***Load balacnce backend :***

The \*\*Activity Log\*\* in Azure is a platform log that provides insights into operations and actions that occur within your Azure resources. It helps in monitoring, auditing, and troubleshooting issues by recording every change, action, or event.

### Key Features of Azure Activity Log:

1. \*\***Tracks Operations**:\*\* It logs all management operations on resources, such as creating, updating, or deleting resources.

2. \*\***Filters:**\*\* You can filter logs based on different parameters such as event category (e.g., administrative, service health, etc.), resource type, and date.

3. \*\***View Insights**:\*\* Helps in understanding what changes were made, by whom, and when.

4. \*\***Alerts**:\*\* You can set up alerts based on specific activity log events to notify you about critical changes or issues.

5. \*\***Integration**:\*\* Activity logs can be exported to services like Log Analytics, Event Hub, or a storage account for long-term retention and further analysis.

### Example Use Case:

- You deployed a new virtual machine, and you want to see who initiated the deployment. The activity log will record the user who performed this action, along with the time, details of the deployment, and its outcome.

**Loadblancing rule:**

A load balancing rule distributes incoming traffic that is sent to a selected IP address and port combination across a group of backend pool instances. Only backend instances that the health probe considers healthy receive new traffic.

Example Flow:

**User Sends Request:**

A user opens a web browser and goes to http://52.167.24.30:80 (Frontend IP with port 80).

Request Reaches Load Balancer:

The load balancer receives the request on port 80 and checks the load balancing rule BE-lb-lbrule01.

***Load Balancer Distributes Request:***

The rule specifies that the request should be forwarded to the backend pool on port 8000. It selects a healthy VM from the backend pool (e.g., BE-VM1), as confirmed by the health probe.

Backend VM Receives the Request:

The backend VM (BE-VM1) receives the request on port 8000, processes it (e.g., serves a webpage or runs an API), and sends a response back to the load balancer.

**Response Sent to User:**

The load balancer forwards the response back to the user.

**Session Persistence (if enabled):**

If session persistence is configured, the same user will continue to be routed to BE-VM1 for future requests, ensuring a consistent user experience.

**Idle Timeout and TCP Reset:**

If the user keeps the session idle for more than 5 minutes, the connection will be closed, and a TCP reset will be sent if configured.

**Health probe:**

A health probe in load balancing is a mechanism that checks the status of backend servers or virtual machines (VMs) to ensure they are healthy and capable of handling requests. The load balancer uses this information to decide whether it should continue routing traffic to a particular backend instance. If an instance is deemed unhealthy, it won't receive new traffic until it recovers.

**Flow Example:**

**User sends a request:**

A user sends a request to the load balancer's frontend IP (http://52.167.24.30:80).

**Load balancer checks backend health:**

Before routing the request, the load balancer checks the health of the backend VMs using the configured health probe (BE-lb-probe01). It checks port 8000 on all VMs in the backend pool.

**Routing decision:**

If a VM (e.g., BE-VM1) is healthy, the load balancer sends the request to it.

If a VM (e.g., BE-VM2) is unhealthy, the load balancer skips it and routes traffic to a healthy VM.

**Unhealthy VM recovery:**

If BE-VM2 becomes healthy again, the load balancer will start routing traffic to it once it passes the health checks.

Inbound NAT (Network Address Translation) rules in Azure are used to allow traffic from specific ports on a public IP address to be forwarded to specific ports on one or more virtual machines (VMs) in the backend pool of a load balancer. These rules enable you to connect to individual VMs in a load-balanced environment using protocols like Remote Desktop Protocol (RDP) or Secure Shell (SSH).

***Application Gateway:***

An Azure Application Gateway is a web traffic load balancer that enables you to manage traffic to your web applications. It works at the application layer (Layer 7) of the OSI model and provides advanced routing capabilities to ensure efficient distribution of incoming web traffic across multiple backend servers.

Azure Front Door and Content Delivery Network (CDN) profiles are services designed to improve the performance, scalability, and security of web applications by distributing content globally and managing traffic efficiently. Let's break down both:

Flow Example:

Let's say we have a scenario where a client sends a request to a website:

**Incoming Traffic:**

A client sends a request to a frontend IP address of the Application Gateway (e.g., 52.168.0.12) on port 80.

Listener:

The listener listens on port 80 and captures the request. The listener is configured to route traffic using HTTP.

**Routing Rule:**

The rule1 kicks in, which says that traffic from this listener should go to the backend pool FE-applicationgateway-backendpool01.

The rule is evaluated based on its priority (10010 in this case), and since no other rule has a higher priority, it is applied.

**Backend Target:**

The request is forwarded to FE-applicationgateway-backendpool01, which contains your backend servers (e.g., a virtual machine scale set).

**Backend Settings:**

The backend settings FE-applicationgateway-httpSetting01 define how the Application Gateway talks to the backend servers. In this example, it forwards traffic to port 80 on the backend servers, using the HTTP protocol.

If session persistence is enabled, the same client would continue to hit the same backend server for subsequent requests.

**Backend Pool:**

The backend servers in FE-applicationgateway-backendpool01 receive the requests, process them (e.g., serve the web page), and then respond to the client through the Application Gateway.

**Response:**

The backend pool sends the response back through the Application Gateway, which forwards it back to the client.

***Backend setting:-***

**Backend Protocol:**

HTTP or HTTPS:

This defines the protocol used by the Application Gateway to forward traffic to the frontend servers.

In your case, HTTP is selected, which means traffic will be forwarded to the backend using the HTTP protocol.

If you select HTTPS, it indicates that secure traffic (encrypted) will be forwarded between the Application Gateway and the backend servers.

**Backend Port:**

80:

The port on which the frontend servers (VM or VMSS) are listening. In this case, it's port 80, which is standard for HTTP traffic.

**Cookie-Based Affinity**:

Enable: connect to same target

Disable: different target

**Connection Draining:**

In case 50 usere are there it take 20 min it will go to different vm

**Request Timeout:**

15 seconds:

This is the time the Application Gateway waits for a backend server to respond before it times out.

If a backend server does not respond within 15 seconds, the request will fail, and the Application Gateway may try another healthy backend server if available.