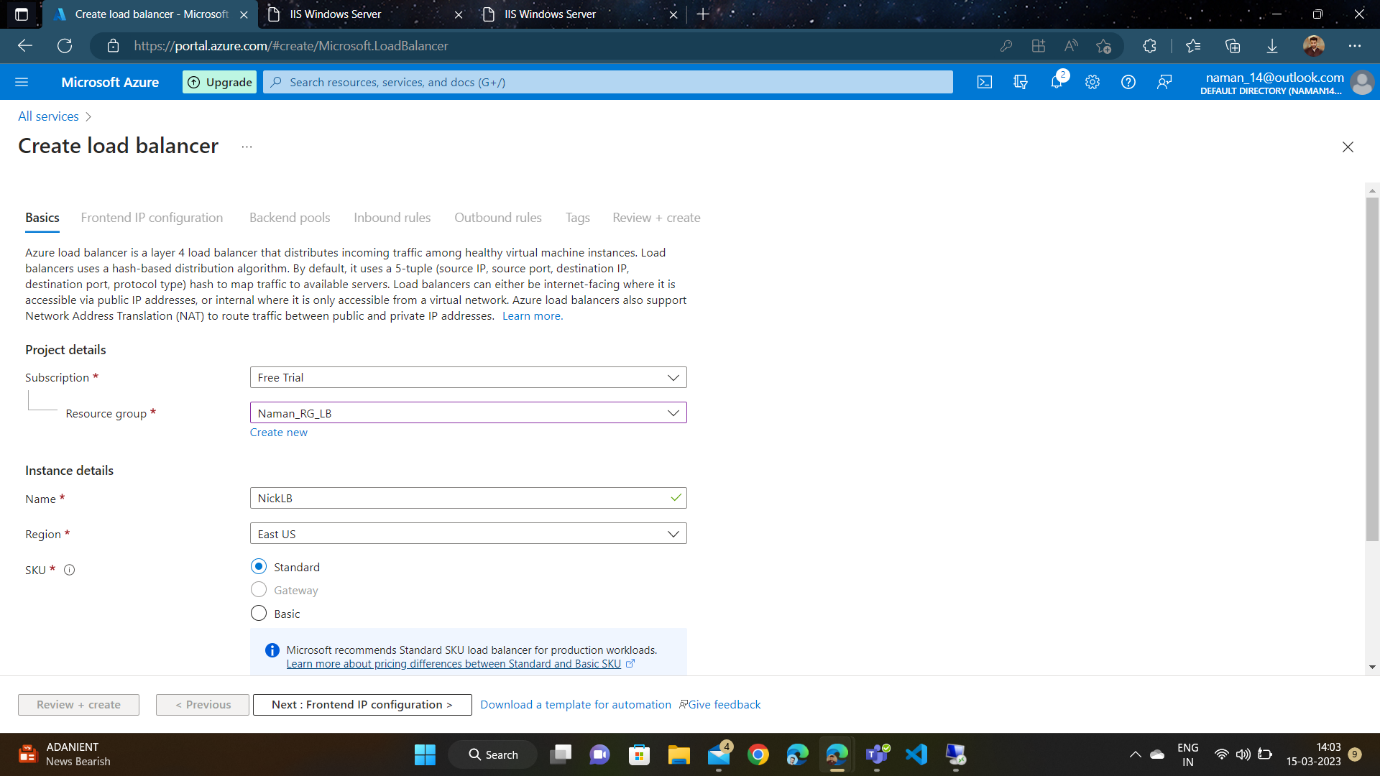
**Graphical user interface, website

Description automatically generated**

1. **Here’s our VM2.**

****

1. **Creating Azure Load Balancer.**

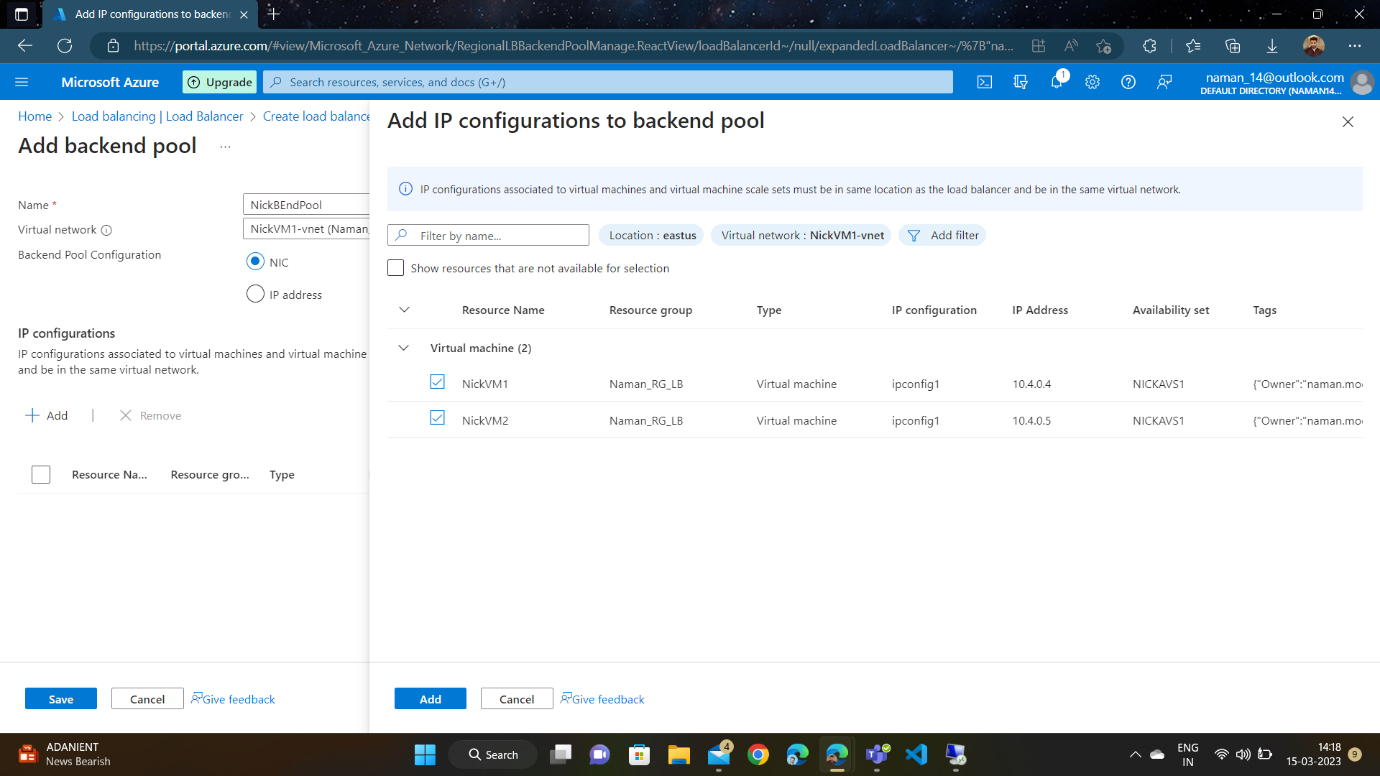
****

1. **Adding Frontend IP Configuration by creating a new public IP.**

**A screenshot of a computer

Description automatically generated**

1. **Adding both VMs to backend pool.**

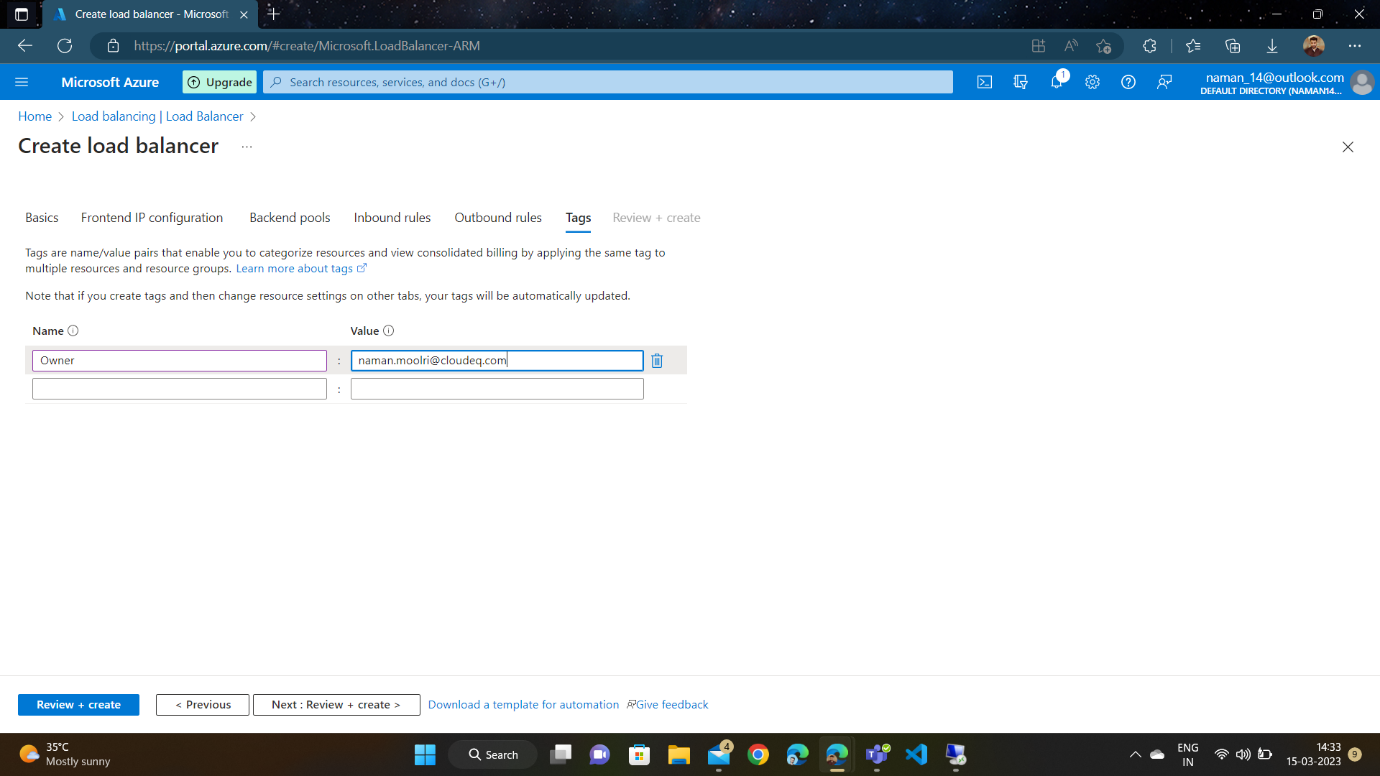
****

1. **Adding Inbound Rules and creating Health Probe while doing it.**

**A screenshot of a computer

Description automatically generated**

1. **Adding Tags.**

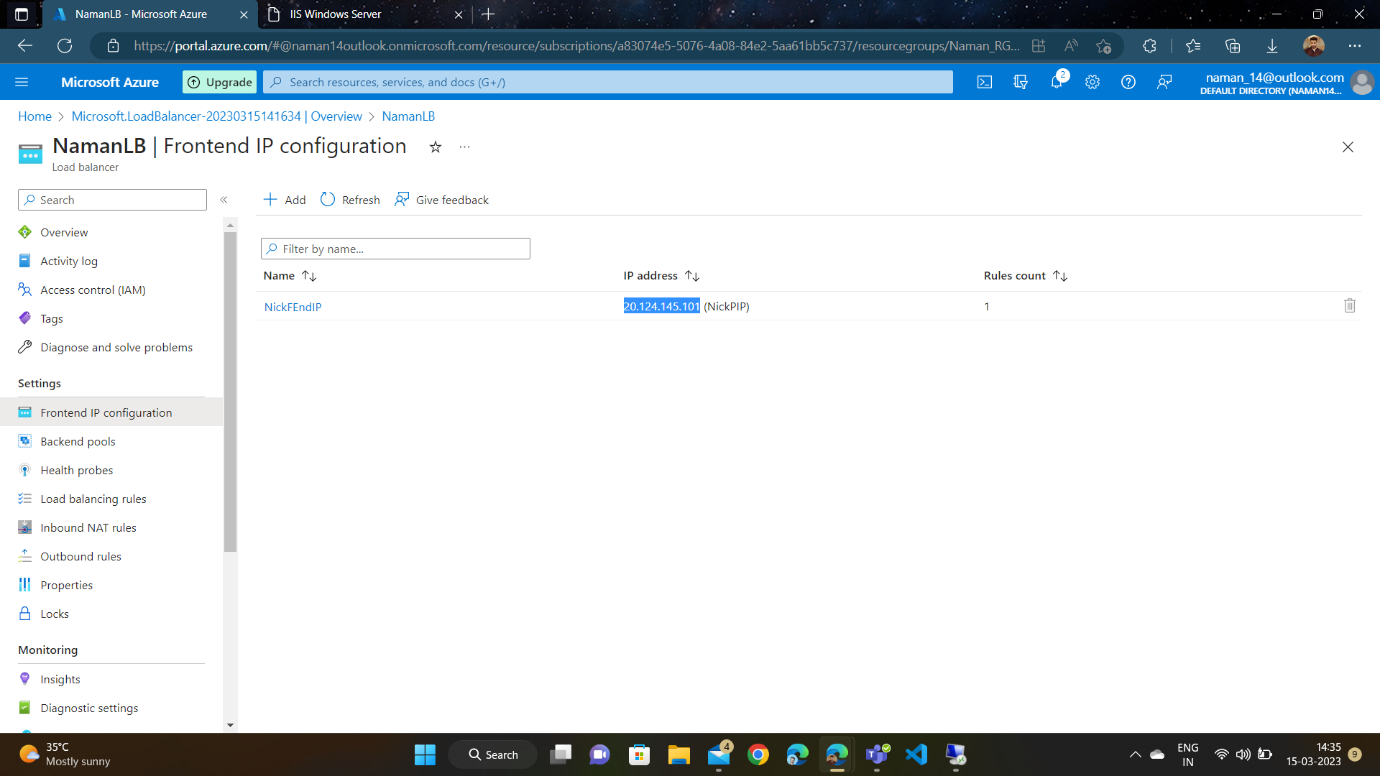
****

1. **Azure Load Balancer is deployed.**

**A screenshot of a computer

Description automatically generated**

1. **Accessing our Azure Load Balancer through Frontend IP.**

****

1. **Our Load Balancer directs us to our VM2.**

**Graphical user interface, website

Description automatically generated**

1. **Waiting for 4 minutes, and it automatically redirects to our VM1.**

**Graphical user interface, website

Description automatically generated**