**Agenda: Load Balancing Solutions**

* Understanding and CreatingAvailability Set
* Understanding Availability Zones
* Load Balancer
* Azure Traffic Manager

**Understanding and Creating Availability Sets**

There are two types of Microsoft Azure platform events that can affect the availability of your virtual machines:

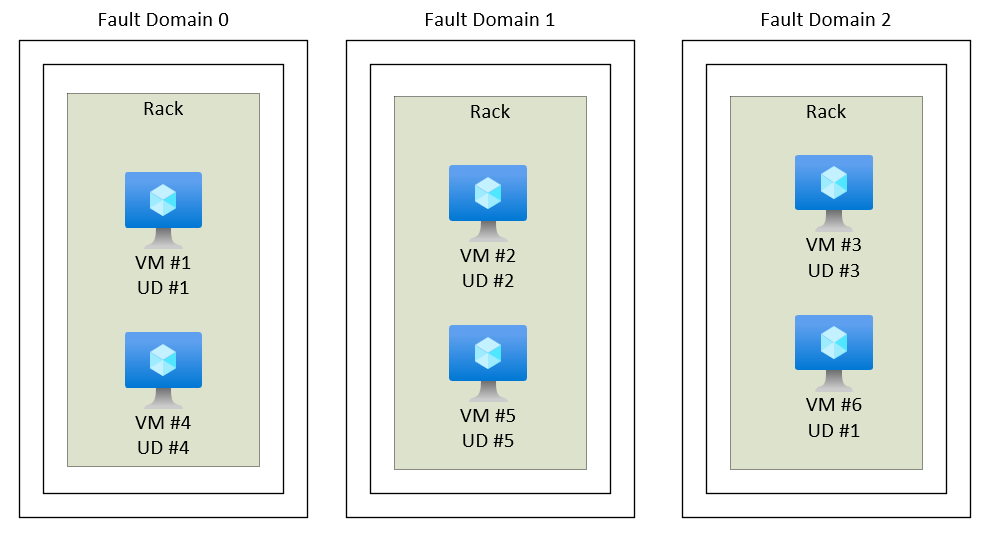
1. **Planned maintenance events** are periodic updates made by Microsoft to the underlying Azure platform to improve overall reliability, performance, and security of the platform infrastructure that your virtual machines run on.
2. **Unplanned maintenance events** occur when the hardware or physical infrastructure underlying your virtual machine has faulted in some way. This may include local network failures, local disk failures, or other rack level failures.

**Fault Domains:**

* A fault domain is a **physical point** of failure. Think of a computer (or a rack of servers) that is physically plugged in to a power outlet in one location (Unplanned update). If a power outage happens, that computer goes offline.
* When creating a new virtual machine instance, Azure will automatically place that instance in a new Fault Domain. This ensures that if you have 2 instances of a service, they cannot be in the same fault domain.

**Update Domains:**

* Whereas Fault Domains are a physical separation, Update Domains are a **logical separation**. Update domains exist so when Microsoft rolls out a new software feature or bug fix (Planned update), each update domain is upgraded at different times. This ensures that if you have at least 1 instances, your service will never go down as the result of an upgrade.
* Azure services can have up to 5 update domains by default (max of 20). When you create a new service instance, Azure automatically places it in the next update domain. If you have more than 5 instances, 7 for example, upgrade domains 0-1 will have 2 instances and upgrade domains 2-4 will have 1 instance.



Both FDs and UDs are assigned in the order that Azure discovers them as they are provisioned. So if you provision machines in the order Srv0, VM1, VM2, VM3, VM4, VM5, VM6, VM7, VM8, VM9, VM10, VM11 you’ll end up with a table that looks like this:

|  |  |  |
| --- | --- | --- |
| VM | **Fault Domain** | **Update Domain** |
| VM1 | 0 | 0 |
| VM2 | 1 | 1 |
| VM3 | 2 | 2 |
| VM4 | 0 | 3 |
| VM5 | 1 | 4 |
| VM6 | 2 | 0 |
| VM7 | 0 | 1 |
| VM8 | 1 | 2 |
| VM9 | 2 | 3 |
| VM10 | 0 | 4 |
| VM11 | 1 | 0 |
| VM12 | 2 | 1 |
| VM13 | 0 | 2 |

**Follow best practices when you design your application for high availability.**

1. Configure multiple virtual machines in an availability set for redundancy.
2. Configure each application tier into separate availability sets.
3. Combine a load balancer with availability sets.

**Important Notes:**

* It is critical to understand that it is **not possible to add an existing Azure virtual machine** to an availability set. You need to specify that a virtual machine will be part of an availability set when you provision it.
* There is also a limit of 200 virtual machines in each Azure availability set.
* Generally it is recommended to keep all VMs in an availability set the same size if you can.
* All running VMs in the availability set must be using the **same physical hardware cluster**.
  + If you resize one VM in you Availability Set to a different size to the rest of the set, but that size is available in the same hardware cluster then that is fine, nothing needs to be done.
  + Therefore, if a change of physical hardware cluster is required to change the VM size then all VMs must be first stopped and then restarted one-by-one to a different physical hardware clusters.

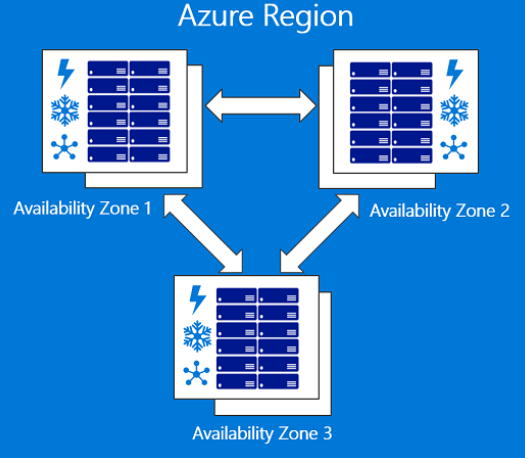
**Note: Because Availability Sets cannot span regions, if this region were to fail we would lose access to the whole application.**

**Understanding Availability Zones**

A map of the world

Description automatically generated

A diagram of a diagram of a structure

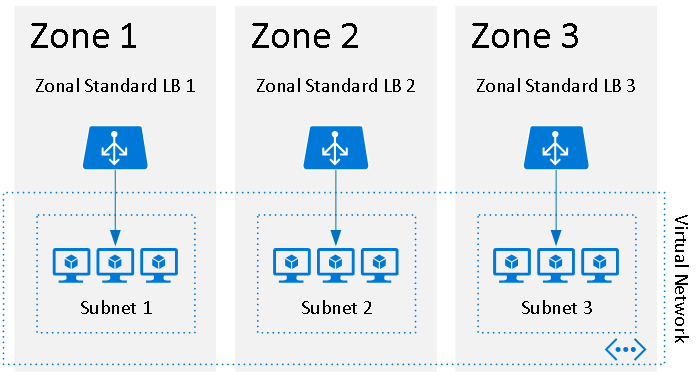
Description automatically generated with medium confidence

* Availability Zones are **unique physical locations** within an Azure region.
* Each zone is made up of one or more datacenters equipped with **independent power source, cooling, and networking**.
* To ensure resiliency, there’s a minimum of **three separate zones** in all **enabled regions**.
* The **physical separation** of Availability Zones within a region protects applications and data from **datacenter failures**.
* **Zone-redundant** services replicate your applications and data across Availability Zones to protect from **single-points-of-failure**.
* With Availability Zones, Azure offers industry best 99.99% VM uptime SLA.
* An **Availability Zone** in an Azure region is a **combination of a fault domain and an update domain**. So, if you’re deploying a web tier consisting of 2 VMs in Ireland, you can now make sure that VM1 is placed in Availability Zone 1 and VM2 is placed in Availability Zone 2. If zone 1 was to fail, you (and your customers) would still be able to access VM2 in AZ2.
* This means your service **won’t** have to run from a **separate Azure region** and will be faster as a result. This is especially useful if your customers are concentrated in a single region.
* Availability Zones are also ideal if you must obey **regulatory requirements and laws** that require your data/services to be highly available inside a single Azure Region.

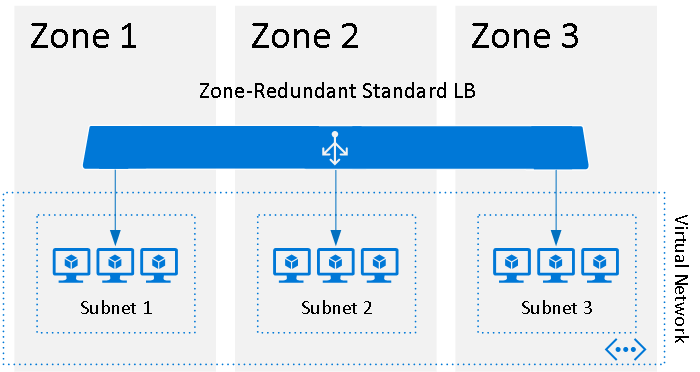
Build high-availability into your application architecture by co-locating your compute, storage, networking, and data resources within a zone and replicating in other zones.

**Azure services that support Availability Zones fall into two categories:**

1. **Zonal services** – you pin the resource to a specific zone (for example, virtual machines, managed disks, Standard SKU IP addresses, Load Balancer), You need to replicate your applications and data to one or more zones within the region so that you're resilient to a zone outage.



1. **Zone-redundant services (ZRS)** – platform replicates automatically across zones (for example, zone-redundant storage, SQL Database, Standard SKU IP addresses, Load Balancer).



## **Services that support Availability Zones**

* Linux Virtual Machines
* Windows Virtual Machines
* Virtual Machine Scale Sets
* Managed Disks
* Standard Load Balancer \*
* Standard public IP address \*
* Zone-redundant storage
* SQL Database
* Event Hubs
* Service Bus (Premium Tier Only)
* VPN Gateway
* ExpressRoute
* Application Gateway

**Bandwidth Pricing:**

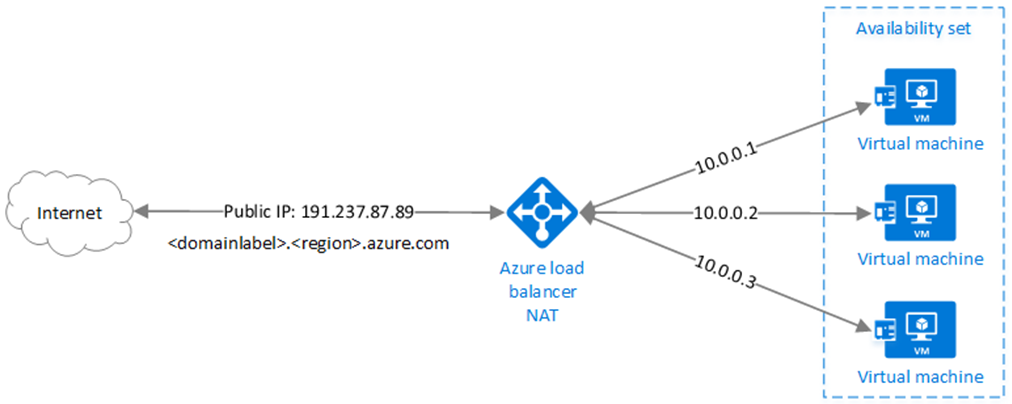
<https://azure.microsoft.com/en-us/pricing/details/bandwidth/>

[**https://infrastructuremap.microsoft.com/explore**](https://infrastructuremap.microsoft.com/explore)

[**https://docs.microsoft.com/en-us/azure/availability-zones/az-overview#regions-that-support-availability-zones**](https://docs.microsoft.com/en-us/azure/availability-zones/az-overview#regions-that-support-availability-zones)

**Azure Load Balancer**

The Azure Load Balancer delivers **high availability** and **network performance** to your applications. It is a **Layer 4** (TCP, UDP) load balancer that distributes incoming traffic among healthy service instances in virtual machines defined in a load-balanced set.

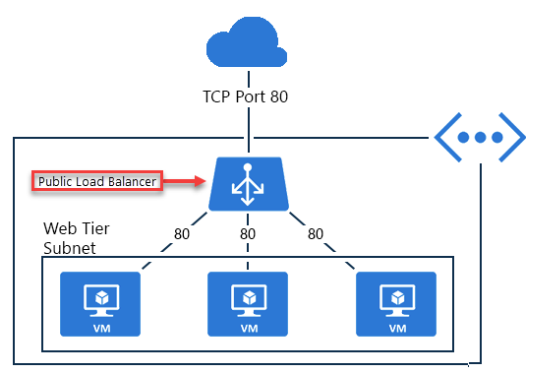
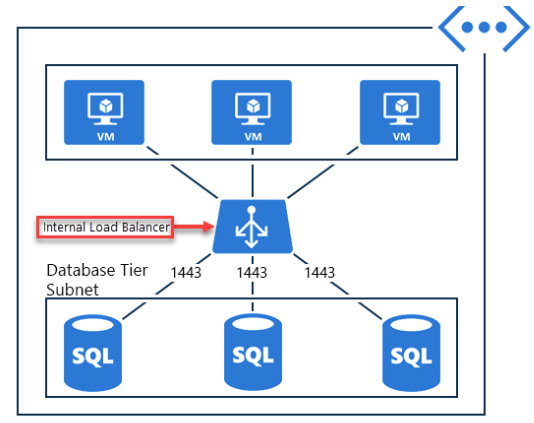


It can be configured to:

* Load balance incoming Internet traffic to virtual machines. This is called **Internet-facing** load balancing.
* Load balance between on-premises computers and virtual machines in a cross-premises virtual network. This is called **internal load balancing** (ILB).
* Forward external traffic to a specific virtual machine using NAT Rule.

All resources in the cloud need a public IP address to be reachable from the Internet. The cloud infrastructure in Microsoft Azure uses non-routable IP addresses for its resources. It uses network address translation (NAT) with public IP addresses to communicate to the Internet.

1. **Public load balancer**. You can use an external load balancer to provide high availability for IaaS VMs and PaaS role instances accessed **from the public Internet**.
2. **Internal load balancer**. You can use an internal load balancer to provide high availability for IaaS VMs and PaaS role instances accessed **from other services** in your Vnet.

**Hash-based distribution:**

Azure Load Balancer uses a **hash-based distribution algorithm**. By default, it uses a **5-tuple** hash composed of **source IP, source port, destination IP, destination port, and protocol type** to map traffic to available servers. It provides **stickiness** only within a transport session. Packets in the same **TCP or UDP** session will be directed to the same instance behind the load-balanced endpoint. When the client closes and reopens the connection or starts a new session from the same source IP, the source port changes. This may cause the traffic to go to a different endpoint.

**Basic SKU LB**

* Basic Load Balancer is free of charge.
* Any VM etiher having **Basic SKU Public IP** address or **not** at all having Public IP address.
* VMs in a **single availability set** or **VM scale set** can be added to Backend Pool.
* Upto 300 VM instances are allowed.
* NSG Rules at NIC or Subnet of the VM is honored but **not mandatory**.
* SLA is not available.
* Diagnostics through Azure Log Analytics for public-facing load balancers.
* Only HTTP and TCP Health Probes are supported.

**Standard SKU LB**

* Charged on the basis of Number of Rules and volume of Data processed.
* Any VM etiher having **Standard SKU Public IP** address or **not** at all having Public IP address.
* Any VM in a **single** virtual network, including a blend of VMs, availability sets, and VM scale sets. VM **may or may not** be in Availability Set.
* Upto 1000 VM instances are allowed.
* Its secured by default. This means Network Security Groups (NSGs) are used to explicitly permit and whitelist allowed traffic. **If you do not have an NSG on a subnet or NIC of your virtual machine resource, traffic is not allowed to reach this resource**.
* Supports Availability Zones. A guaranteed SLA (99.99% for two or more virtual machines in different zones)
* Diagnostics through Azure Log Analytics for public-facing load balancers. Diagnostics through **Azure Monitor**, for multidimensional metrics.
* HTTP, **HTTPS** and TCP health probes are supported.
* Supports Out-bound SNAT Rules

**Step1: Create Web1-vm**

1. Portal 🡪 Search 🡪 Virtual Machine 🡪 +Create 🡪 Azure Virtual Machine
2. Basic Tab
   1. Resource Group = Demo-rg
   2. Virual machine name = Web1-vm
   3. Region = EastUS
   4. Image = Windows Server 2022 Datacenter: Azure Edition – x64 Gen2
   5. Username = dssadmin
   6. Password = \*\*\*\*\*\*\*

A screenshot of a computer

Description automatically generated

1. Next: Disks >
2. Next: Networking >
   1. Virtual network = **Create new**
      1. Name = Demo-vnet
      2. OK

A screenshot of a computer

Description automatically generated

1. Review + create
2. RDP to VM and Install IIS Web Server 🡪 Edit image in c:\inetpub\wwwroot\iisstart.png (WRITE Digit 1 on Image)
3. Web1-vm 🡪 Overview 🡪 Note the Public IP Address
4. Browser 🡪 http://<Public-IP-Of-Web1-vm> 🡪 Error…Hmmm… can't reach this page
5. Figure out Problem and resolve.

**Step2: Create Web2-vm**

1. Portal 🡪 Search 🡪 Virtual Machine 🡪 +Create 🡪 Azure Virtual Machine
2. Basic Tab
   1. Resource Group = Demo-rg
   2. Virual machine name = **Web2-vm**
   3. Region = EastUS
   4. Image = Windows Server 2022 Datacenter: Azure Edition – x64 Gen2
   5. Username = dssadmin
   6. Password = \*\*\*\*\*\*\*

A screenshot of a computer

Description automatically generated

1. Next: Disks >
2. Next: Networking >
   1. Virtual network = **Demo-vnet** (Should be already existing)

A screenshot of a computer

Description automatically generated

1. Review + create
2. RDP to Web2-vm
   1. Install IIS 🡪 Edit image in c:\inetpub\wwwroot\iisstart.png (WRITE Digit 2 on Image)
   2. Browser 🡪 Visit <http://www.myipaddress.com> and note that you get the IP address of Web2-vm
3. Web2-vm 🡪 Overview 🡪 Note the Public IP Address
4. Browser 🡪 http://<Public-IP-Of-Web2-vm> 🡪 Error = Hmmm… can't reach this page
5. Figure out Problem and resolve

**Step3: Create Load Balancer**

1. Search 🡪 Load Balancer 🡪 +Create
2. Basics Tab:
   1. Name = Demo-loadBalancer
   2. Region = East US
   3. SKU = Standard
   4. Type = Public
   5. Tier = Regional
3. **Frontend IP Configuration**
   1. +Add a frontend IP Configuration
      1. Name = Demo-lb-ip
      2. Public IP address = Create new
         1. Name= **Demo-lb-ip**
         2. Availability zone = No Zone
         3. OK
      3. Add

A screenshot of a computer

Description automatically generated

1. **Backend pools Tab**
   1. Add Backend pool
      1. Name = Demo-pool
      2. Virtual network = Demo-vnet
         * IP configuration 🡪 +Add
         * Check Web1-vm and Web2-vm
         * Add

A screenshot of a computer

Description automatically generated

* + 1. Save

1. **Inbound rules**
   1. **Add a load balancing rule**
      1. Name = DemoRule
      2. Frontend IP address = Demo-lb-ip
      3. Backend pool = DemoPool
      4. Port = 80
      5. Backend port = 80
      6. Health probe = **Create new**
         1. Name = DemoHealthProbe
         2. Save
      7. Save

A screenshot of a computer

Description automatically generated

1. **Skip Outbound rules**
2. Review + Create 🡪 Create

**Step4: Testing Load Balancer**

1. Demo-loadBalancer 🡪 Frontend IP Configuration 🡪 Note the Public IP Address
2. Browser 🡪 http://<IP-of-LoadBalancer>
3. Continuously press Ctrl+F5 in browser and Note that the page is toggling between o/p of Web1-vm and Web2-vm

**STOP HERE…**

**Step5: To disable direct traffic to Web Server.**

Dissociate Public IP from the NIC of the Web1-vm and Web2-vm

1. Search **Public IP Address** 🡪 Select Web1-vm-ip

🡪 **Dissociate**

A screenshot of a computer

Description automatically generated

1. Repeat previous steps for Web2-vm-ip

Question: Will you be able to RDP to Web1-vm and Web2-vm?

A diagram of a diagram

Description automatically generated

**Step6: Add both IPs to Load Balancer**

1. Demo-loadBalancer 🡪 Frontend IP Configuration 🡪 Add
   1. Name = Web1-vm-ip
   2. Public IP adderess = **Web1-vm-ip**
   3. Add
2. Demo-loadBalancer 🡪 Frontend IP Configuration 🡪 Add
   1. Name = Web2-vm-ip
   2. Public IP adderess = **Web2-vm-ip**
   3. Add

**Step7: Create NAT Rules for RDP**

1. Repeat below steps for both Web1-vm and Web2-vm
2. Demo-loadBalancer 🡪 NAT Rules 🡪 + Add

A screenshot of a computer

Description automatically generated

**Step 8: Try RDP into both VMs with respective Load Balancer IP Address**

1. Windows Start 🡪 RDP 🡪 Enter Web1-vm-ip value (eg: 172.171.218.150)

A screenshot of a computer

Description automatically generated

**New Issue!!!!**

1. From the RDP Session of VM 🡪 Open browser 🡪 Visit <http://www.google.com> 🡪 Error…(Because VM doesn’t have Public IP Address)

**Step9: Add Load Balancer Outbound Rule**

1. Demo-loadBalancer 🡪 Outbound rules 🡪 + Add
2. Use the configuration as below

A screenshot of a computer

Description automatically generated

1. From the RDP Session of VM 🡪 Open browser 🡪 Visit <http://www.google.com> 🡪 Success this time
2. Visit <http://www.myipaddress.com> and note that you get the IP address of LoadBalancer.

**Step 10: Replace Outbound Rule with NAT Gateway**

1. Demo-loadBalancer 🡪 Outbound rules 🡪 Click on . . . for BrowsingInternet rule 🡪 Delete
2. From the RDP Session of VM 🡪 Open browser 🡪 Visit <http://www.google.com> 🡪 Error…(Because VM doesn’t have Public IP Address)

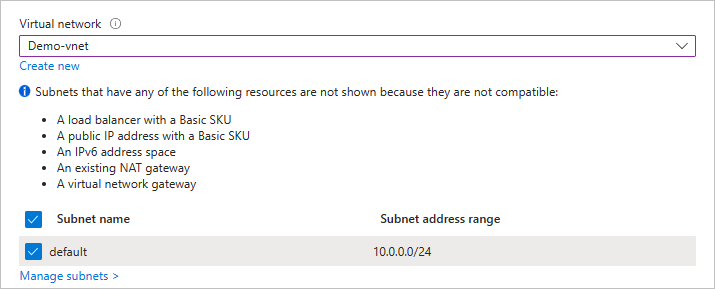
**Create NAT Gateway**

1. Portal 🡪 Search NAT Gateways 🡪 + Create
2. Basics Tab
   1. NAT gateway name = Demo-natGateway
   2. Region = EastUS
   3. Availability zone = No Zone
3. Outbound IP Tab
   1. Public IP addresses = Create a new public IP address
      1. Name = Demo-natGateway-ip
      2. OK
   2. Public IP Prefixes
      1. Name = Demo-natGateway-ip-prefix
      2. Prefix size = /31 (2 addresses)
      3. OK

A screenshot of a computer

Description automatically generated

1. Subnet Tab



1. Review + create

**Step 11: Testing NAT Gateway**

1. From the RDP Session of VM 🡪 Open browser 🡪 Visit <http://www.google.com> 🡪 Success this time
2. Visit <http://www.myipaddress.com> and note that you get the IP address as in NAT Gateway, Outbound IP (Public IP Prefixes).

<https://learn.microsoft.com/en-us/azure/load-balancer/quickstart-load-balancer-standard-public-portal>

**Azure Traffic Manager**

Microsoft Azure Traffic Manager allows you to control the distribution of user traffic for service endpoints in **different datacenters around the world.**

Service endpoints supported by Traffic Manager include Azure VMs, Web Apps, and cloud services. You can also use Traffic Manager with external, non-Azure endpoints.

Traffic Manager uses the **Domain Name System (DNS)** to direct client requests to the most appropriate endpoint based on a [**traffic-routing method**](https://docs.microsoft.com/en-us/azure/traffic-manager/traffic-manager-routing-methods)and the health of the endpoints.

It’s a Global Service and doesn’t have any location associated with it.

**Endpoints:**

1. Azure Endpoint
   1. ~~Cloud Service~~
   2. App Service
   3. App Service Slot
   4. Public IP address (Must have domain name)
2. External endpoint
   1. FQDN or IP

**There are four traffic routing methods available in Traffic Manager:**

1. **Priority:** Select 'Priority' when you want to use a primary service endpoint for all traffic, and provide backups in case the primary or the backup endpoints are unavailable. DR Solutions.

Timeline

Description automatically generated

1. **Performance:** Select 'Performance' when you have endpoints in different geographic locations and you want end users to use the **"closest**" endpoint in terms of the **lowest network latency**.

The closest endpoint is not necessarily measured by geographic distance. Instead Traffic Manager determines closeness by **measuring network latency**. Traffic Manager maintains an Internet Latency Table to track the round-trip time between IP address ranges and each Azure datacenter.

With this method Traffic Manager looks up the source IP address of the incoming DNS request in the Internet Latency Table. Traffic Manager chooses an available endpoint in the Azure datacenter that has the **lowest latency** for that IP address range, then returns that endpoint in the DNS response.

Timeline

Description automatically generated

1. **Geographic:** Select 'Geographic' so that users are directed to specific endpoints (Azure, External or Nested) based on **which geographic location** their **DNS query originates** from. This empowers Traffic Manager customers to enable scenarios where knowing a user’s geographic region and routing them based on that is important. Examples include **complying with** **data sovereignty** mandates, localization of content & user experience and measuring traffic from different regions.

Timeline

Description automatically generated

1. **Weighted:** Select 'Weighted' when you want to distribute traffic across a set of endpoints, either evenly or according to weights, which you define. The weight is an integer from 1 to 1000. The higher weight, the higher the priority. Use Canary Deployment (New versions)

Diagram

Description automatically generated

**Benefits of Traffic Manager**

1. Improve availability of critical applications.
2. Improve responsiveness for high performance applications.
3. Upgrade and perform service maintenance without downtime.
4. Combine on-premises and Cloud-based applications.
5. Distribute traffic for large, complex deployments.

**Step1:** Create Virtual Machine in **Southeast Asia** (Demo-asia-vm)

1. Portal 🡪 Search 🡪 Virtual Machine 🡪 +Create 🡪 Azure Virtual Machine
   1. Resource Group = Demo-asia-rg (new)
   2. Virual machine name = Demo-asia-vm
   3. Region = **Southeast Asia**
   4. Image = Windows Server 2022 Datacenter: Azure Edition – x64 Gen2
   5. Username = dssadmin
   6. Password = \*\*\*\*\*\*\*
   7. Select inbound ports = HTTP (80) + RDP (3389)
2. **Skip other tabs**
3. **Review + create 🡪 Create**
4. RDP to **Demo-asia-vm** and Install **IIS Web Server** 🡪 Edit image in c:\inetpub\wwwroot\iisstart.png (put text South ASIA)
5. Demo-asia-vm 🡪 Overview 🡪 Note the Public IP Address
6. Browser 🡪 http://<Public-IP-of-vm>

**Step2:** Create Virtual Machine in **East US** (Demo-eastus-vm)

1. Portal 🡪 Search 🡪 Virtual Machine 🡪 +Create 🡪 Azure Virtual Machine
   1. Resource Group = Demo-eastus-rg (new)
   2. Virual machine name = Demo-eastus-vm
   3. Region = **East US**
   4. Image = Windows Server 2022 Datacenter: Azure Edition – x64 Gen2
   5. Username = dssadmin
   6. Password = \*\*\*\*\*\*\*
   7. Select inbound ports = HTTP (80) + RDP (3389)
2. **Skip other tabs**
3. **Review + create 🡪 Create**
4. RDP to **Demo-eastus-vm** and Install **IIS Web Server** 🡪 Edit image in c:\inetpub\wwwroot\iisstart.png (put text East US)
5. Demo-eastus-vm 🡪 Overview 🡪 Note the Public IP Address
6. Browser 🡪 http://<Public-IP-of-vm>

**Step3: Assign Domain Name to IP Addresses**

1. Search 🡪 IP Addresses
2. Select Demo-eastus-vm-ip 🡪 Configuration
   1. DNS name label = **sandeep-demo-eastus-vm**
   2. Save

A screenshot of a computer

Description automatically generated

1. Select Demo-asia-vm-ip 🡪 Configuration
   1. DNS name label = **sandeep-demo-asia-vm**
   2. Save

## **To implement Traffic Manager**

1. Search 🡪 Traffic manager profiles 🡪 + Create
   * Name = sandeep-demo
   * Create

A screenshot of a computer

Description automatically generated

1. Search 🡪 Traffic manager profiles 🡪 **Sandeep-demo** 
   * **End Points 🡪 + Add**
     + **Name = Asia**
     + **Target resource type = Public IP address**
     + **Public IP address = Demo-asia-vm-ip**
     + A screenshot of a computer

       Description automatically generated
   * **End Points 🡪 + Add**
     + **Name = EastUS**
     + **Target resource type = Public IP address**
     + **Public IP address = Demo-eastus-vm-ip**
2. **Sandeep-demo 🡪 Overview 🡪 Ensure that Both the Endpoints are Online**
3. **In Browser Open:** <http://sandeep-demo.trafficmanager.net>

**Note: You will get output from the location which can deliver faster response as Routhing method=Performance**

**Routing method = Priority**

1. **Sandeep-demo 🡪 Configuration 🡪 Routing method = Priority 🡪 Save**
2. **Sandeep-demo 🡪 Endpoints 🡪 Select Asia 🡪 Priority=1 (High Priority)**
3. **Sandeep-demo 🡪 Endpoints 🡪 Select EastUS 🡪 Priority=2 (Low Priority)**
4. **In Browser Open:** <http://sandeep-demo.trafficmanager.net> **🡪 O/p will be delivered from Asia.**
5. **Similarly, try other routing methods.**