UNIVERSITY OF
SCIENCE AND TECHNOLOGY
OF HANOI

GROUP 05 REPORT
WEB SECURITY
2023-2024

Network Load Balancing

Lecturer : Assoc. Prof. Dr. Hoang Xuan Dau
Class : Cyber Security
Group members

BA11-004	**:**	**Lại Hải Anh**
BA11-062	**:**	**Nguyễn Ngọc Linh**
BI12-249	**:**	**Nguyễn Ngọc Nhật Linh**
BI12-201	**:**	**Trương Gia Huy**
BI12-485	**:**	**Nguyễn Đức Phúc Tường**

Hanoi, January 12th 2024

Table of Contents

A. Introduction

In today's increasingly digital world, where businesses and organizations rely heavily on online services, ensuring high availability, scalability, and optimal performance of computing systems is paramount. One crucial aspect of achieving these objectives is load balancing. In order to avoid any one component from getting overloaded and to ensure effective use of the resources that are available, load balancing refers to the distribution of incoming network traffic across different servers, systems, or resources. In today's computer settings, when workloads and demands on applications and services are always growing, it is essential. Distributing traffic and compute-intensive jobs among several resources, like servers, virtual machines, or containers, in an equitable manner is the main goal of load balancing. By distributing the load evenly over all resources, this reduces the possibility of bottlenecks, system breakdowns, or performance degradation. Load balancing contributes to the stability, scalability, and responsiveness of a system by strategically allocating workloads. Load balancing algorithms come in various types, including round-robin, least connection, weighted round-robin, least response time, weighted response time, and so on. These algorithms use different criteria, such as the number of active connections, server response times, or server capacities, to determine how to distribute incoming requests effectively. Selecting the appropriate load balancing algorithm depends on the specific requirements and characteristics of the system or application being load balanced.

This report aims to provide an overview of load balancing, the study of load balancing algorithms, the investigation of web server load balancing clusters by setting up and experimenting a web load balancing cluster of 3 Windows servers based on Network Load Balancing Protocol of MS Windows servers. In the report, we focus on the architecture, features and the pros and cons of load balancing. Regarding practical, we start with the steps of installation process, the configuration and then, the demonstration of the web load balancing cluster.

B. Load balancing algorithms and web server load balancing

1. Architecture and Features

1.1 Load balancing architecture

Load balancing is indeed the process of evenly distributing workload or network traffic across multiple servers or resources. It involves load distribution, routing adjustments, and directing client requests to the appropriate servers or resources. The load balancer plays a crucial role in this process. Regarding the load balancing architecture, a load balancer acts as a “traffic controller” positioned in front of the servers, directing client requests to multiple servers capable of handling them. A load balancing algorithm refers to a set of rules that the load balancer uses to determine the most suitable server for each client request. Load balancing algorithms are implemented to ensure that the traffic is distributed fairly and efficiently, optimizing resource utilization, enhancing system performance, and preventing any server from being overwhelmed. If a server becomes unavailable, the load balancer redirects traffic to the remaining operational servers. Additionally, when a new server joins the server group, the load balancer automatically begins routing requests to it. The diagram below illustrates the architecture of load balancing.

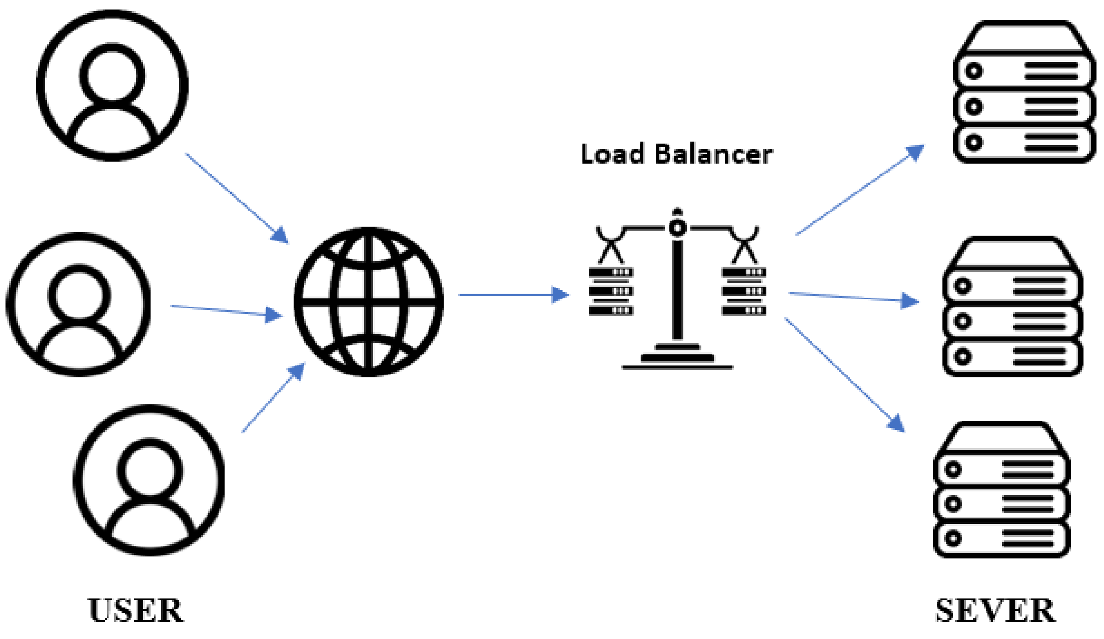


Figure 1: Load balancing architecture

1.2 Load balancing features

1.2.1 Classification based on architecture

Load balancing technology (load balancer) can be broadly categorized into two types:

- Hardware load balancers: A hardware-based load balancer is a dedicated physical device that securely processes and directs large volumes of network traffic to multiple servers. It is commonly deployed in data centers and supports the creation of virtual load balancers through virtualization, enabling centralized management.
- Software load balancers: Software-based load balancers are applications that handle all the tasks related to load balancing. They can be installed on any server of your choice or accessed as a fully managed service provided by a third-party.

1.2.2 Classification based on load balancing methods

Regardless of whether software or hardware is used, the load balancing algorithms is divided into two main types:

- Static load balancing: evenly distributes traffic across servers without making real-time adjustments. Certain static algorithms, following a specified order, distribute an equal amount of traffic to each server in a group. These algorithms assign a fixed weight or priority to each server in the server pool, and the load balancer utilizes this predetermined configuration to determine which server should handle each client request.
- Dynamic load balancing: employs algorithms that consider the real-time status of each server and distribute network traffic accordingly. These algorithms make real-time decisions based on the current state and load of the servers. The load balancer continuously monitors factors such as server health, response times, and current capacity. It then dynamically adjusts the distribution of incoming requests based on this information.

1.2.3 Classification based on the function and purpose of each type

Load balancing can be divided into three primary categories based on the criteria that the load balancer examines in the client request to determine how to distribute the traffic:

- Network load balancing: This type is based on considering network information such as source IP address, destination IP address, source port, and destination port in the client's request. Based on this information, the load balancer determines the appropriate server to handle the request and distributes traffic to that server.
- Global server load balancing: This type involves distributing traffic based on the geographical location of the clients or their network proximity to different servers. The load balancer uses the client's location and the global servers information to route traffic to the nearest geographically or best network performance server.
- DNS load balancing: This type utilizes load balancing techniques based on DNS. When a client sends a request to a domain, the DNS load balancer returns an IP address of a selected server based on load balancing

algorithms. When the client sends subsequent requests, the load balancer may return a different IP address of a different server to distribute traffic.

1.3 Types of load balancing algorithms

1.3.1 Static load balancing algorithms

Round-robin load balancing is a method of distributing traffic to a group of servers in a rotational manner using the Domain Name System (DNS). The round-robin load balancer sequentially forwards client requests to each server in the list. It starts from the first server in the list, then moves to the next server for the next request, and so on. Once it reaches the end of the list, the load balancer loops back to the beginning and continues the rotation. This means that each server in the group will receive an equal share of the traffic over time, assuming all servers are available and responsive. This type of load balancing algorithm is well-suited for situations where all servers possess similar capabilities, and the objective is to evenly distribute the workload.

Weighted Round-robin (WRR) is an extension of the Round-robin load balancing algorithm. Each server in load balancing can be assigned a weight, which is determined based on specific criteria chosen by the site administrator. The most commonly used criterion for assigning weights is the server's traffic-handling capacity. A higher weight indicates that the server is capable of handling a larger proportion of client requests, while a lower weight means it will receive a smaller proportion of the traffic. The weight assignment allows the load balancer to distribute the workload among servers in a way that reflects their individual capacities. Weighted round-robin load balancing is particularly advantageous in scenarios where servers have varying performance capacities. It allows you to distribute the workload in proportion to the capabilities of each server, ensuring a proportional allocation of the load.

IP Hash is an advanced load balancing method that makes use of a specific type of algorithm called a hashing algorithm. IP Hash is primarily designed to evenly distribute network load among multiple servers. It achieves this by employing a hashing algorithm to create a distinct hash value using the source and destination IP addresses of a packet. This generated hash value is then utilized to select a server from a pool of servers, ensuring a balanced distribution of network load. IP Hash is a popular method employed to maintain session persistence by ensuring that user sessions stay on the same server for the entire duration of their session.

1.3.2 Dynamic load balancing algorithms

Least connection load balancing is a dynamic load balancing algorithm that evenly distributes client requests to application servers based on the server's current number of active connections. When a client request is received, it is directed to the server with the fewest active connections at that moment. This approach is especially useful when the application servers have similar specifications, as it ensures that one server does not become overloaded due to longer-lived connections. Least connection load balancing is particularly well-suited for situations where incoming requests have different connection durations and when the servers in the pool have similar processing capabilities and available resources.

Weighted least connection load balancing enhances the basic least connection algorithm by considering the varying characteristics of application servers. In this approach, the administrator assigns a weight to each server in the server farm based on factors like processing power and available resources. The load balancer, such as LoadMaster, then takes into account both the number of active connections and the assigned server weights when making load balancing decisions. For example, if multiple servers have the lowest number of connections, the server with the highest weight will be selected. This weighting mechanism allows for more fine-grained control over the distribution of workload and ensures that servers with higher capabilities receive a proportionate share of the traffic. It is beneficial in environments where servers have diverse capacities, as it ensures a proportional distribution of workloads and maximizes resource utilization.

The weighted response time load balancing algorithm assigns server weights based on the response time of each application server. The server with the fastest response time is given a higher weight, and subsequently, it receives the next request. This algorithm is particularly suitable for situations where the primary concern is the application's response time.

The Least Response Time technique is a load balancing approach that directs incoming traffic to the server with the lowest response time or latency. It constantly monitors the performance of servers and routes requests to the server that can provide the fastest response, thereby ensuring an optimal real-time user experience. The Least Response Time technique is especially valuable for applications that demand minimal latency and high responsiveness, as it guarantees an optimal user experience in real-time environments. This technique prioritizes routing requests to the server with the lowest response time, enabling swift and efficient handling of user interactions.

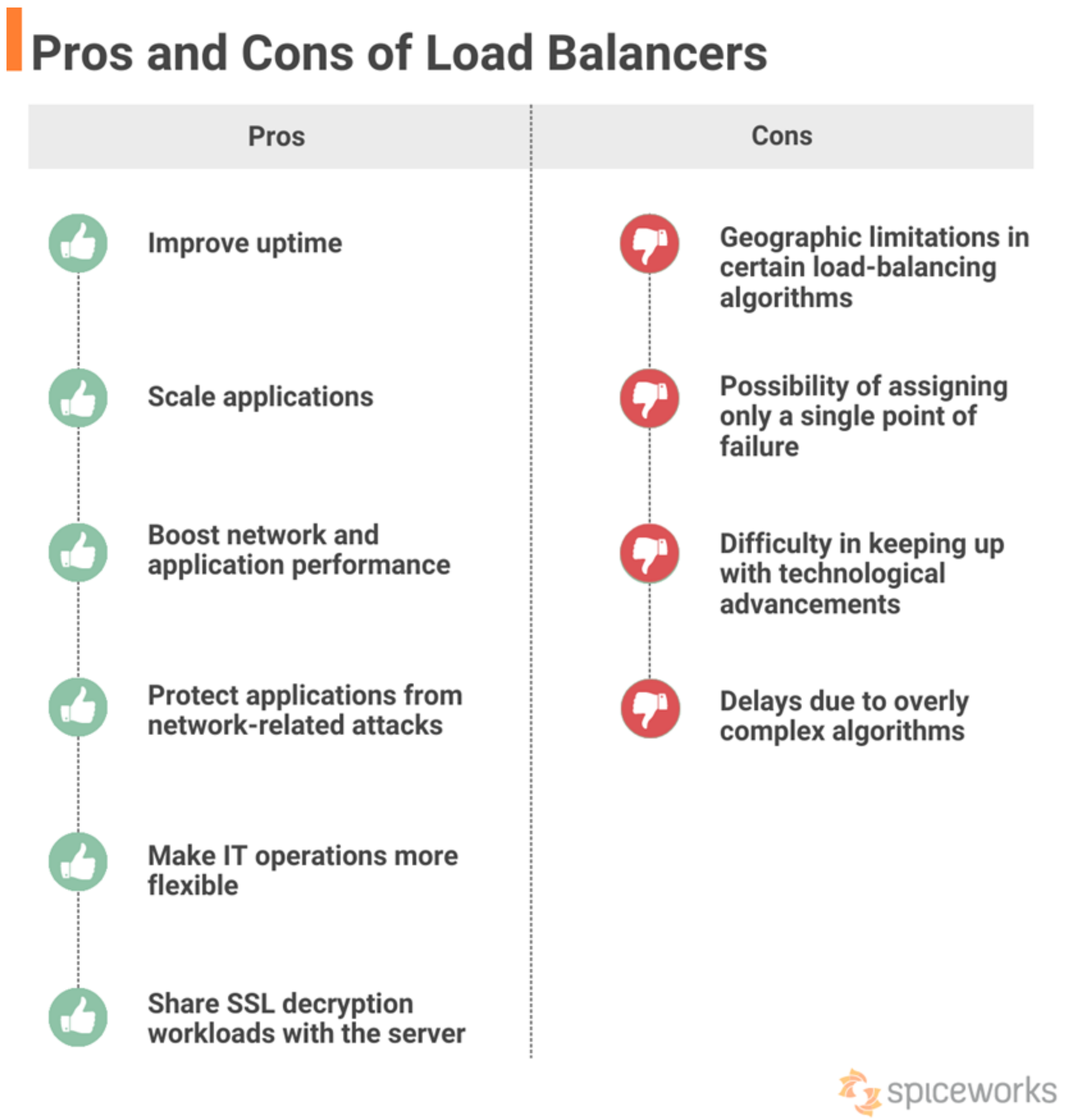
Resource-based (or adaptive) load balancing relies on status indicators obtained by LoadMaster from the back-end servers to make load balancing decisions. These status indicators are determined by a custom program, known as an "agent," running on each server. LoadMaster regularly queries each server for its status information and adjusts the dynamic weight of the server accordingly. This method is suitable for any situation that necessitates detailed health check information from each server in order to make informed load balancing decisions.

1.4 The function of load balancing

Load balancers manage incoming user requests for various services and information by acting as intermediaries between the internet and the servers responsible for handling those requests. Upon receiving a request, the load balancer assesses the availability and online status of the servers within a designated pool and subsequently directs the request to an appropriate server. In periods of high demand, load balancers respond swiftly by dynamically incorporating additional servers to efficiently handle the surge in traffic. Conversely, load balancers can also remove servers from the pool during periods of low demand to optimize resource allocation.

2. Pros and Cons

This category contains the analysis of the advantages with disadvantages of load balancers. Below is a contrasting illustration of them:



Pros and Cons of Load Balancers

Pros	Cons
Improve uptime	Geographic limitations in certain load-balancing algorithms
Scale applications	Possibility of assigning only a single point of failure
Boost network and application performance	Difficulty in keeping up with technological advancements
Protect applications from network-related attacks	Delays due to overly complex algorithms
Make IT operations more flexible	
Share SSL decryption workloads with the server	

Figure 2: Load Balancers Advantages and Disadvantages

2.1 Pros of Load Balancers

Improve Uptime:

- Fault Resilience: Load balancers automatically detect server issues and redirect client traffic to healthy servers, enhancing system fault resilience.
- Zero Disruption Maintenance: Server repairs or upgrades can be carried out without disrupting service by rerouting traffic to alternative servers.
- Disaster Recovery: Load balancers can proactively direct traffic to backup sites in case of disasters, ensuring continuous service availability.

Scale Applications with Confidence:

- Intelligent Traffic Distribution: Load balancers distribute network traffic intelligently across multiple servers, eliminating congestion and allowing applications to handle large numbers of client requests seamlessly.
- Dynamic Scaling: Load balancers can dynamically adjust to changing traffic patterns, allowing you to add or remove servers based on demand.

Boost Network and Application Performance:

- Reduced Latency: Load balancers lower network latency and improve response time by distributing workload proportionately across servers.
- Geographic Load Balancing: Configuring load balancers to route client requests to geographically closer servers significantly reduce application latency.

Protect Applications from Network-Related Attacks:

- Security Protocols: Load balancers include built-in security protocols to protect web applications, making them effective against DDoS attacks.
- Traffic Monitoring: Load balancers monitor traffic and can block malicious content, providing an additional layer of security.

Deliver Better User Experience (UX):

- Resource Optimization: Load balancers optimize resource consumption, data transmission, and response time, offering end-users a seamless and responsive experience.
- High-Traffic Optimization: In high-traffic scenarios, load balancing ensures the precise execution of user requests, avoiding sluggish or unresponsive applications.

Make IT Operations More Flexible:

- Server Maintenance: Load balancers enable IT administrators to perform server maintenance without affecting site availability by ensuring at least one server is always accessible during maintenance.
- DevOps Support: In a DevOps culture, load balancers facilitate frequent application changes by providing flexibility in managing website traffic.

Share SSL Decryption Workloads with the Server:

- Efficient SSL Processing: Load balancers handle SSL decryption, allowing web servers to focus on processing decrypted data and conserving computational resources.

2.2 Cons of Load Balancers

Geographic Limitations in Certain Load-Balancing Algorithms:

- Communication Challenges: Some load-balancing algorithms are not suitable for remote locations, leading to communication interruptions and network pauses.
- Centralized Decision Making: Certain protocols may rely on a centralized node, posing challenges in dispersed node scenarios.

Possibility of Assigning Only a Single Point of Failure:

- Central Node Dependency: Some load-balancing protocols depend on a central node for decision-making, making the system vulnerable to a single point of failure.
- Mitigation Strategy: Configuring multiple nodes to handle traffic and routing decisions helps mitigate this risk.

Difficulty in Keeping Up with Technological Advancements:

- Evolving Technology Landscape: Load balancers need to adapt to changing consumer requirements, cloud services, IoT, high-bandwidth applications, and blockchain, which can be challenging to keep up with.

Delays Due to Overly Complex Algorithms:

- Algorithmic Complexity: Some load-balancing algorithms can be overly complex, impacting migration time, defect tolerances, and speed of response.
- Optimization Challenge: Balancing the algorithm's complexity with the need for optimal system performance is crucial.

With the ability to enhance system stability, improve application performance, and defend against network attacks, load balancers optimize user experience and minimize system downtime. Their flexibility in managing traffic, especially in DevOps environments, is another notable advantage. However, it's crucial to be mindful of challenges such as geographic limitations in certain load balancing algorithms, the risk of a single point of failure in some protocols, adapting to the rapidly evolving technology landscape, and the complexity of certain load balancing algorithms that may introduce delays and impact system performance. Therefore, integrating load balancers demands meticulous consideration to maximize their benefits in the context of ensuring resilience against the increasingly complex challenges of modern networking environments.

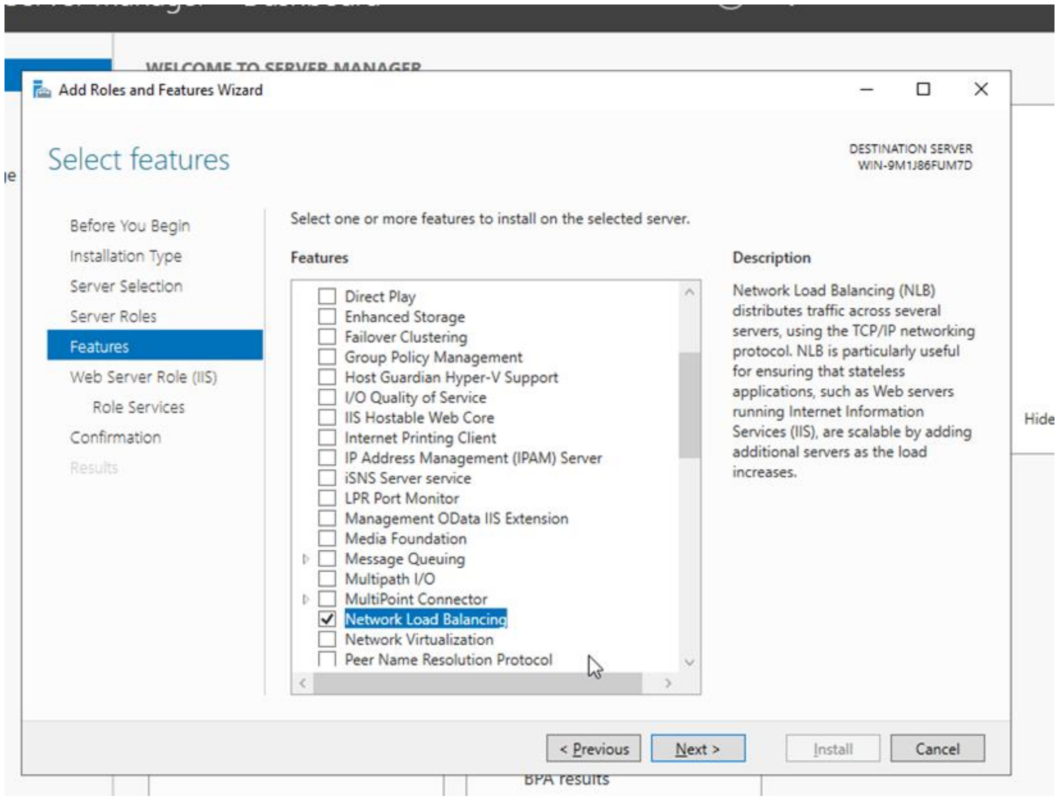
3. Installation and Configuration

The environment used: Windows Server 2019

In this installation, we use **3 windows servers** (there are 3 virtual machines utilized, which include **2 nodes – NLB01 and NLB02** in the cluster which can be used to automatically substitute when the other one does not response, and the last one is used to test their function):

+ Host: 192.168.204.138
+ NLB01: 192.168.204.141
+ NLB02: 192.168.204.142

First, we install 2 services **- Internet Information Services** and **Network Load Balancing** in server manager of 2 windows servers:



Add Roles and Features Wizard

Select features

DESTINATION SERVER
WIN-9M1J86FUM7D

Before You Begin
Installation Type
Server Selection
Server Roles
Features
Web Server Role (IIS)
Role Services
Confirmation
Results

Select one or more features to install on the selected server.

Features

- [ ] Direct Play
- [ ] Enhanced Storage
- [ ] Failover Clustering
- [ ] Group Policy Management
- [ ] Host Guardian Hyper-V Support
- [ ] I/O Quality of Service
- [ ] IIS Hostable Web Core
- [ ] Internet Printing Client
- [ ] IP Address Management (IPAM) Server
- [ ] iSNS Server service
- [ ] LPR Port Monitor
- [ ] Management OData IIS Extension
- [ ] Media Foundation
- [ ] Message Queuing
- [ ] Multipath I/O
- [ ] MultiPoint Connector
- [x] Network Load Balancing
- [ ] Network Virtualization
- [ ] Peer Name Resolution Protocol

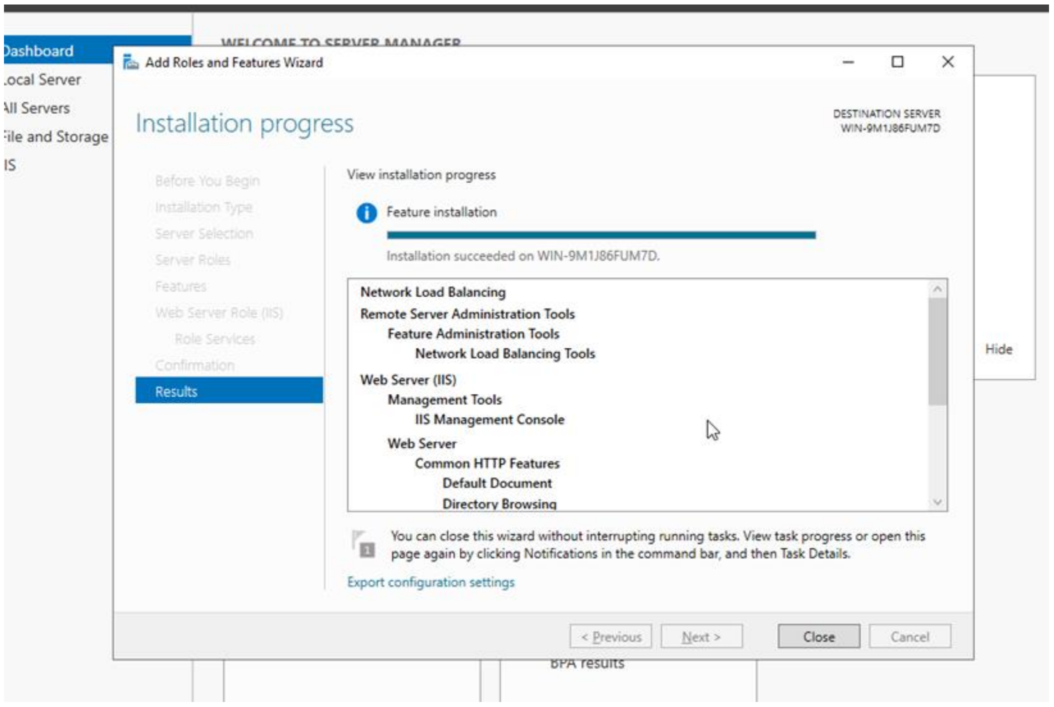
Description

Network Load Balancing (NLB) distributes traffic across several servers, using the TCP/IP networking protocol. NLB is particularly useful for ensuring that stateless applications, such as Web servers running Internet Information Services (IIS), are scalable by adding additional servers as the load increases.

< Previous | Next > | Install | Cancel

Figure 3: Installing 2 services IIS and NLB

This is the result:



Add Roles and Features Wizard

Installation progress

DESTINATION SERVER
WIN-9M1J86FUM7D

View installation progress

Feature installation

Installation succeeded on WIN-9M1J86FUM7D.

Network Load Balancing
Remote Server Administration Tools
    Feature Administration Tools
        Network Load Balancing Tools
Web Server (IIS)
    Management Tools
        IIS Management Console
    Web Server
        Common HTTP Features
            Default Document
            Directory Browsing

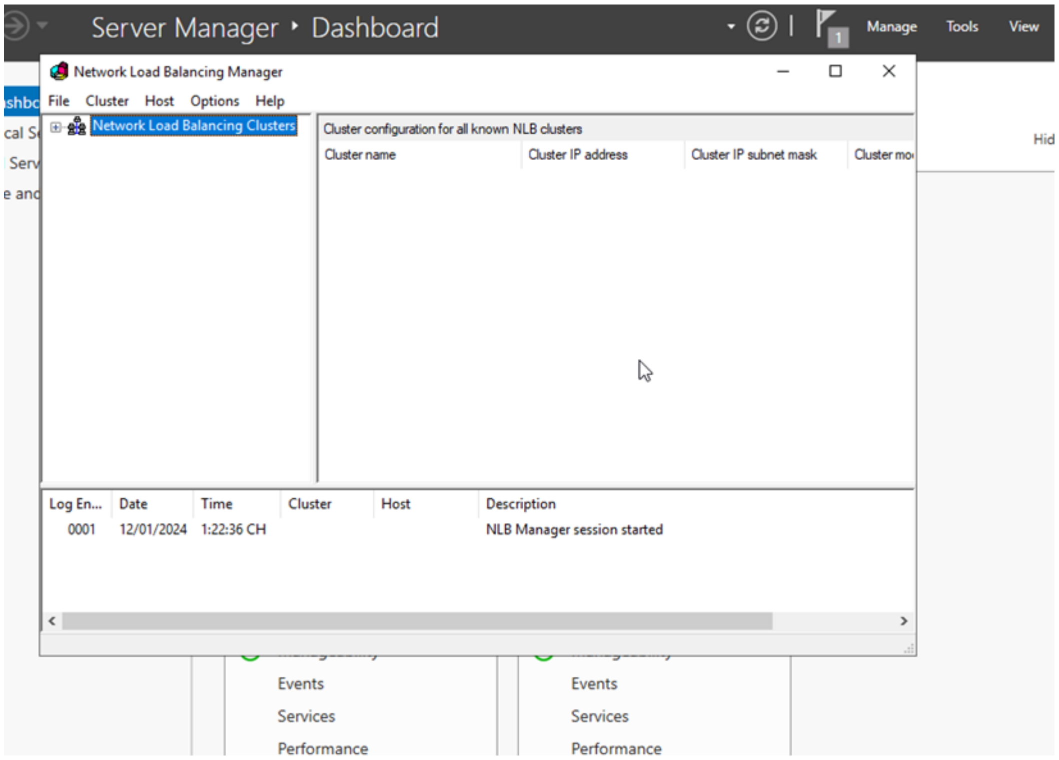
You can close this wizard without interrupting running tasks. View task progress or open this page again by clicking Notifications in the command bar, and then Task Details.

Export configuration settings

< Previous | Next > | Close | Cancel

Figure 4: Successful installation

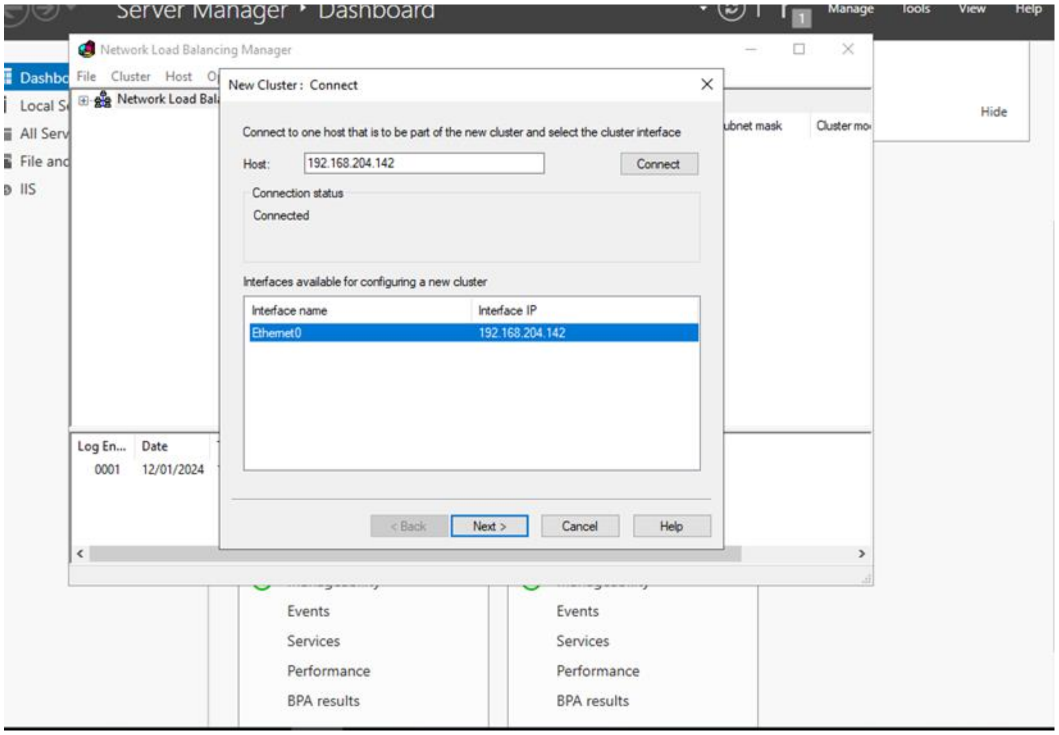
After that, we select Tool -> Network load balancing manager:


Server Manager ‣ Dashboard
Manage
Tools
View
Network Load Balancing Manager
File
Cluster
Host
Options
Help
Network Load Balancing Clusters
Cluster configuration for all known NLB clusters
Cluster name
Cluster IP address
Cluster IP subnet mask
Events
Services
Performance

Log En...	Date	Time	Cluster	Host	Description
0001	12/01/2024	1:22:36 CH			NLB Manager session started

Figure 5: Selecting Network Load Balancing manager

To create a new cluster, we enter the IP address of NLB02:



Connect to one host that is to be part of the new cluster and select the cluster interface

Host: 192.168.204.142

Connection status

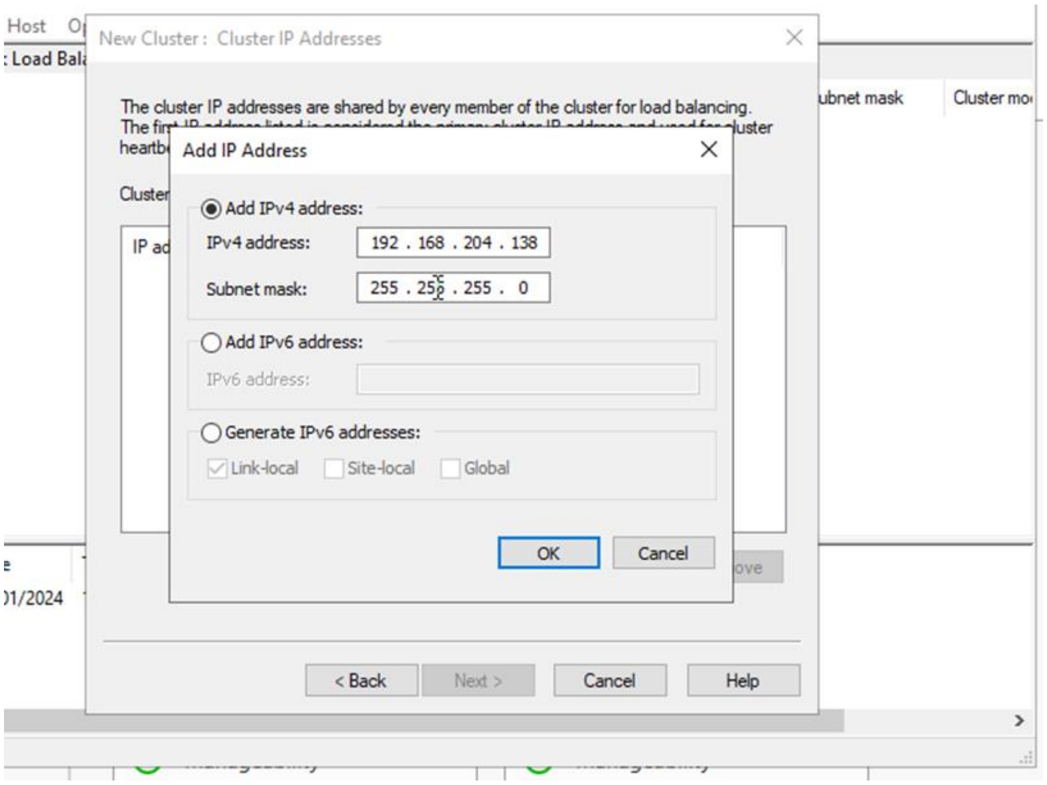
Connected

Interfaces available for configuring a new cluster

Interface name	Interface IP
Ethernet0	192.168.204.142

Figure 6: Creating a new node NLB02

Then, we enter the IP address of the host:



The cluster IP addresses are shared by every member of the cluster for load balancing.

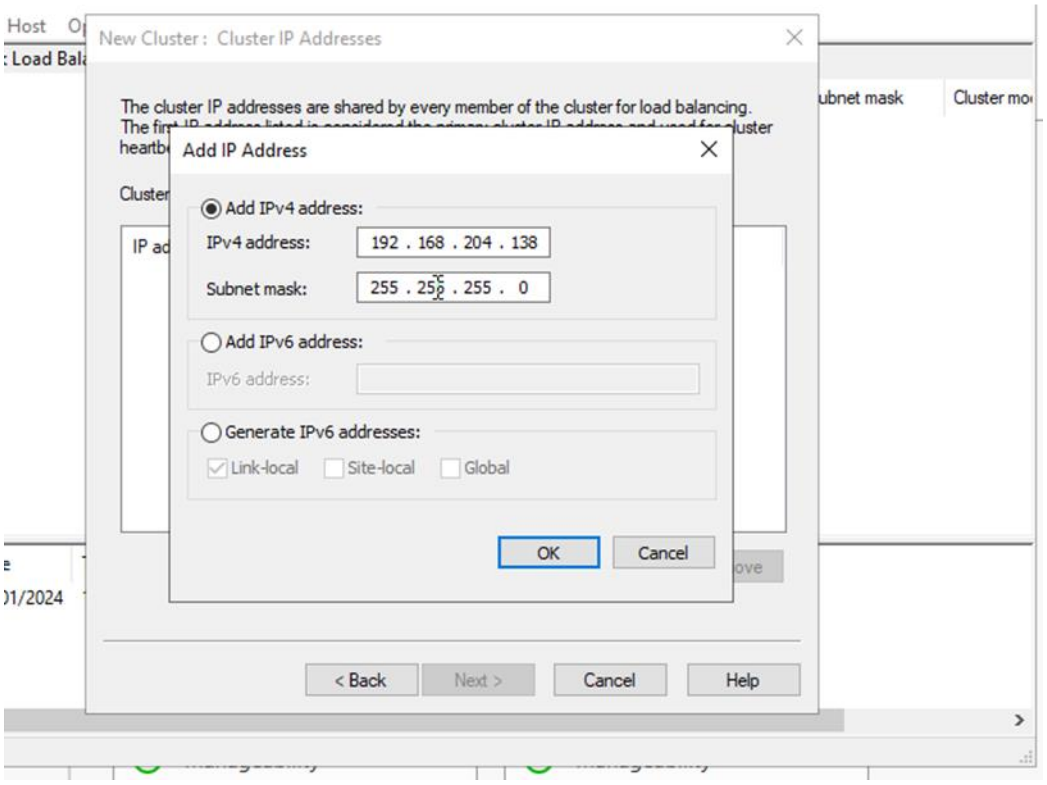
Add IPv4 address: [✓]
IPv4 address: 192 . 168 . 204 . 138
Subnet mask: 255 . 255 . 255 . 0

Add IPv6 address:
IPv6 address:

Generate IPv6 addresses:
[✓] Link-local [ ] Site-local [ ] Global

Figure 7: Adding cluster IP address

Next, we edit the port of cluster in the Add/Edit Port Rule section and add new port with the range from 443 to 443.


New Cluster : Cluster IP Addresses

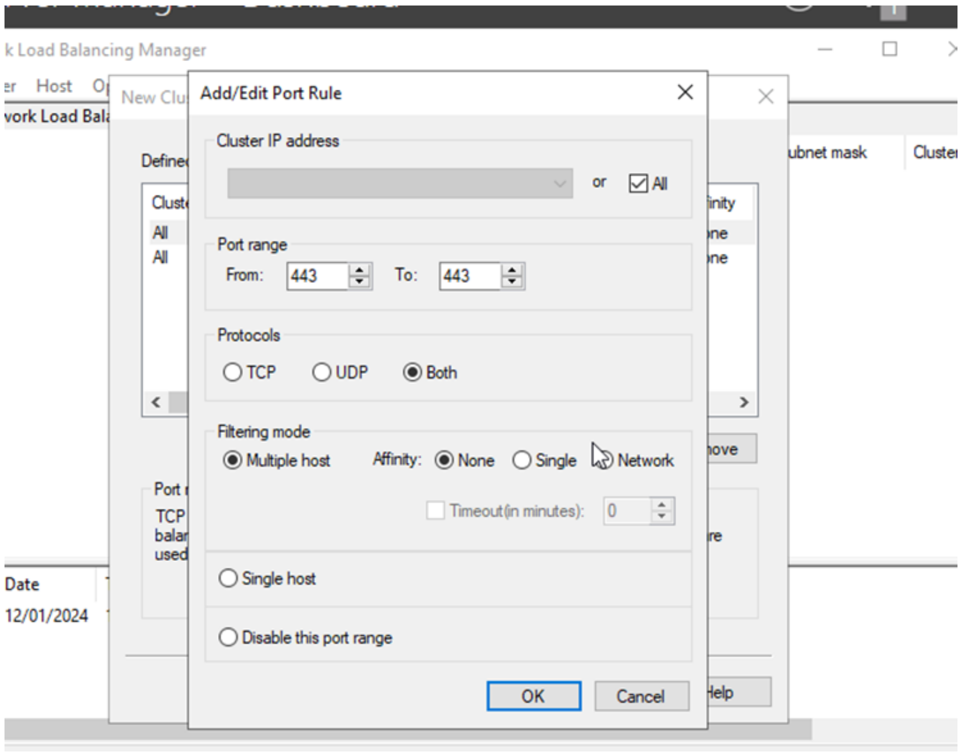
The cluster IP addresses are shared by every member of the cluster for load balancing.

Add IP Address

Add IPv4 address: [✓]
IPv4 address: 192 . 168 . 204 . 138
Subnet mask: 255 . 255 . 255 . 0

Add IPv6 address:
IPv6 address:

Generate IPv6 addresses:
[✓] Link-local [ ] Site-local [ ] Global



All [✓]

Port range: From: 443 To: 443

Protocols: TCP UDP [✓ Both]

Filtering mode: [✓ Multiple host] Affinity: [✓ None] Single Network

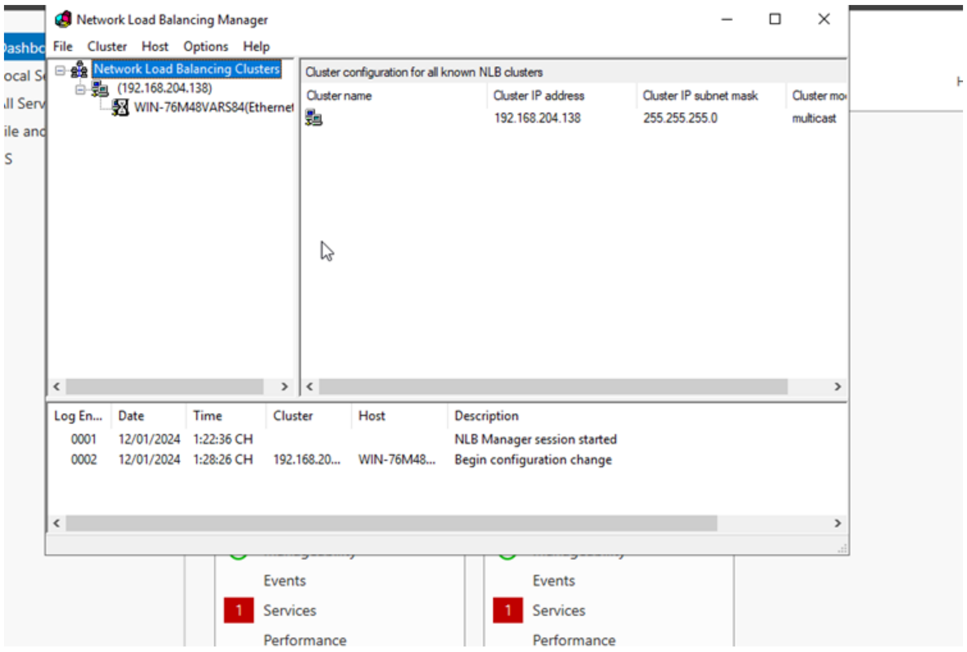
Timeout (in minutes): 0

Single host

Disable this port range

Figure 8: Editing Port range of the cluster

This is the result after successfully adding the port for NLB02:



Cluster configuration for all known NLB clusters

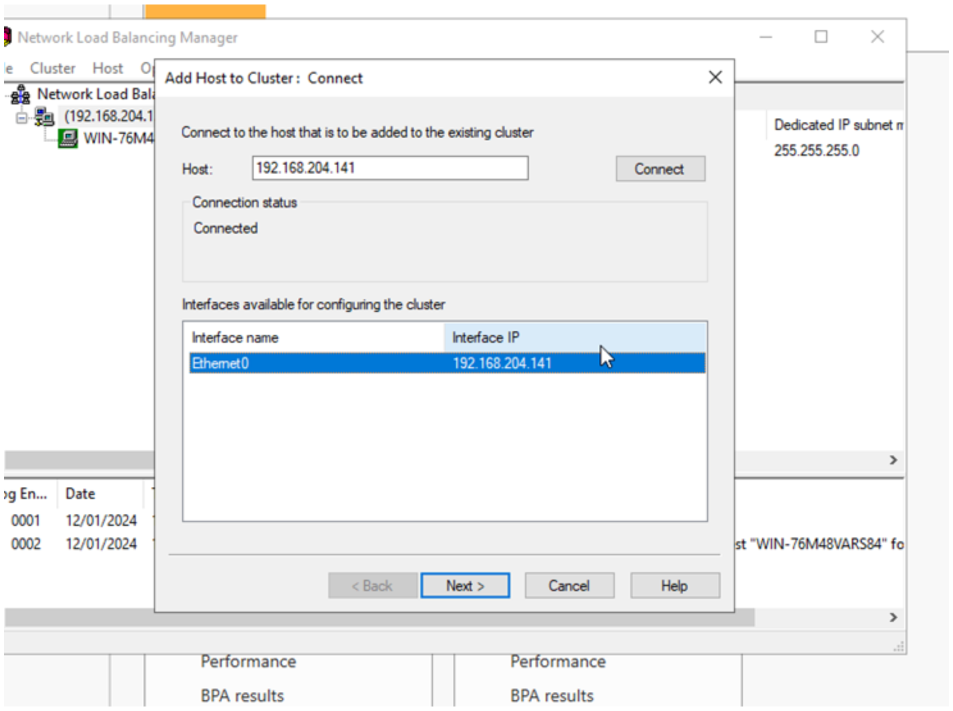
Cluster name	Cluster IP address	Cluster IP subnet mask	Cluster mo
	192.168.204.138	255.255.255.0	multicast

Log En...	Date	Time	Cluster	Host	Description
0001	12/01/2024	1:22:36 CH			NLB Manager session started
0002	12/01/2024	1:28:26 CH	192.168.20...	WIN-76M48...	Begin configuration change

Figure 9: Successful creation for NLB02 port

In the next step, we enter the network load balancing of NLB01 and add a host into the cluster.

Firstly, we enter the IP address of NLB01:



Connect to the host that is to be added to the existing cluster

Host: 192.168.204.141 Connect

Connection status

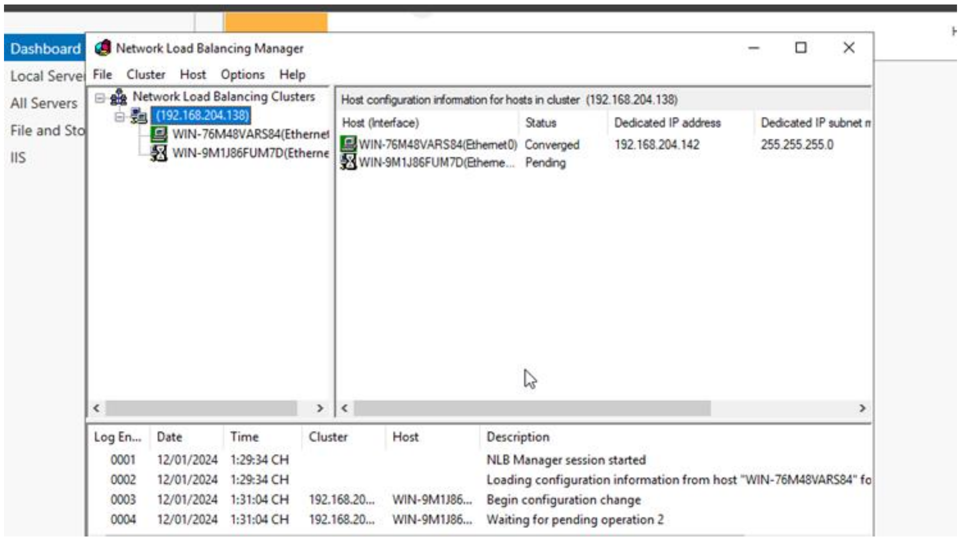
Connected

Interfaces available for configuring the cluster

Interface name	Interface IP
Ethernet0	192.168.204.141

Figure 10: Creating a new node NLB01

Similar to the previous step, this is the result after successfully adding the port for NLB01:



Host configuration information for hosts in cluster (192.168.204.138)

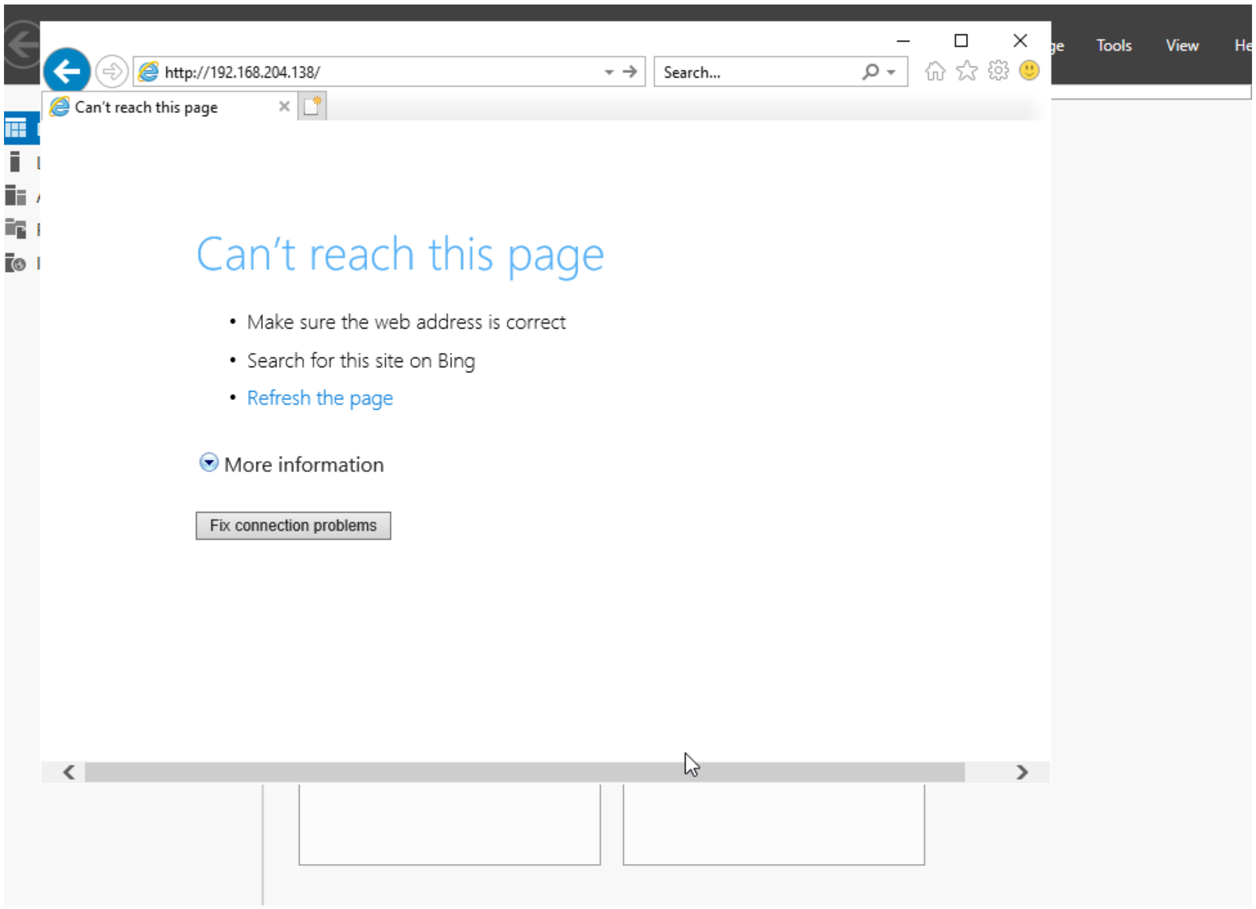
Host (Interface)	Status	Dedicated IP address	Dedicated IP subnet m
WIN-76M48VARS84(Ethernet0)	Converged	192.168.204.142	255.255.255.0
WIN-9M1J86FUM7D(Etherne...	Pending		

Log En...	Date	Time	Cluster	Host	Description
0001	12/01/2024	1:29:34 CH			NLB Manager session started
0002	12/01/2024	1:29:34 CH			Loading configuration information from host "WIN-76M48VARS84" fo
0003	12/01/2024	1:31:04 CH	192.168.20...	WIN-9M1J86...	Begin configuration change
0004	12/01/2024	1:31:04 CH	192.168.20...	WIN-9M1J86...	Waiting for pending operation 2

Figure 11: A cluster involving 2 nodes

4. Demo

When **Not** start the host for this public IP:192.168.204.138



Can't reach this page

- Make sure the web address is correct
- Search for this site on Bing
- Refresh the page

More information

Fix connection problems

Figure 12: Turn off 2 nodes in cluster (NBL01-NBL02)

When **open** 2 host, NBL02 will come first because it has been given higher priority

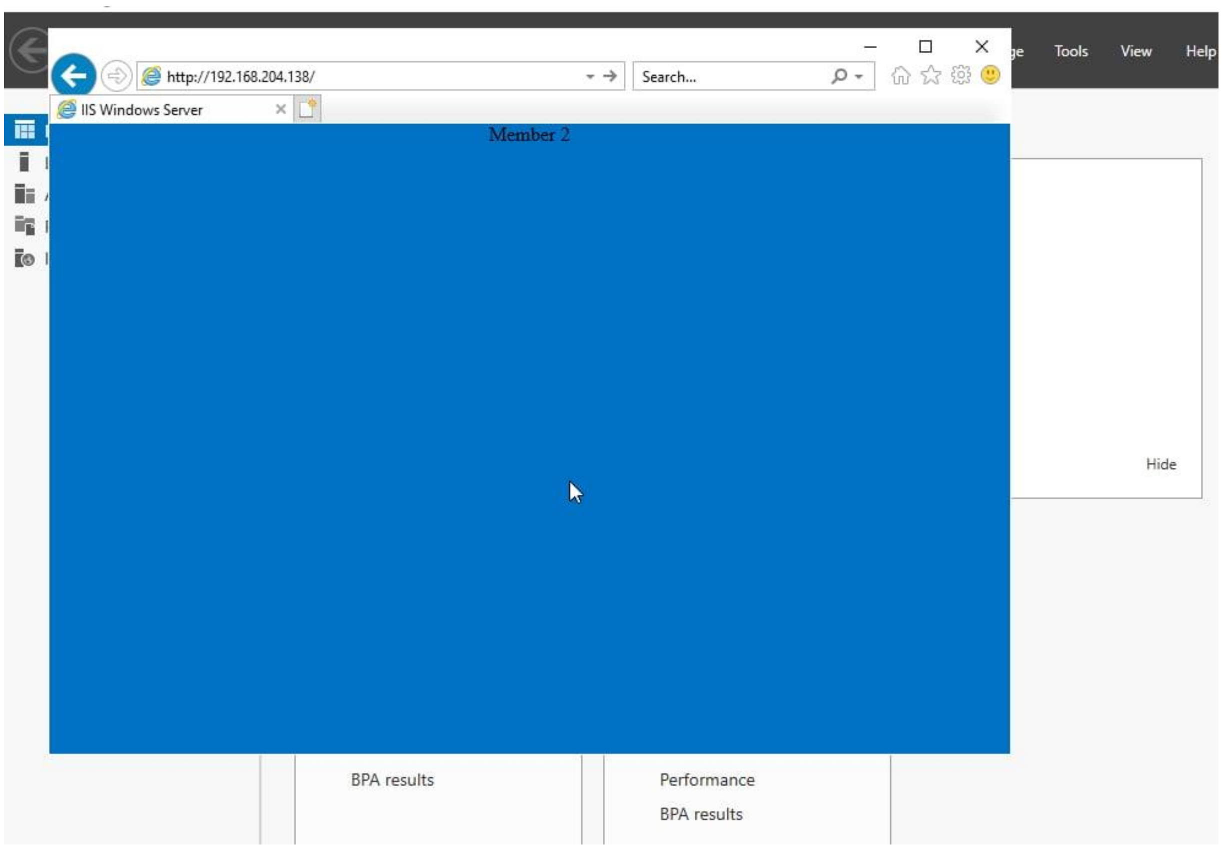


Figure 13: Turn on 2 nodes in cluster (NBL01-NBL02)

Then, I will **turn off NLB02**, we can see NLB01 come and replace for NLB02

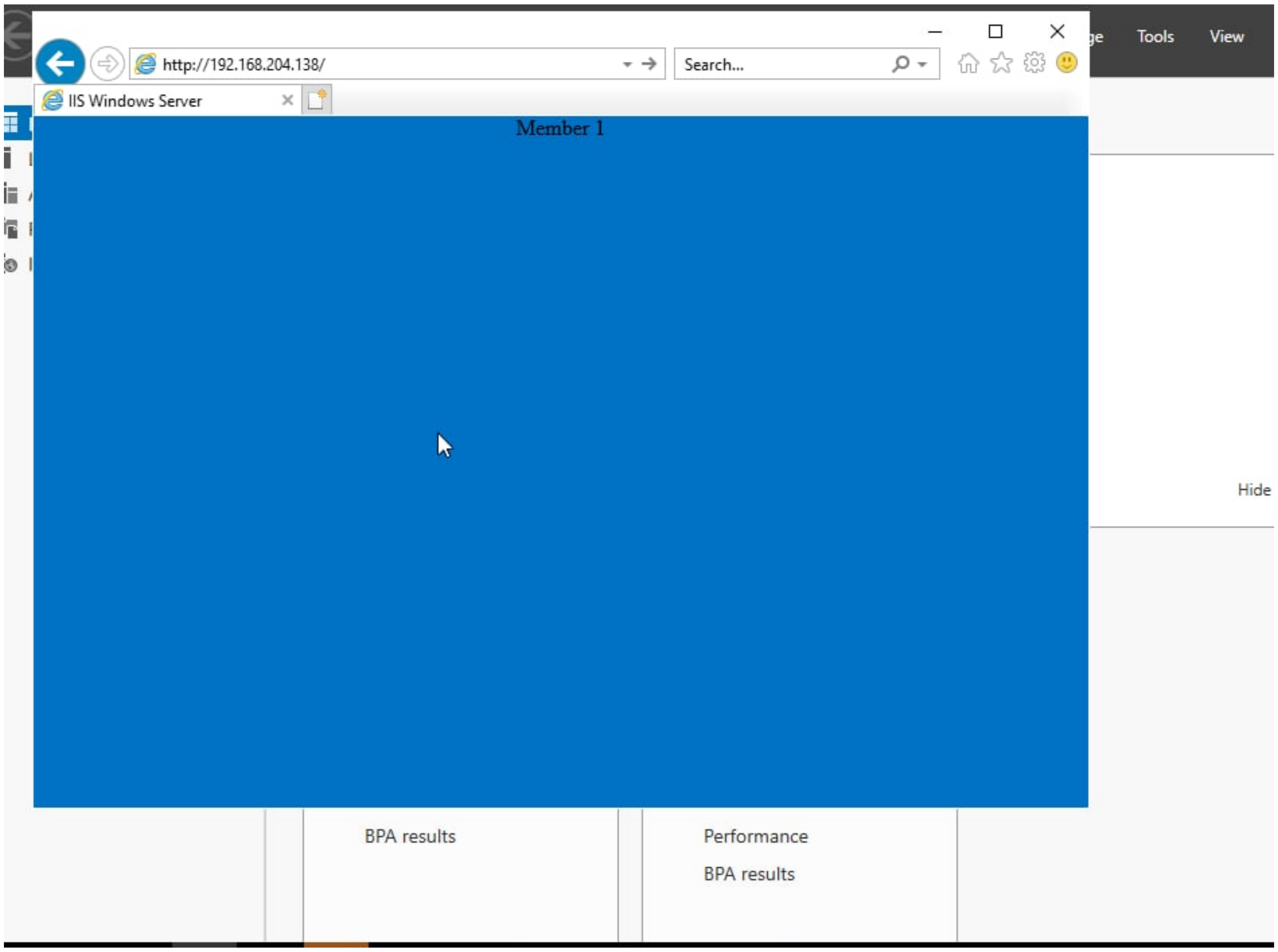


Figure 14: Turn off 2 NBL02

C. Conclusion

In conclusion, load balancing is a critical technique in modern computing systems that helps distribute workload efficiently across multiple resources, such as servers, networks, or storage devices. It plays a vital role in improving performance, maximizing resource utilization, enhancing scalability, and ensuring high availability of applications and services. Throughout this report, we have explored various load balancing algorithms and techniques. Each algorithm has its considerations, and the choice of the algorithm depends on the specific requirements of the system and the workload characteristics. Additionally, load balancing solutions can be hardware-based, software-based, or a combination of both, offering flexibility and scalability to meet diverse workload demands. We then implemented the web server load balancing based on 2 services and showed that with the network load balancer utilization, the network traffic is distributed efficiently and the web page is always available and responsive.

References

BasuMallick, C. (2023). *What Is a Load Balancer? Definition, Working, Benefits, and Drawbacks*. Retrieved 2023, from https://www.spiceworks.com/tech/networking/articles/what-is-a-load-balancer/

How does a load balancer distribute client traffic across servers? (n.d.). Retrieved 2024, from https://kemptechnologies.com/load-balancer/load-balancing-algorithms-techniques

Kinza Yasar, A. I. (n.d.). *DEFINITION: load balancing*. Retrieved 2024, from https://www.techtarget.com/searchnetworking/definition/load-balancing

Majeed, A. (2023). *Load Balancing Algorithms*. Retrieved 2024, from https://www.linkedin.com/pulse/load-balancing-algorithms-ahmed-majeed

Nek, D. (n.d.). *What is IP Hash?* Retrieved 2024, from https://webhostinggeeks.com/blog/what-is-ip-hash/

The Role of Network Load Balancing in Reducing Latency. (2023). Retrieved 2023, from https://utilitiesone.com/the-role-of-network-load-balancing-in-reducing-latency

Types of load balancing algorithms. (n.d.). Retrieved 2023, from https://www.cloudflare.com/learning/performance/types-of-load-balancing-algorithms/

What Is Load Balancing? (n.d.). Retrieved 2023, from https://www.nginx.com/resources/glossary/load-balancing/

What Is Round-Robin Load Balancing? (n.d.). Retrieved 2023, from https://www.nginx.com/resources/glossary/round-robin-load-balancing/