**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

**Set Up a Load Balancer in the Cloud Configure a load balancer to distribute traffic across multiple VMs hosting your web application**

Name: Kirubalini S I Department: IT



**Introduction**

In today's dynamic and demanding web application landscape, ensuring high availability, scalability, and performance is paramount. A load balancer serves as a crucial component in achieving these goals. It acts as a traffic director, distributing incoming network traffic across multiple virtual machines (VMs) or servers hosting your web application. By evenly distributing the load, a load balancer prevents any single server from becoming overwhelmed, leading to improved responsiveness, reduced latency, and enhanced reliability. Setting up a load balancer in the cloud involves configuring its parameters to match your application's requirements, ensuring optimal traffic distribution and failover capabilities.

**Objective**

**High Availability:** To ensure that the web application remains accessible even if one or more servers fail.

**Scalability:** To distribute traffic effectively as the number of users and requests increases, allowing the application to scale horizontally.

**Performance Optimization:** To minimize latency and improve response times by distributing traffic evenly across available servers.

**Fault Tolerance:** To provide automatic failover capabilities, redirecting traffic away from unhealthy servers.

**Traffic Management:** To provide control over how traffic is routed, including features like session persistence, SSL termination, and content-based routing.

**Importance**

**Enhanced User Experience:** Load balancers ensure a smooth and responsive user experience, even during peak traffic periods.

**Improved Reliability:** They prevent single points of failure, ensuring that the application remains available even if individual servers go down.

**Increased Scalability:** Load balancers enable applications to handle increasing traffic loads without experiencing performance degradation.

**Simplified Maintenance:** They allow for seamless server maintenance and updates without causing downtime.

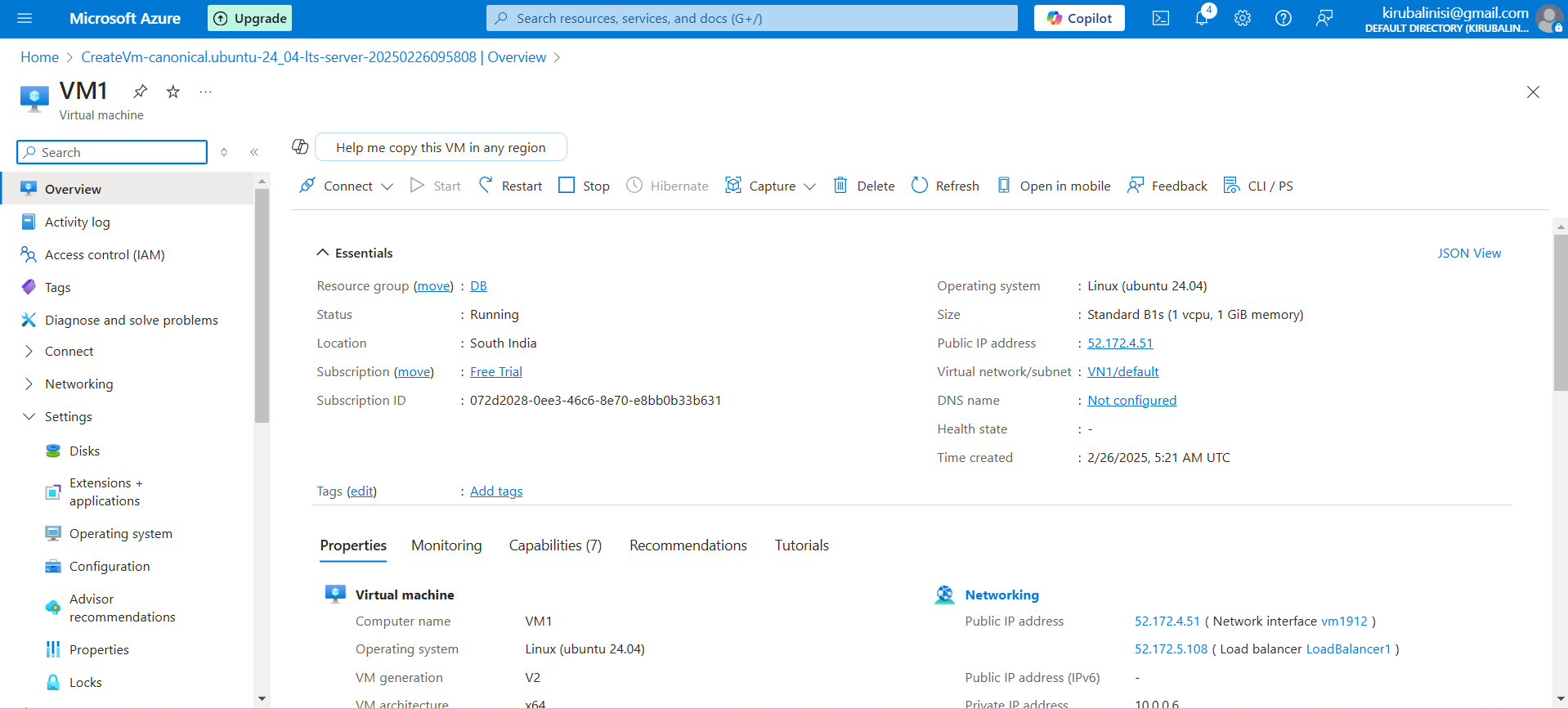
**Security:** Load balancers can enhance security by providing features like SSL termination and DDoS protection.

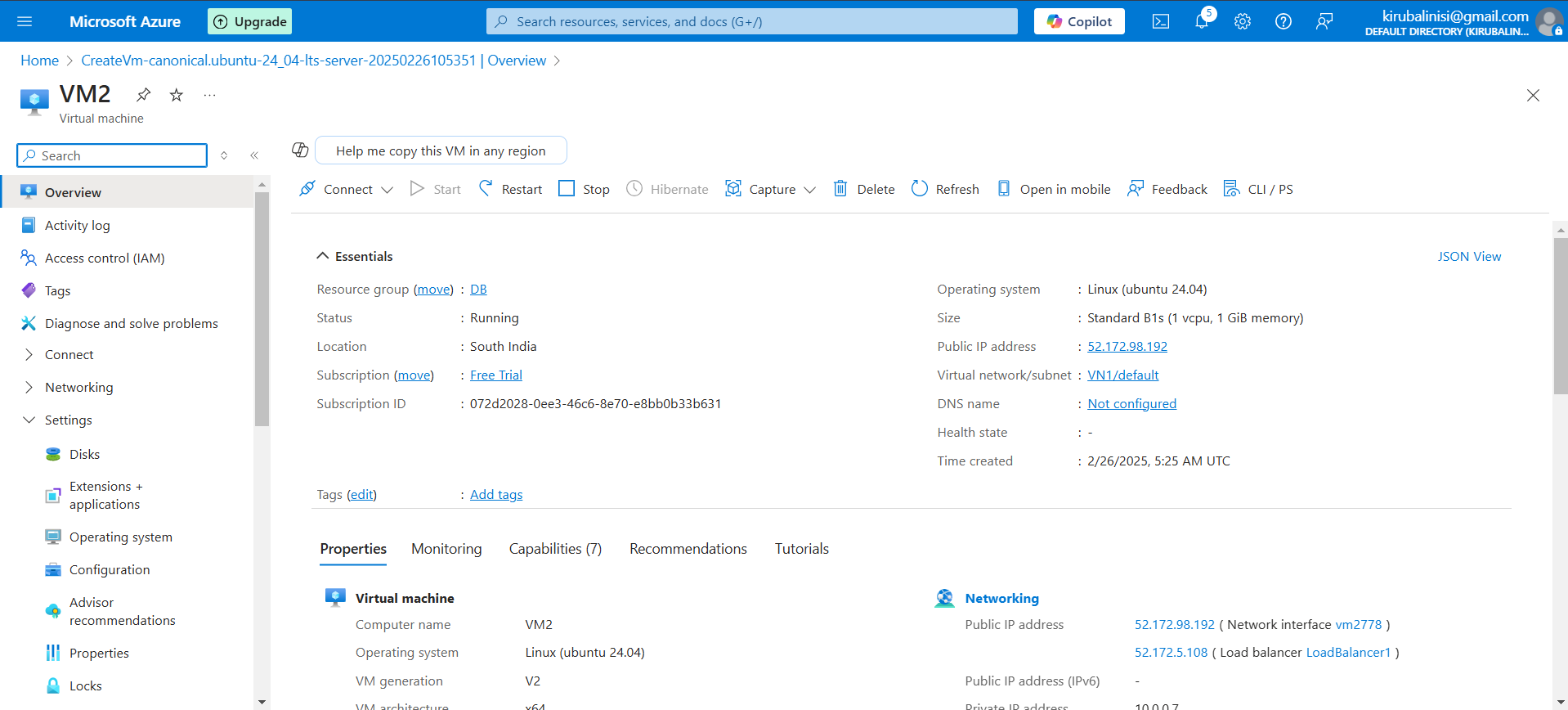
**Cost Efficiency:** By efficient traffic distribution, load balancers allow for better resource utilization, and therefore cost savings.

**Step-by-Step Overview**

Step1:

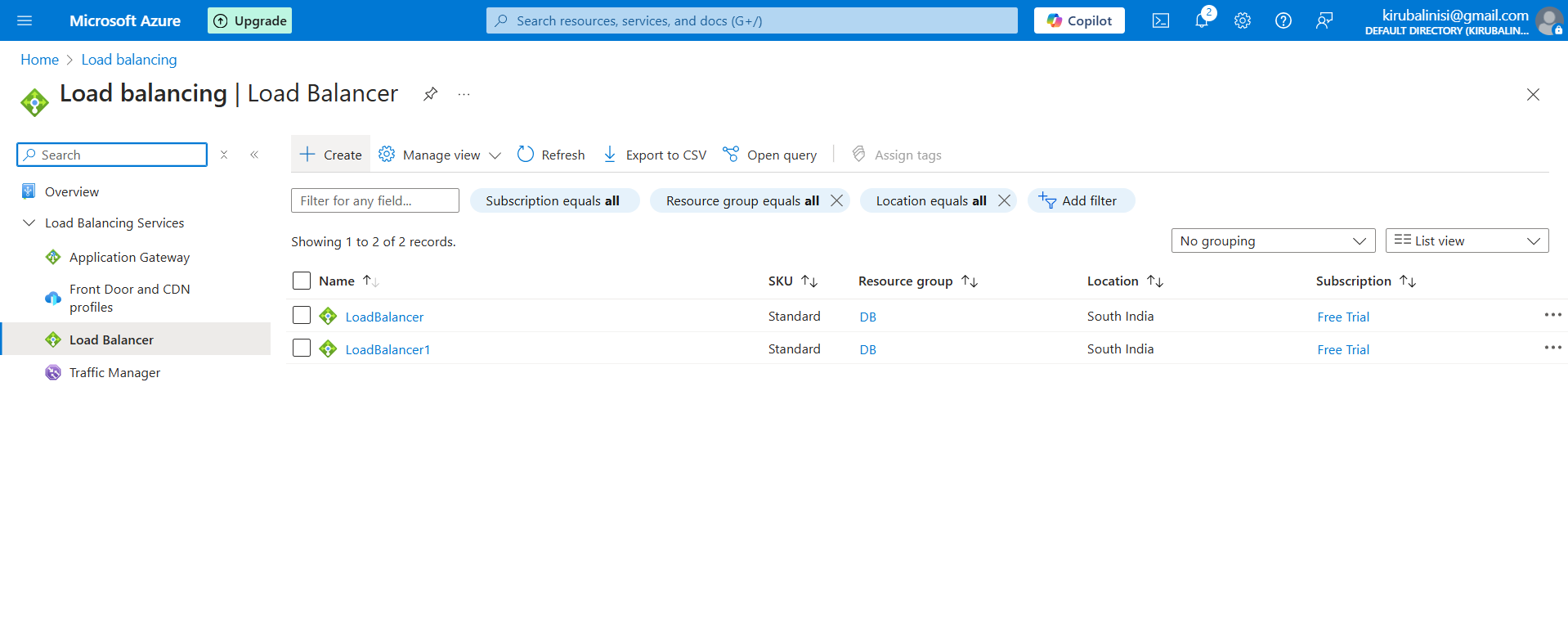
Go to the portal and Create at least two virtual machines. Choose the same region and resource group for all VMs.

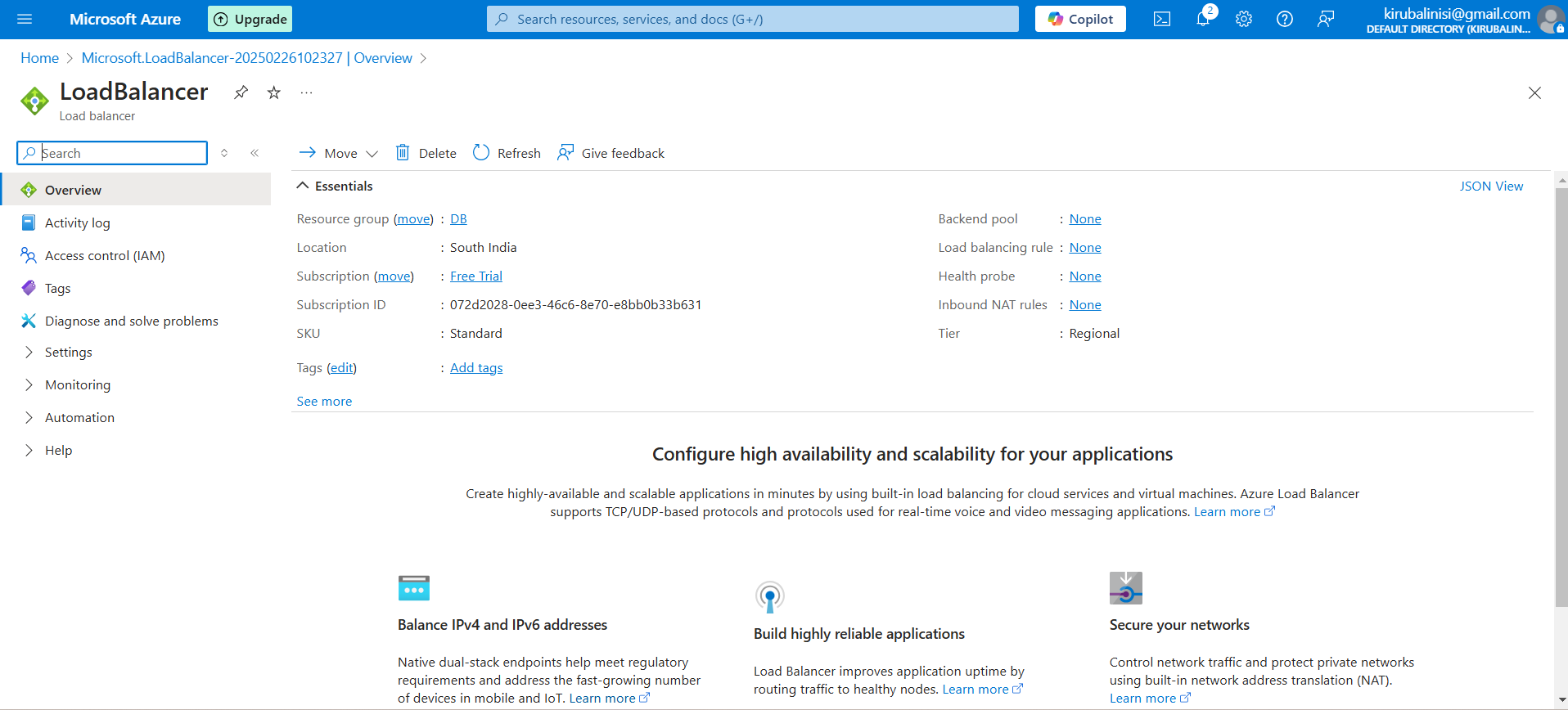




Step 2:

Navigate to Load Balancers and Fill in appropriate details and click “Review&Create”



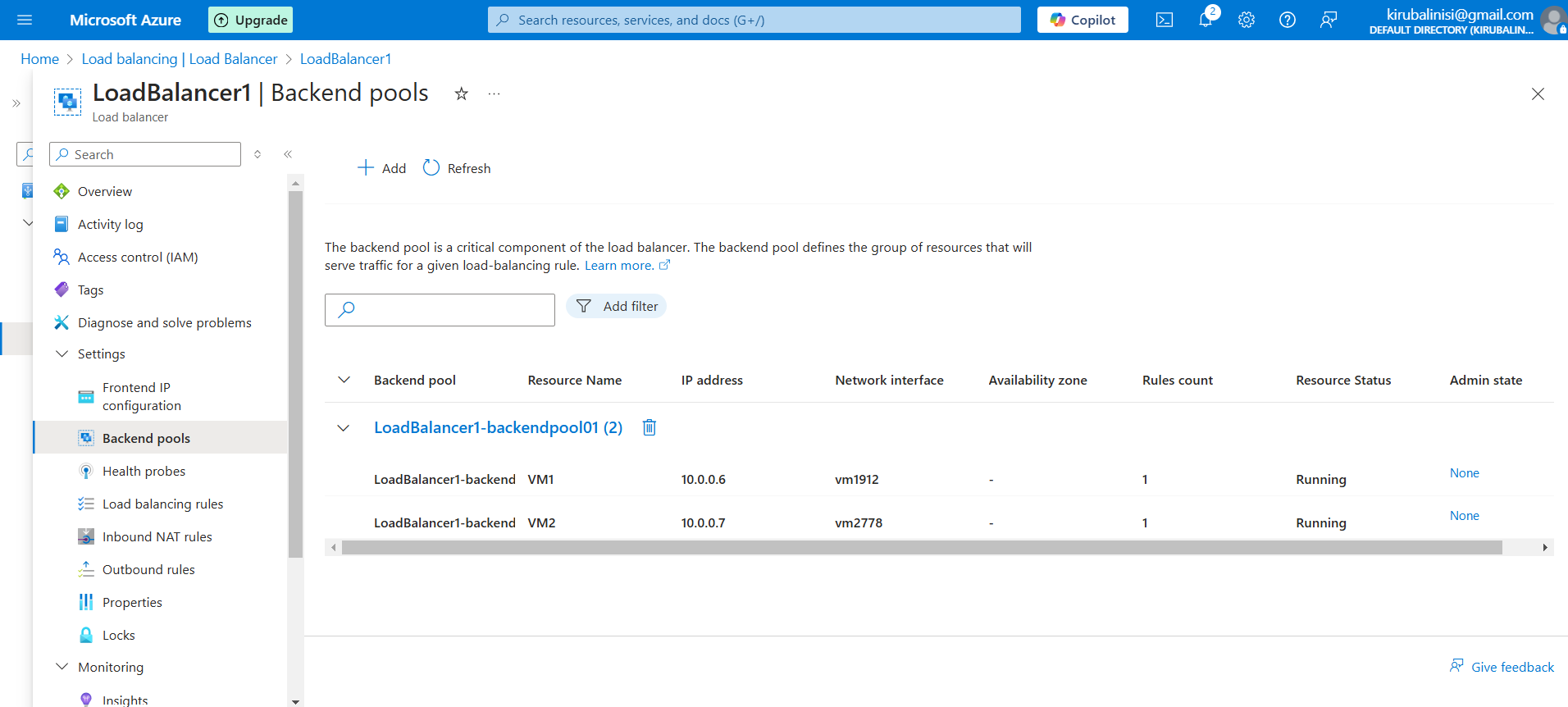


Step 3:

Configure the Backend Pool

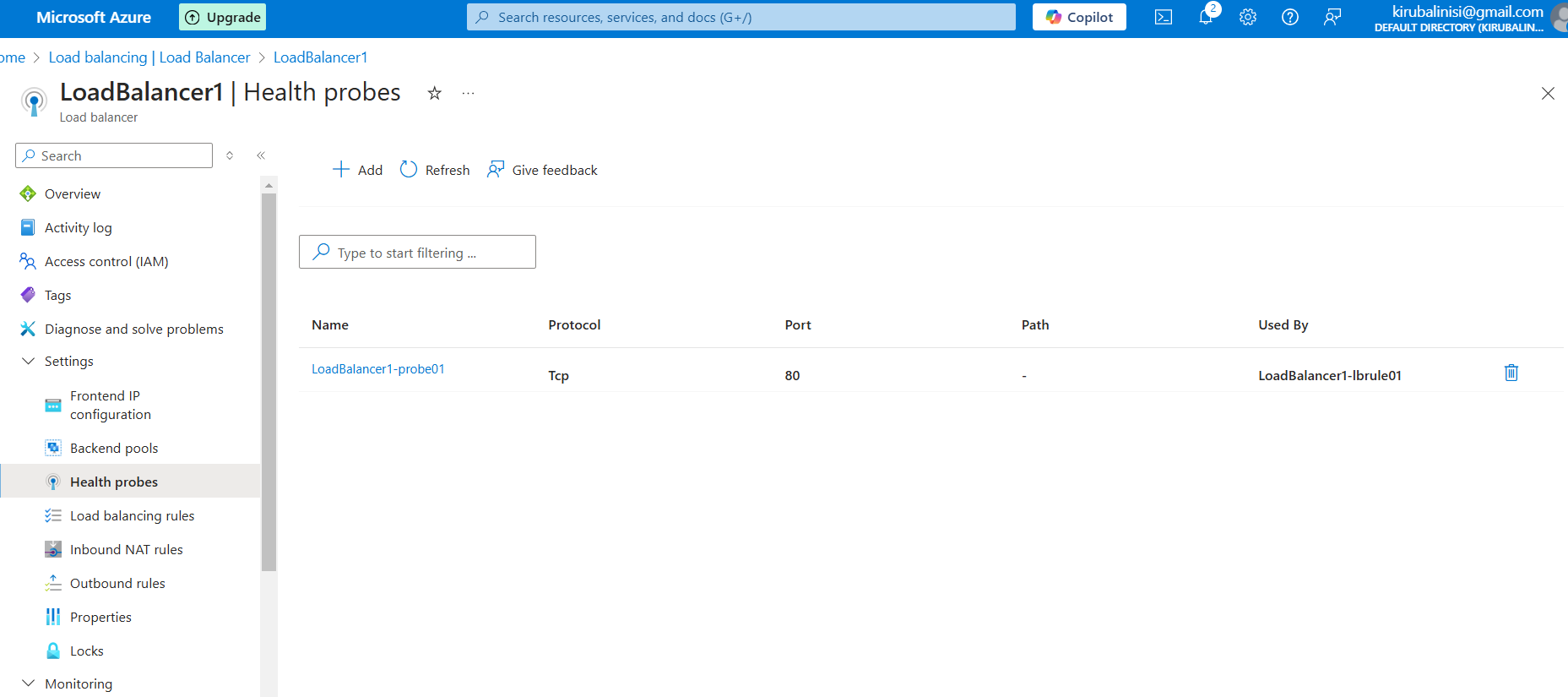
Create Backend Pools for each Virtual Networks

Fill in the required details to configure the pools



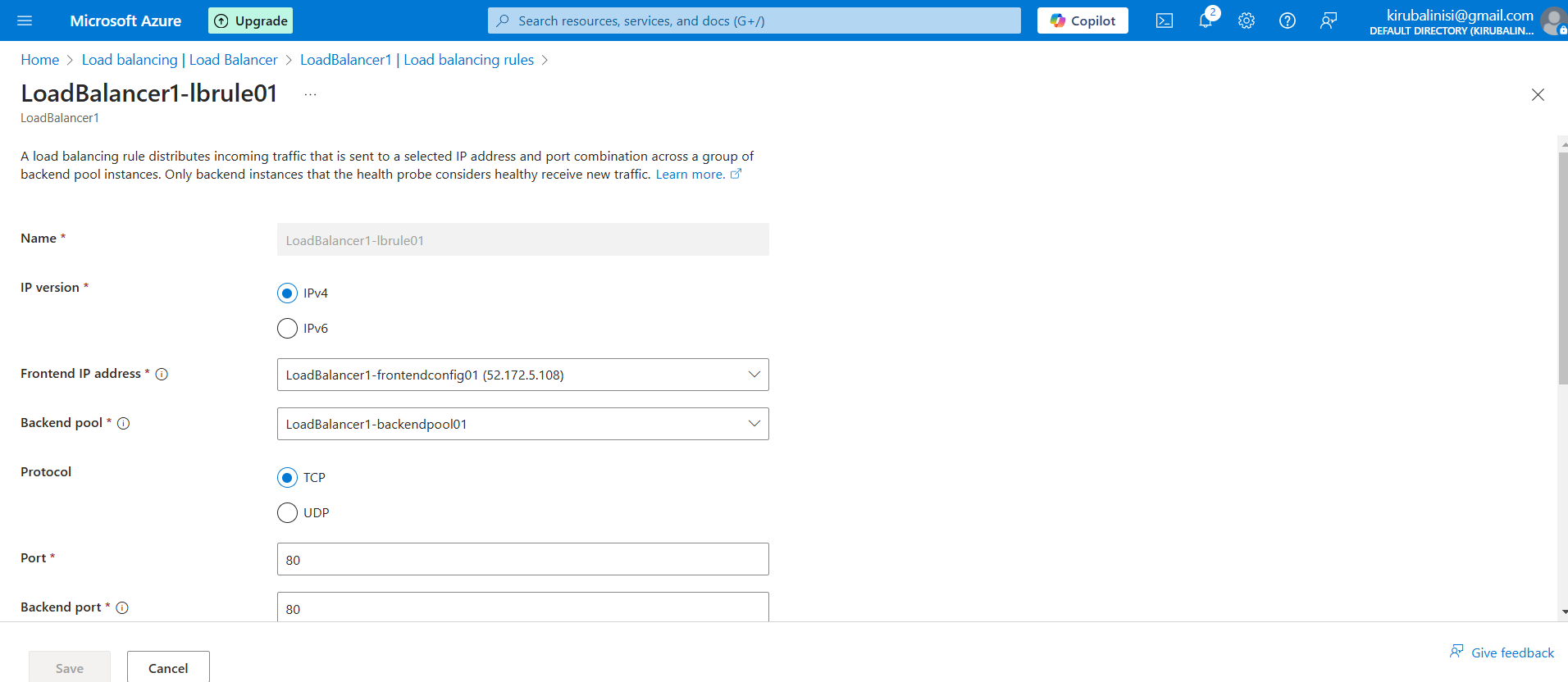
Step 4:

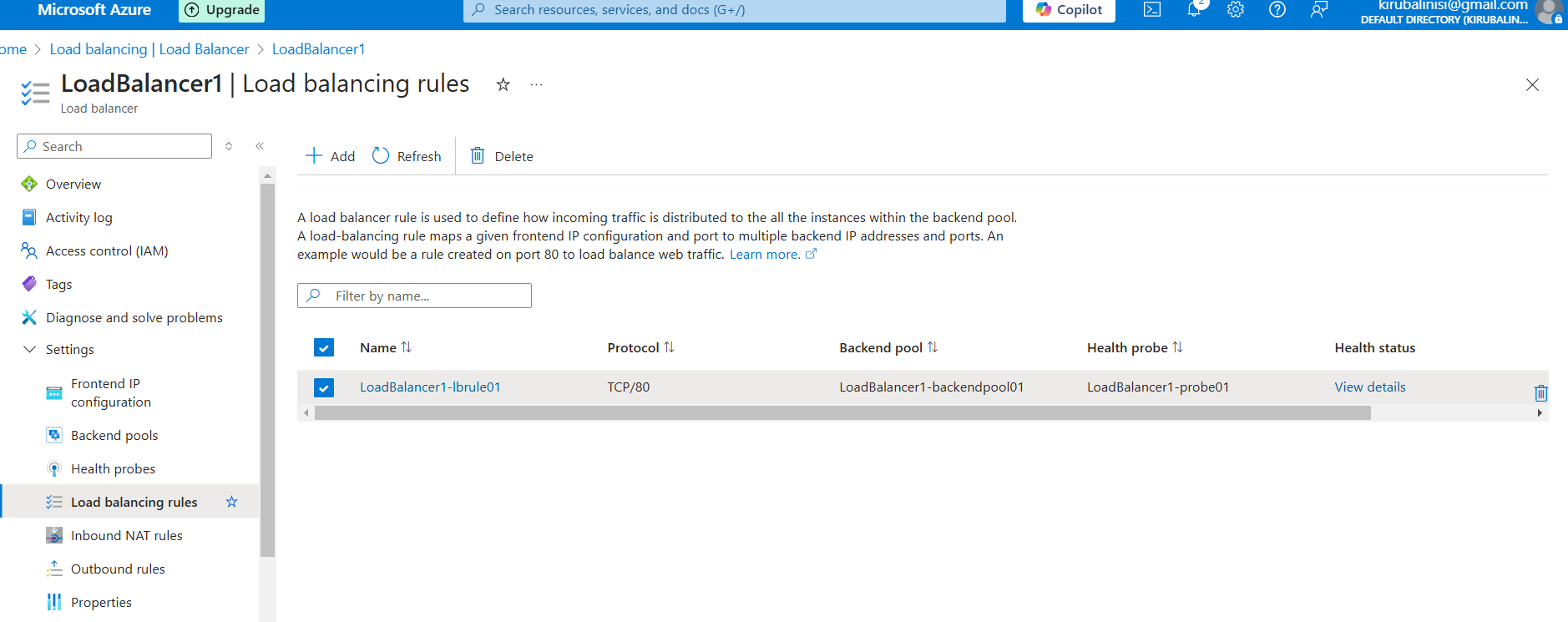
Create Health Probe in the Load Balancing



Step 5:

Create a Load Balancing Rule





Step 6:

Test the Load Balancer

Public Load Balancer:-

Access the public IP address of the load balancer in your web browser.

Internal Load Balancer:

Access the internal IP address of the load balancer from a VM within the same virtual network.

Access the Load Balancer's IP address in your browser

**OUCOMES**

**Zero Downtime Deployments:** Seamless updates and maintenance without interrupting service.

**Improved Application Performance:** Reduced latency and faster response times.

**Increased Application Availability:** Enhanced reliability and fault tolerance.

**Enhanced User Satisfaction:** A smooth and responsive user experience.

Better Resource Utilization: Efficient distribution of traffic across servers.

**Scalable Infrastructure:** Ability to handle growing traffic demands.

**Increased resilience:** The application is more resilient to traffic spikes, and server failures.

**Simplified management:** Centralized traffic management