**Research and Development Document**

**ON**

**Azure Load Balancer**

****

****

**By**

**Anirudh Sharma**

**(CT\_CSI\_CI\_66)**

**Under**

**Celebal Summer Internship**

**in**

**Cloud Infra & Security**

**Table of Contents**

**Load Balancing 3**

**Azure Load Balancer 3**

**Components of Azure Load Balancer 3**

**Azure Load Balancing Services 4**

**Choosing a Load Balancer Service in Azure 5**

**Types of Azure Load Balancer 5**

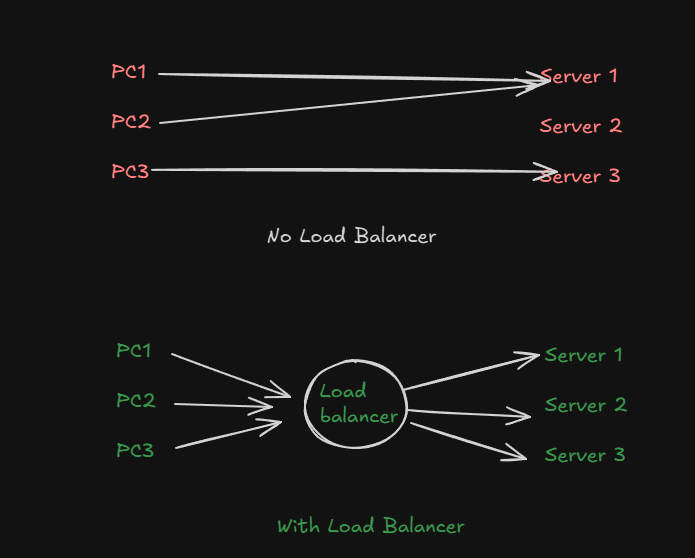
**Assignment 6**

**References 14**

**Load Balancing**

Load balancing is the practice of distributing network traffic or computational workloads across multiple servers to improve an application's performance and reliability using a load balancer. It is the practice of distributing computational workloads between two or more computers. A load balancer is a networking device or software application that distributes and balances the incoming traffic among the servers to provide high availability, efficient utilization of servers, and high performance. The term load balancing refers to the distribution of workloads across multiple computing resources. Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overloading any single resource. It can also improve availability by sharing a workload across redundant computing resources.

**Use of a load balancer:**



**Azure Load Balancer**

Azure Load Balancer is a cloud service that distributes incoming network traffic across backend virtual machines (VMs) .

Load balancing refers to efficiently distributing incoming network traffic across a group of backend virtual machines (VMs). It operates at **layer 4** of the Open Systems Interconnection (OSI) model.

**Components of Azure Load Balancer:**

* **Frontend IP configuration**: The nature of the IP address determines the type of load balancer created. Private IP address selection creates an internal load balancer. Public IP address selection creates a public load balancer.

## **Backend pool:** The group of virtual machines or instances in a virtual machine scale set that is serving the incoming request. To scale cost-effectively to meet high volumes of incoming traffic, computing guidelines generally recommend adding more instances to the backend pool. Load balancer instantly reconfigures itself via automatic reconfiguration when you scale instances up or down. Adding or removing VMs from the backend pool reconfigures the load balancer without other operations. **The scope of the backend pool is any virtual machine in a single virtual network.**

* **Health probes:** A health probe is used to determine the health status of the instances in the backend pool. During load balancer creation, configure a health probe for the load balancer to use. This health probe determines if an instance is healthy and can receive traffic.

## **Load Balancer rules:** A load balancer rule is used to define how incoming traffic is distributed to all the instances within the backend pool. A load-balancing rule maps a given frontend IP configuration and port to multiple backend IP addresses and ports. Load Balancer rules are for inbound traffic only.

## **High Availability Ports:** A load balancer rule configured with 'protocol - all and port - 0' is known as a High Availability (HA) port rule. This rule enables a single rule to load-balance all TCP and UDP flows that arrive on all ports of an internal Standard Load Balancer.

The load-balancing decision is made per flow. This action is based on the following five-tuple connection:

1. **source IP address**
2. **source port**
3. **destination IP address**
4. **destination port**
5. **protocol**

## **Inbound NAT rules:** An inbound NAT rule forwards incoming traffic sent to frontend IP address and port combination. The traffic is sent to a specific virtual machine or instance in the backend pool. Port forwarding is done by the same hash-based distribution as load balancing.

## **Outbound rules:** An outbound rule configures outbound Network Address Translation (NAT) for all virtual machines or instances identified by the backend pool. This rule enables instances in the backend to communicate (outbound) to the internet or other endpoints.

## **Azure load-balancing services**

* **Azure Front Door** is an application delivery network that provides global load balancing and site acceleration service for web applications. It offers Layer 7 capabilities for your application like SSL offload, path-based routing, fast failover, and caching to improve performance and high availability of your applications.
* **Traffic Manager** is a DNS-based traffic load balancer that enables you to distribute traffic optimally to services across global Azure regions, while providing high availability and responsiveness. Because Traffic Manager is a DNS-based load-balancing service, it load balances only at the domain level. For that reason, it can't fail over as quickly as Azure Front Door, because of common challenges around DNS caching and systems not honouring DNS TTLs.
* **Application Gateway** provides application delivery controller as a service, offering various Layer 7 load-balancing capabilities and web application firewall functionality. Use it to transition from public network space into your web servers hosted in private network space within a region.
* Load Balancer is a high-performance, ultra-low-latency Layer 4 load-balancing service (inbound and outbound) for all UDP and TCP protocols. It's built to handle millions of requests per second while ensuring your solution is highly available.

**Choosing a Load Balancer in Azure:**

## **Service categorizations:**

Azure load-balancing services can be categorized along two dimensions: global versus regional and HTTP(S) versus non-HTTP(S).

### **Global vs. regional**

* **Global:** These load-balancing services distribute traffic across regional back-ends, clouds, or hybrid on-premises services. These services support managing a single control plane responsible for globally routing end-user traffic to an available back-end. They often react to changes in service reliability or performance to maximize availability and performance. You can think of them as systems that load balance between application stamps, endpoints, or scale-units hosted across different regions/geographies.
* **Regional:** These load-balancing services distribute traffic within virtual networks across virtual machines (VMs) or zonal and zone-redundant service endpoints within a region. You can think of them as systems that load balance between VMs, containers, or clusters within a region in a virtual network.

### **HTTP(S) vs. non-HTTP(S)**

* **HTTP(S):** These load-balancing services are Layer 7 load balancers that only accept HTTP(S) traffic. They're intended for web applications or other HTTP(S) endpoints. They might have features such as SSL offload, web application firewall, path-based load balancing, and session affinity.
* **Non-HTTP(S):** These load-balancing services are either Layer 4 TCP or UDP services, or DNS-based load balancing.

**Consider these factors such as these when you select a load balancing solution:**

* **Traffic type:** Is it a web HTTP(S) application? Is it public facing or a private application?
* **Global vs. regional**: Do you need to load balance VMs or containers within a single virtual network, or load balance scale unit/deployments across regions, or both?
* **Availability**
* **Cost**
* **Features and limits**

**Types of Azure Load Balancer:**

* **Public load balancer**
* **Internal/ private load balancer**

**External/Public Load Balancer:** A public load balancer can be used to load balance internet traffic to virtual machines. It can provide outbound connections for virtual machines (VMs) inside your virtual network.

**Internal/ Private Load Balancer:** An internal (or private) load balancer is used to balance traffic from within a virtual network.

Along with load balancer, there are two pricing tiers (SKU’s)available **Basic** and **Standard:**

**Basic:** Basic tier load balancer provides basic features and is restricted to some limits for backend pool size; it is restricted to only 300 instances; it’s restricted to a single availability set and it only supports multiple frontends for inbound traffic. This will be retiring soon

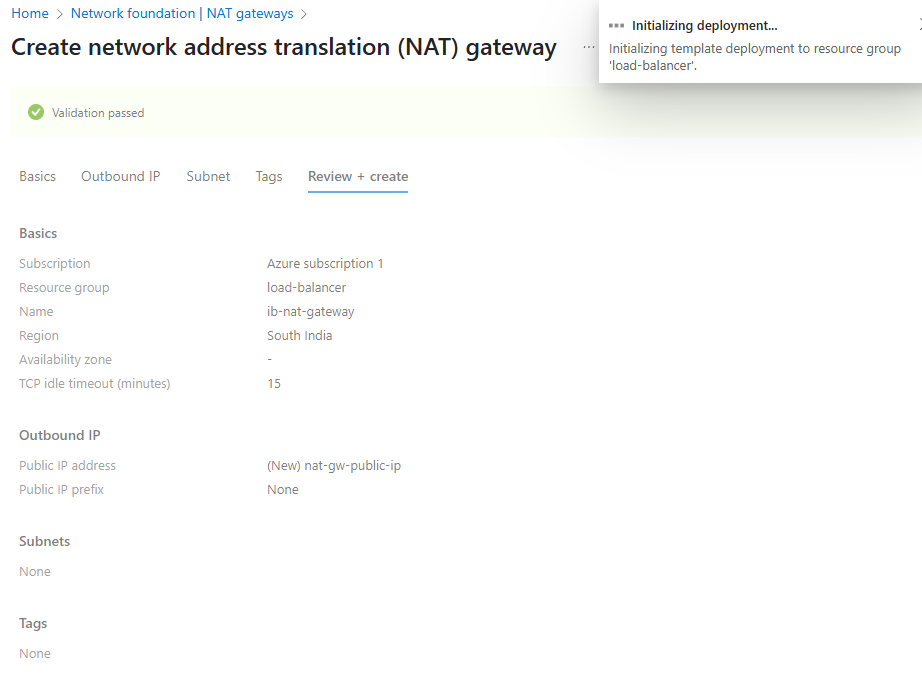
**Standard:** Standard tier load balancer is generally available and offers higher-scale and new features. It is a paid-for feature using a complex set of consumption-based charges and the Basic tier continues to be free. Also, we can scale out to 1000 instances and can span any virtual machine in a single virtual network, including blends of scale sets, availability sets, and machines.

**Features of Azure Load Balancer:**

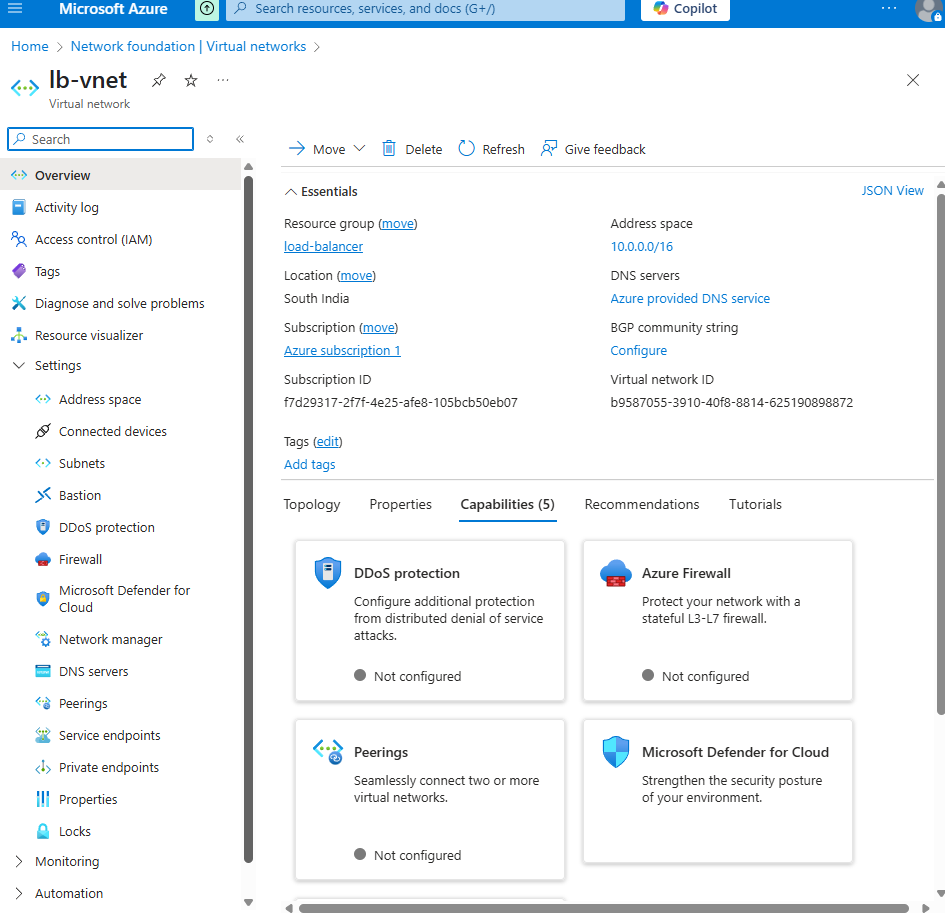
* **Load Balancing:** Azure load balancer uses a 5-tuple hash that contains source IP, source port, destination IP, destination port, and protocol.
* **Outbound connection:** All the outbound flows from a private IP address inside our virtual network to public IP addresses on the Internet can be translated to a frontend IP of the load balancer.
* **Automatic reconfiguration:** The load balancer is able to reconfigure itself when it scales up or down instances on the basis of conditions. So, if more virtual machines are added into the backend pool, automatically load balancer will reconfigure.
* **Application agnostic and transparent:** It doesn’t directly interact with TCP or UDP protocols. We can route the traffic based on URL or multi-site hosting
* **Health probes:** When any failed virtual machines in a load balancer are recognized by the health probe in the backend pool then it stops routing the traffic to that particular failed virtual machine. It can configure a health probe to determine the health of the instances in the backend pool.
* **Port forwarding:** The load balancer supports port forwarding ability if we have a pool of web servers, and we don’t want to attach a public IP address for every web server in that pool.

**Assignment: Create an Internal & External Load balancer, also Verify It is working.**

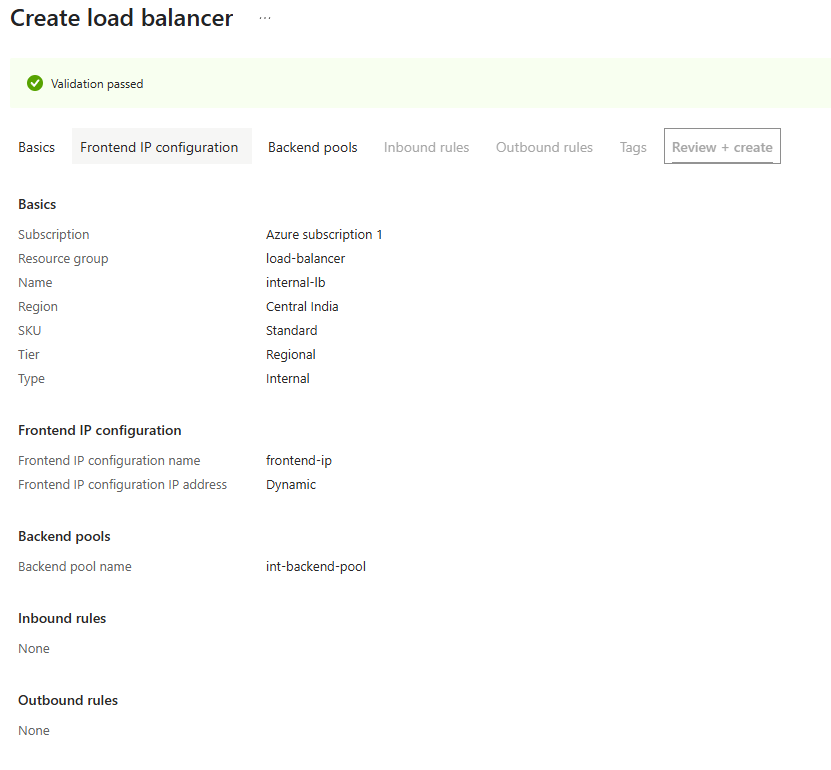
**Step 1: Create NAT Gateway**

****

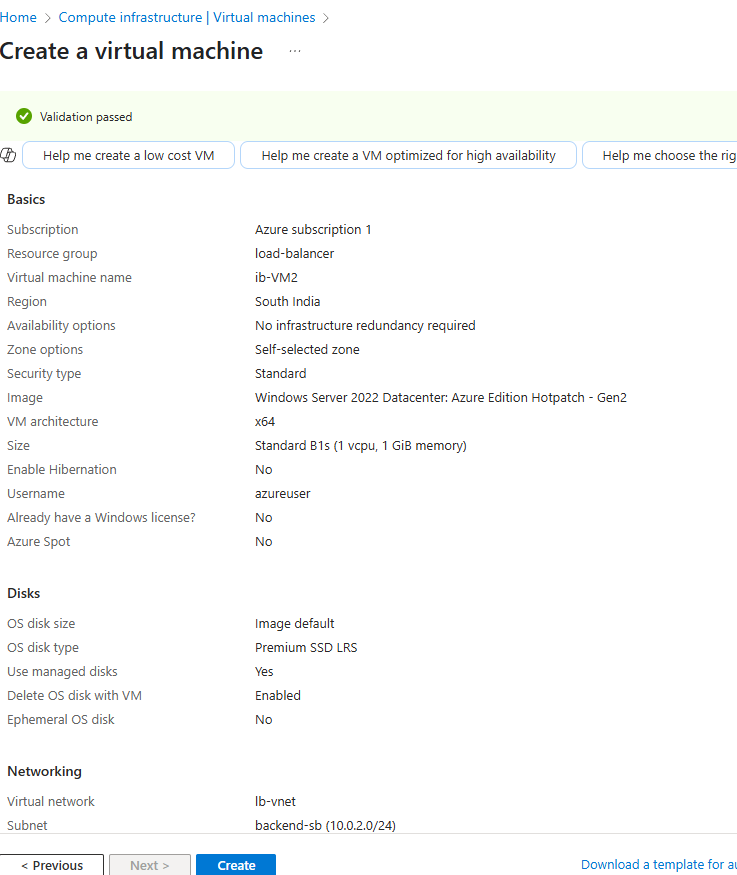
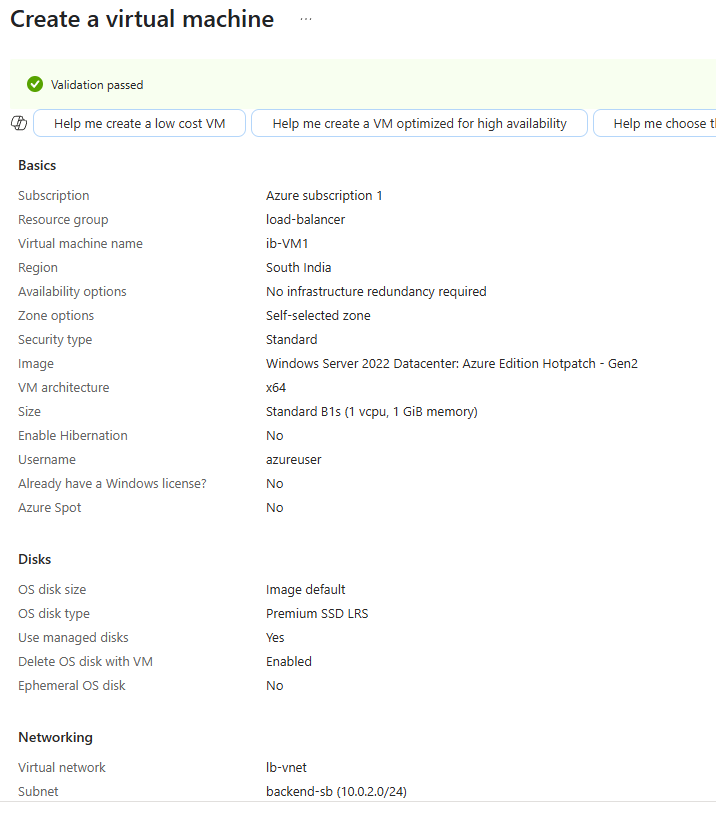
**Step 2 : Create Virtual Network:**

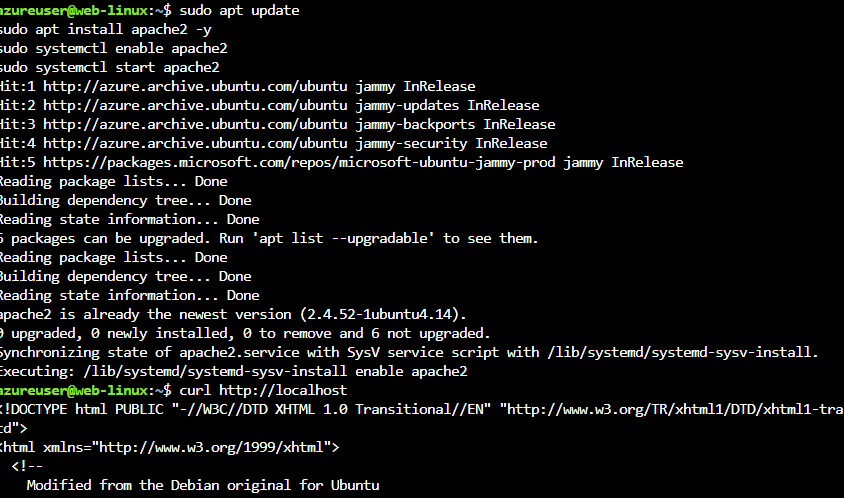
****

**Step 3: Creating Load Balancer:**

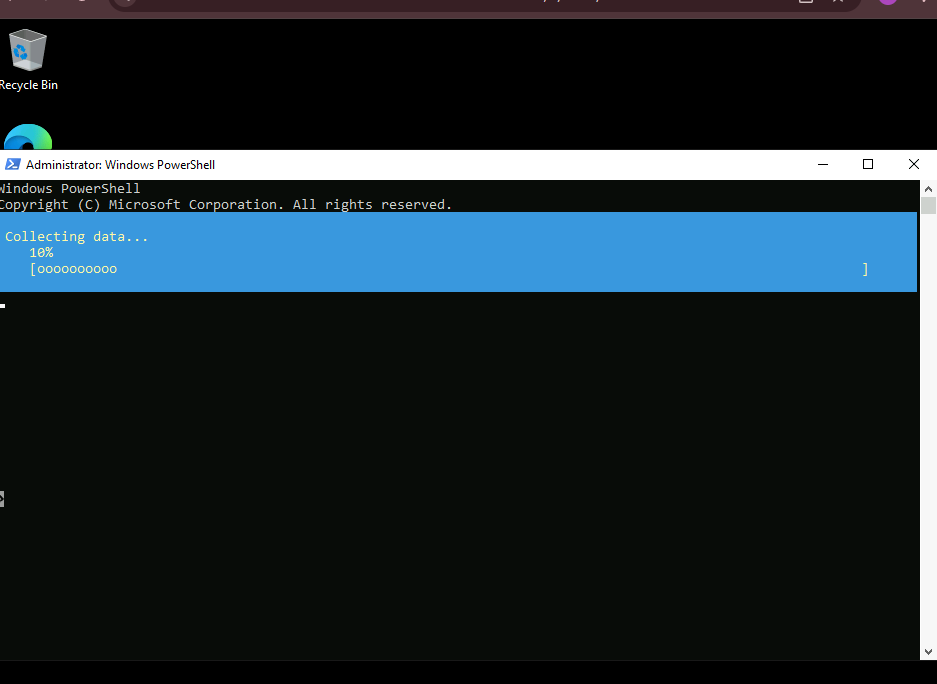
****

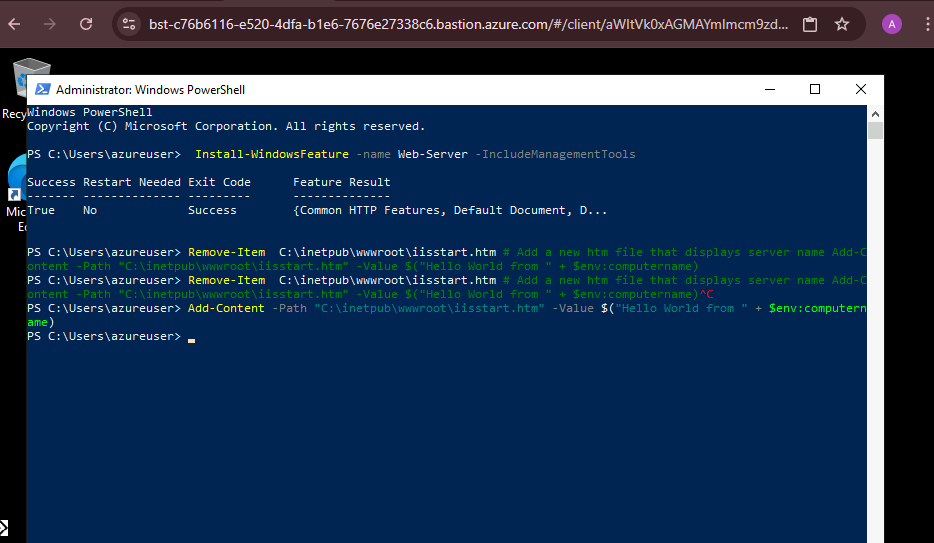
**Step 4 : Creating VMs (Two VMs and third one for Testing of Load Balancers):**

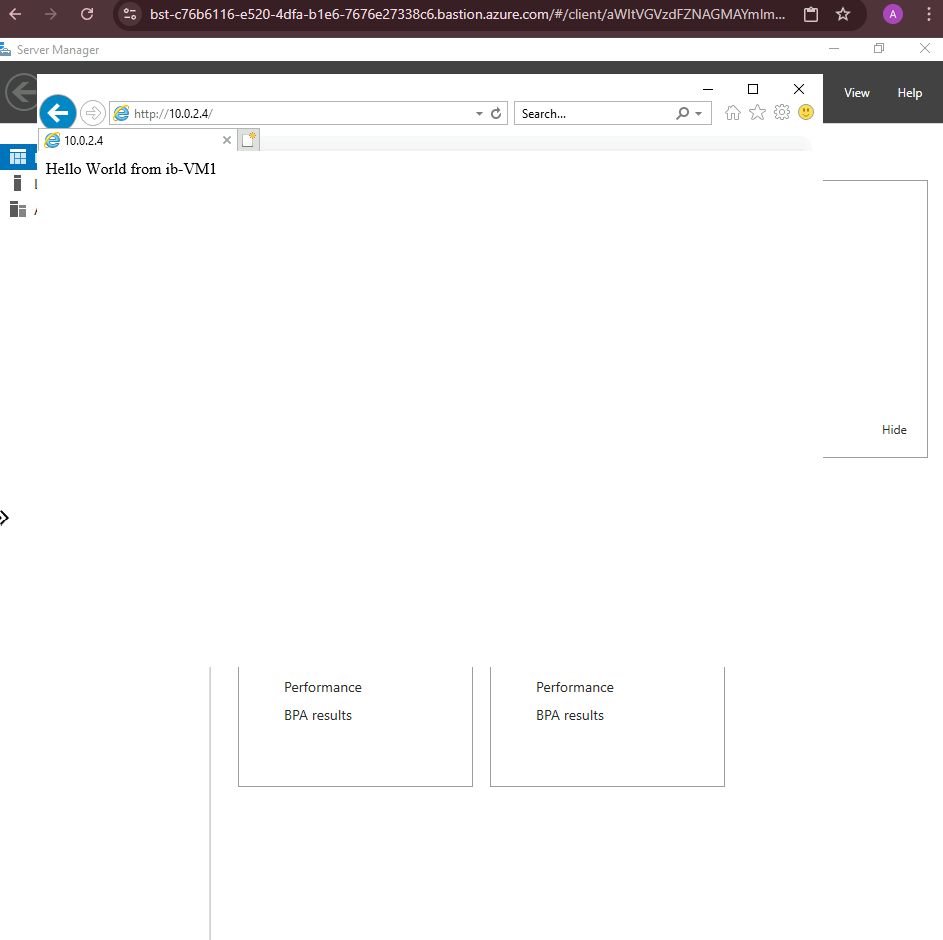
****

**Configuring VMs (Verifying Load Balancers working correctly):**

**Step 5: Installing IIS**

****



**Step 6: Testing Load Balancer:** 

**Both public & private load balancers are working properly.**

**References**

* **Microsoft Azure Documentation**
* **Cloudfare Documentation**
* **Microsoft Developer Blogs**
* **GeeksForGeeks**
* **K21 Academy**