A
Summer Internship Report
On
"Azure Portal"

(IT446 – Summer Internship - I)

Prepared by
Ridham Chavda (20IT016)

Under the Supervision of
Dr. Nirav Bhatt

Submitted to

Charotar University of Science & Technology (CHARUSAT)
for the Partial Fulfillment of the Requirements for the
Degree of Bachelor of Technology (B.Tech.)
for Semester 7

Submitted at

CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY
Accredited with Grade A+ by NAAC
Accredited with Grade A by KCG

SMT. KUNDANBEN DINSHA PATEL DEPARTMENT OF INFORMATION TECHNOLOGY

Chandubhai S. Patel Institute of Technology (CSPIT)
Faculty of Technology & Engineering (FTE), CHARUSAT
At: Changa, Dist: Anand, Pin: 388421.

August, 2023

CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

Accredited with Grade A by NAAC
Accredited with Grade A by KCG

CERTIFICATE

This is to certify that the report entitled "**Azure Portal**" is a bona fide work carried out by Ridham Chavda (20IT016) under the guidance and supervision of **Dr. Nirav Bhatt** & **Mr. Lalit Sharma** for the subject **Summer Internship – II (IT446)** of 7th Semester of Bachelor of Technology in **Department of Information** at Chandubhai S. Patel Institute of Technology (CSPIT), Faculty of Technology & Engineering (FTE) – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of the candidate herself, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred by the examiner(s).

Under the supervision of,

Dr. Nirav Bhatt
Head of Department (AI-ML),
CSPIT,CHARUSAT, Changa, Gujarat.

Mr. Lalit Sharma
Associate Devops Consultant
Celebal Technologies, Jaipur

Dr. (Prof.) Parth Shah
Head of Department (IT)
CHARUSAT, Changa, Gujarat.

ABSTRACT

Project 1:
The project focuses on implementing an Azure Load Balancer to distribute incoming network traffic efficiently between two Virtual Machines (VMs) hosted in Microsoft Azure cloud. Load balancing plays a critical role in maintaining high availability and ensuring optimal performance of applications and services. The primary objective of this project is to achieve fault tolerance and improved scalability by evenly distributing incoming requests among the VMs.

Project 2:
The project revolves around the implementation and utilization of Azure Application Gateway in a cloud-based infrastructure, along with two Virtual Machines (VMs), to enhance the performance and security of web applications. Azure Application Gateway acts as a powerful Application Delivery Controller (ADC) that provides advanced load balancing, routing, and secure application delivery services.

ACKNOWLEDGEMENT

I would like to extend my heartfelt congratulations and appreciation to [Your Name] for the successful completion of two mini-projects focused on Azure Load Balancer and Application Gateway. Your dedication and efforts in understanding and implementing these crucial components in Microsoft Azure are commendable.

Through your hard work, you have demonstrated a profound understanding of load balancing and application delivery mechanisms in the Azure cloud environment. Your ability to configure, manage, and optimize these services showcases your expertise in cloud architecture and infrastructure.

I commend your commitment to learning and problem-solving, as evidenced by your thorough exploration of Azure's load balancing capabilities and application delivery options. Your projects have undoubtedly contributed to a better understanding of these technologies among your peers and the wider community.

As we continue to embrace cloud technologies, your contributions in mastering Azure's load balancing and application delivery mechanisms will prove invaluable in enhancing the performance, reliability, and security of cloud-based applications.

Once again, congratulations on your achievements, and I look forward to witnessing your continued growth and success in the world of cloud computing.

Yours thankfully,
Ridham Chavda

Table of Contents

CHAPTER 1: INTRODUCTION

1.1 PROJECT DEFINITION:

I made Two mini projects in this internship, that are Implementing Azure Load Balancer for High Availability and Scalability and Securing and Optimizing Web Traffic using Azure Application Gateway.

1.2 DESCRIPTION:

Project 1: The project focuses on implementing an Azure Load Balancer to distribute incoming network traffic efficiently between two Virtual Machines (VMs) hosted in Microsoft Azure cloud. Load balancing plays a critical role in maintaining high availability and ensuring optimal performance of applications and services. The primary objective of this project is to achieve fault tolerance and improved scalability by evenly distributing incoming requests among the VMs.

Project 2: The project revolves around the implementation and utilization of Azure Application Gateway in a cloud-based infrastructure, along with two Virtual Machines (VMs), to enhance the performance and security of web applications. Azure Application Gateway acts as a powerful Application Delivery Controller (ADC) that provides advanced load balancing, routing, and secure application delivery services.

1.3 PROJECT REQUIREMENTS:

1.3.1 Software Requirements:

Azure Portal

1.3.2 Hardware Requirements:

All new generation processors and RAM should be at least 2 GB or above.

CHAPTER 2: SYSTEM FUNCTIONALITY

2.1 MAJOR FUNCTIONALITY:

Project 1: The primary functionality of your project involving Azure Load Balancer and two Windows Virtual Machines is to achieve high availability, fault tolerance, and efficient distribution of incoming network traffic to the web application hosted on the VMs.

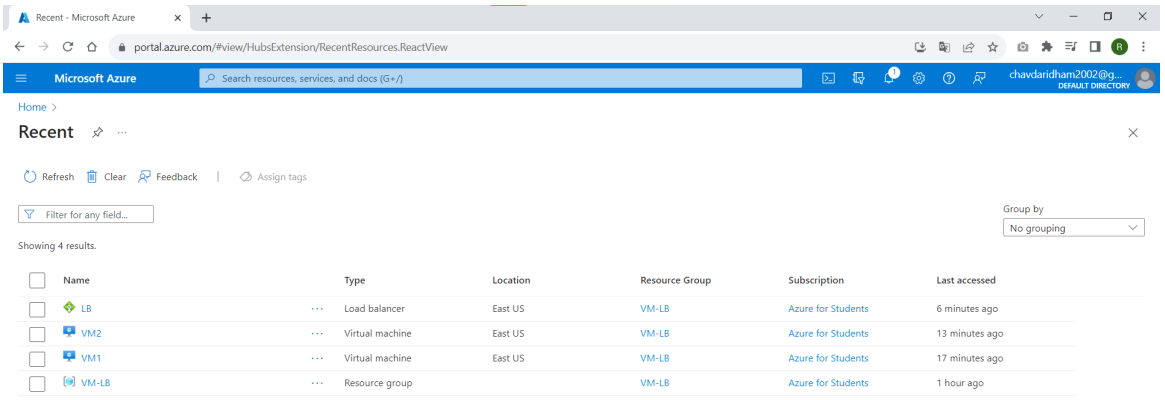
Project 1: The primary functionality of mini project 2 involving Azure Application Gateway and two Windows Virtual Machines is to enhance the security, scalability, and performance of your web application by utilizing Azure Application Gateway as a powerful application delivery controller.

CHAPTER 3: IMPLEMENTATION

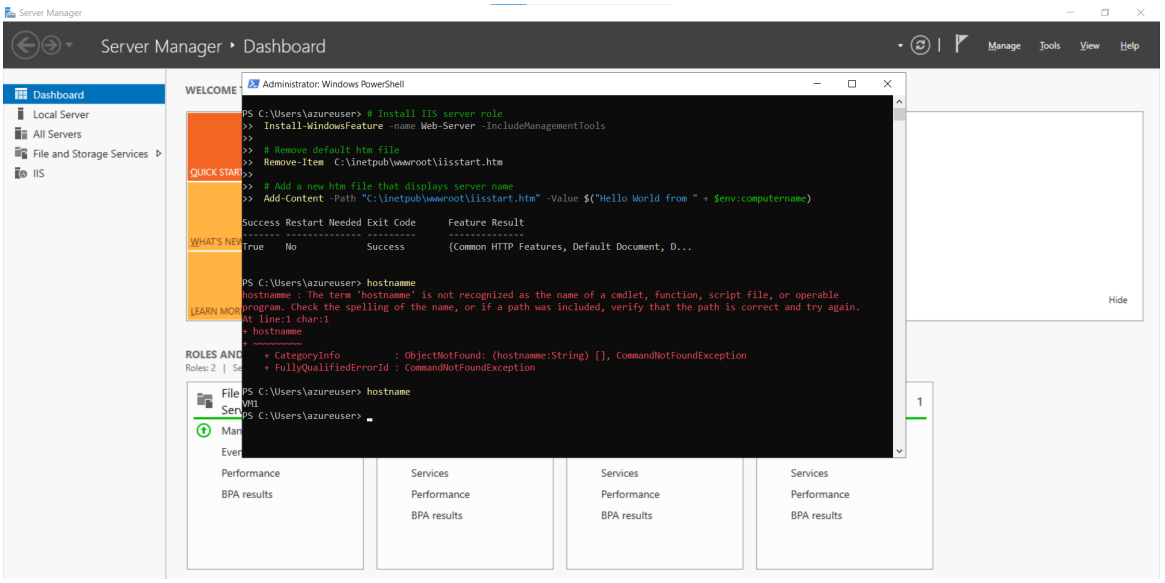
TASK 1: Load Balancer with two VMs

The Azure Load Balancer is used to distribute incoming traffic across multiple virtual machines (VMs) in Azure. It ensures high availability, fault tolerance, and scalability for your applications. By creating load balancing rules and configuring health probes, it efficiently routes traffic to healthy VMs, providing a seamless user experience and optimal performance.

The Azure Load Balancer is a crucial component for achieving high availability and fault tolerance in Azure. It evenly distributes incoming network traffic, such as TCP and UDP flows, across multiple backend virtual machines within a virtual network.

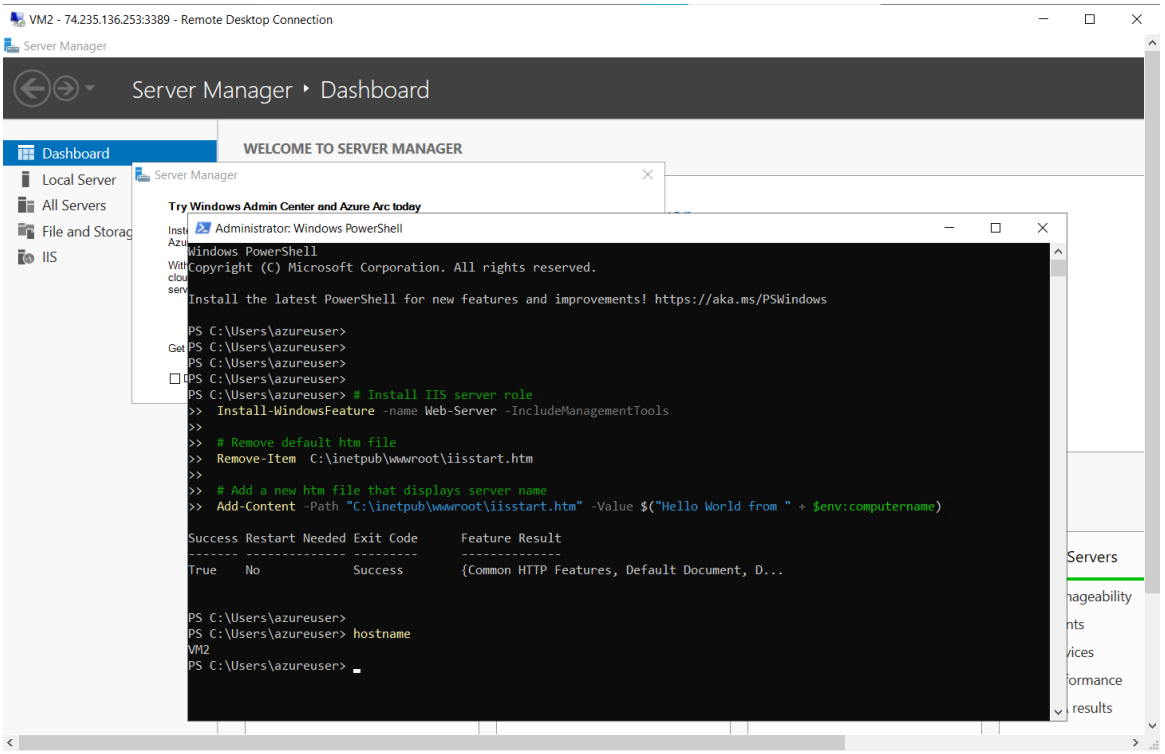


1.1 Resources Which are created

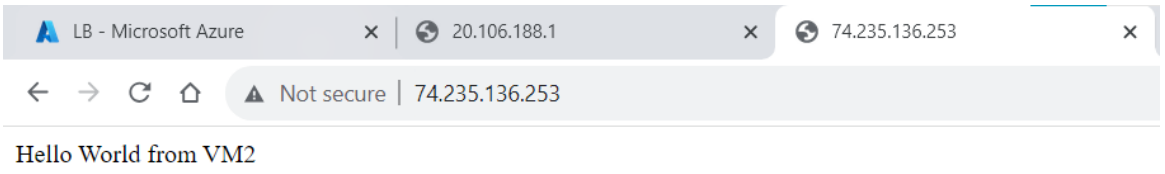


1.2 Install IIS server on VM1

1.3 IP address of the first VM



1.4 Install the IIS server on VM2

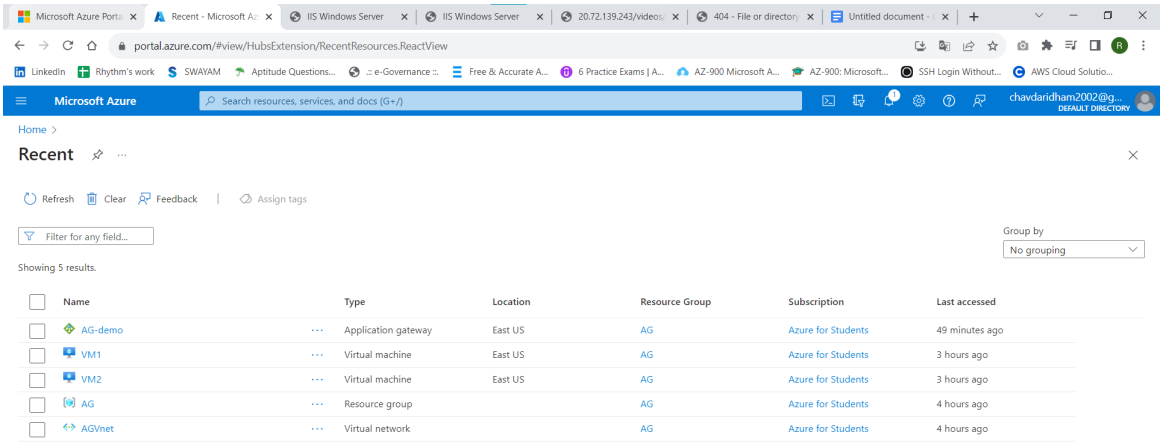


1.5 IP address of the second VM

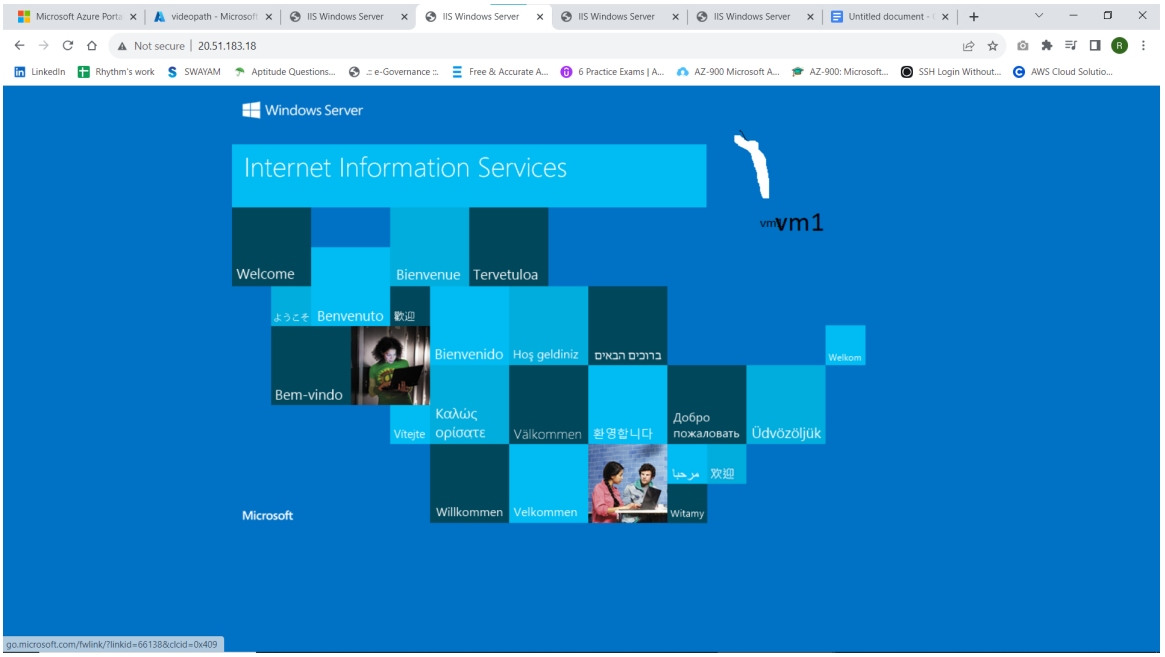
TASK 2: Application Gateway with two VMs

Application Gateway is a web traffic load balancer and application delivery controller service in Microsoft Azure. It distributes incoming HTTP/HTTPS requests to multiple backend servers, improving application responsiveness and reliability. Key features include URL-based routing, SSL termination, a Web Application Firewall (WAF) for security, session affinity, autoscaling, and health probes. It enhances performance, security, and availability for web applications in Azure.

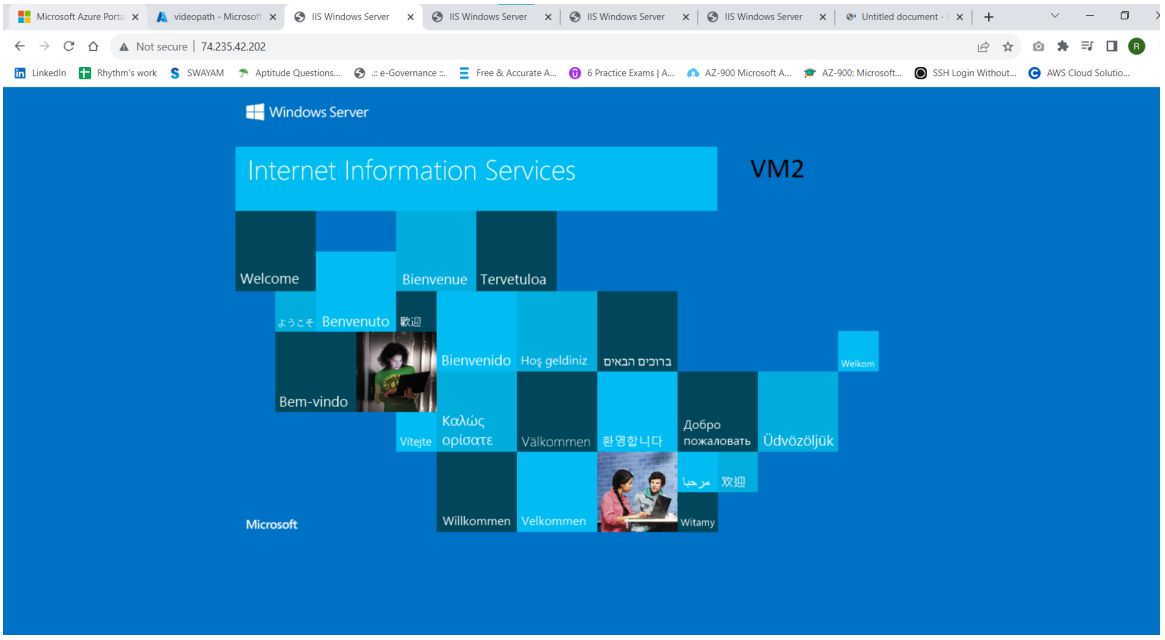
- Load Balancing
- High Availability
- SSL Termination
- Web Application Firewall (WAF)
- URL-Based Routing
- Session Persistence:
- Autoscaling
- Health Probes
- Header Manipulation
- Microservices Support



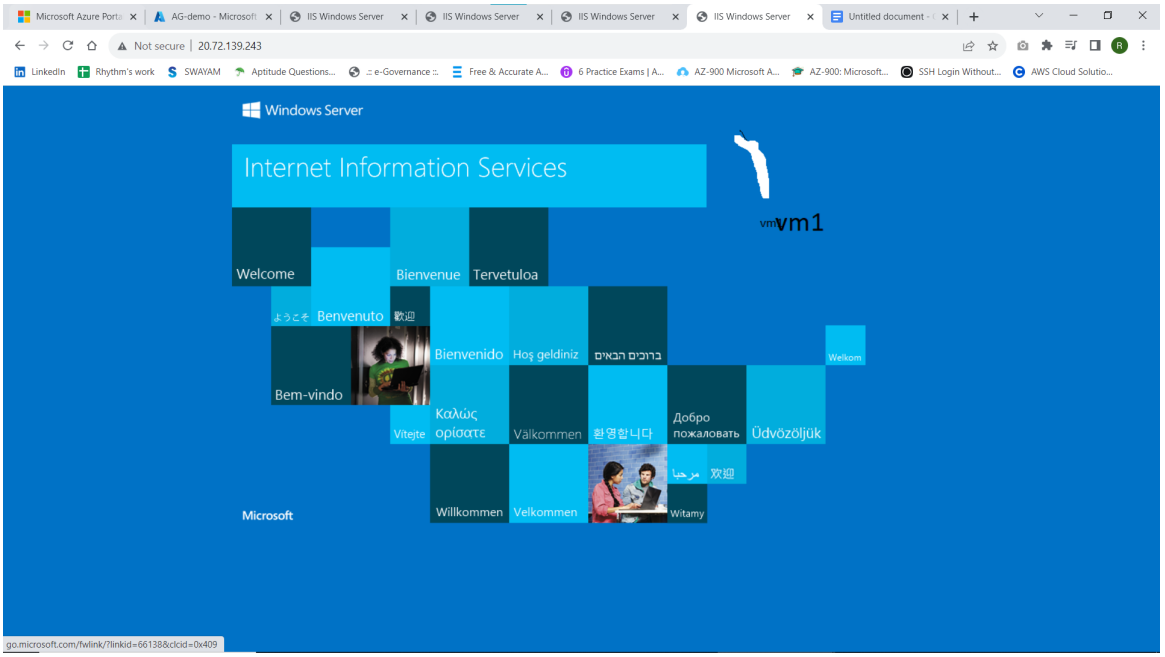
2.1 Resources Which are created



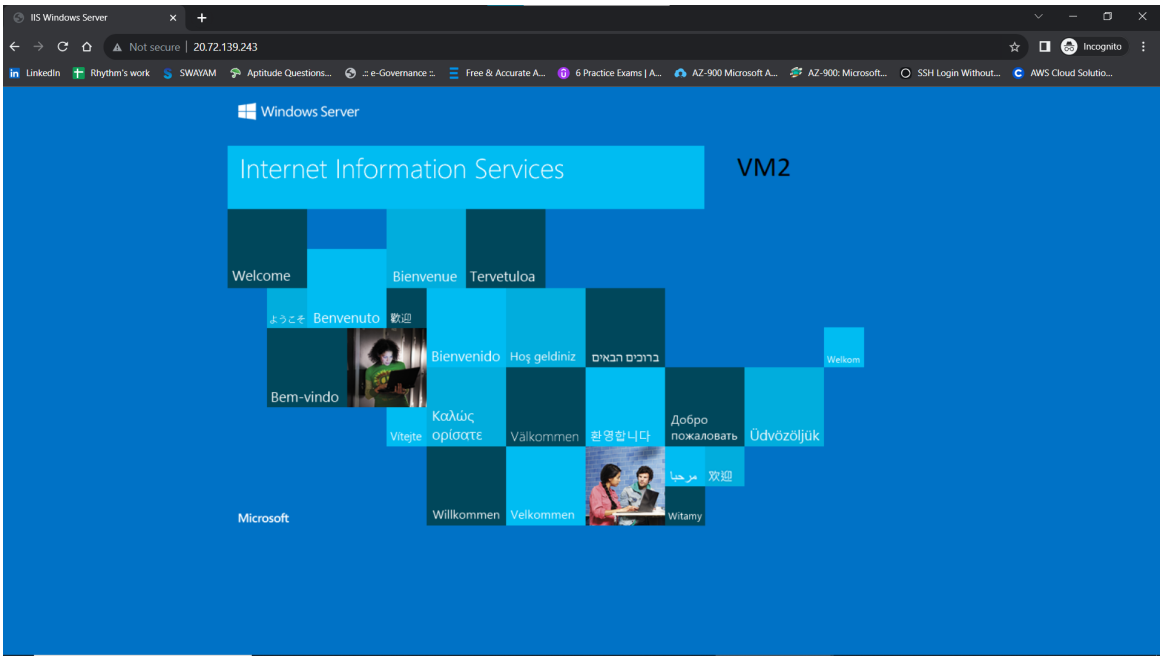
2.2 IP address of the first VM



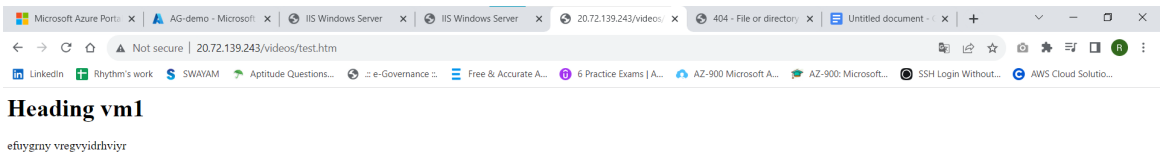
2.3 IP address of the second VM



2.4 Application Gateway IP address with first VM output



2.5 Application Gateway IP address with second VM output



2.6 Application Gateway IP address with HTML file output from the first VM

CHAPTER 4: LIMITATIONS AND OUTCOMES

4.1 LIMITATIONS:

1. **Minimum VM Instances**: Azure Load Balancer requires at least two VM instances to achieve fault tolerance and load balancing. This means you must have two or more VMs in your backend pool. Operating with just one VM in the backend pool may not provide the desired load balancing and high availability benefits.

2. **Single Datacenter**: Azure Load Balancer is limited to distributing traffic within a single Azure datacenter or availability zone. If you require load balancing across multiple data centers or regions for disaster recovery or global distribution, you might need to explore other load balancing solutions like Azure Traffic Manager.

3. **Health Probe Sensitivity**: The effectiveness of load balancing relies on health probes to determine the health of VM instances. Configuring the health probe's sensitivity correctly is crucial. If the probe sensitivity is too low, it may lead to the inadvertent removal of healthy VMs from the load balancer's rotation, causing service disruptions.

4.2 OUTCOMES:

1. **High Availability**: By configuring the two Windows Virtual Machines in an Availability Set or Availability Zone and setting up the Azure Load Balancer, your web application achieves high availability. If one VM becomes unavailable due to maintenance or failure, the load balancer automatically redirects traffic to the healthy VM, ensuring continuous service availability.

2. **Scalability**: The load balancer evenly distributes incoming traffic across the two Windows VMs, allowing your web application to handle increased load and traffic spikes more efficiently. This results in better scalability, ensuring a smoother user experience during periods of high demand.

3. **Load Distribution**: The load balancing algorithm ensures that traffic is distributed evenly among the VM instances in the backend pool. This prevents any single VM from being overloaded and helps maximize resource utilization.

4. **Enhanced Security**: By leveraging the Web Application Firewall (WAF) feature of Azure Application Gateway, your web application gains protection against common web vulnerabilities and attacks, such as SQL injection, cross-site scripting (XSS), and more. This improves the overall security posture of your application.

CHAPTER 5: FUTURE ENHANCEMENTS

5.1 FUTURE ENHANCEMENTS IN PROJECT:

1. **Integration with Azure Virtual Machine Scale Sets (VMSS):** Consider integrating Azure Load Balancer with VMSS instead of using standalone VMs. VMSS allows for automatic scaling of VM instances based on demand, enabling seamless horizontal scaling without manual intervention.

2. **Multi-Region Deployment**: Extend your load balancing solution across multiple Azure regions for improved disaster recovery and high availability. This way, you can ensure your application remains accessible even if an entire region becomes unavailable.

3. **Content-Based Routing**: Explore content-based routing options where traffic is directed to specific backend VMs based on the content or data being requested. This can be particularly useful for serving different types of content or routing traffic to specialized backend instances.

CHAPTER 6: REFERENCES AND CONCLUSION

6.1 WEB REFERENCES:

- https://learn.microsoft.com/en-us/azure/load-balancer/
- https://learn.microsoft.com/en-us/azure/application-gateway/

6.2 CONCLUSION:

Azure Load Balancer Mini Project:

- **Achieved High Availability**: By configuring the Windows Virtual Machines in an Availability Set or Availability Zone and setting up the Azure Load Balancer, the web application achieved high availability. The load balancer automatically routed traffic to healthy VMs, ensuring continuous service availability in case of VM failures.

- **Improved Scalability**: The load balancer evenly distributed incoming traffic across the Windows VMs in the backend pool, allowing the web application to handle increased load and traffic spikes efficiently. This resulted in better scalability and a smoother user experience during peak periods.

- **Enhanced Performance**: With the load balancer directing traffic to available and healthy VMs, the web application's performance was optimized, leading to faster response times and reduced latency for users.

Azure Application Gateway Mini Project:

- **Enhanced Security**: By leveraging the Web Application Firewall (WAF) feature of Azure Application Gateway, the web application was protected against common web vulnerabilities and attacks. This improved the overall security posture of the application.

- **SSL Termination and Performance Optimization**: The Application Gateway handled SSL termination, relieving the backend Windows Virtual Machines from the resource-intensive task of SSL encryption/decryption. This optimization improved the application's performance and responsiveness.

- **Load Balancing and URL-Based Routing**: The Azure Application Gateway's load-balancing capabilities distributed incoming traffic evenly across the Windows VMs, ensuring efficient resource utilization and better response times.