**Question 1: You are setting up a web application in Azure and want to ensure high availability by distributing incoming traffic across multiple virtual machines (VMs). Describe how you would set up an Azure Load Balancer to achieve this.**

Answer:

Follow the following steps to set up web application in Azure which will provide high availability by distributing incoming traffic across multiple virtual machine with the help of load balancer:-

**Create a Resource Group:**

* Start by creating a resource group to logically group all the resources we'll be using for our web application.

**Create Virtual Network and Subnets:**

* Create a virtual network (VNet) and subnets for our VMs. We'll need at least one subnet for our VMs and one for the load balancer if we are using an internal load balancer.

**Create Availability Set or Availability Zones:**

* For high availability, place our VMs in an availability set or across availability zones (depending on the region and our requirements). This ensures that our VMs are distributed across multiple fault domains and update domains.

**Create Virtual Machines:**

* Deploy multiple VMs into the VNet and subnet we created, ensuring they are part of the availability set or zones. Install and configure our web application on each VM.

**Create a Public IP Address:**

* If we want our application to be accessible over the internet, create a public IP address that will be associated with the load balancer.

**Create the Load Balancer:**

* Create a new Load Balancer resource in Azure. Choose between a Basic Load Balancer (for simple scenarios) or a Standard Load Balancer (for more advanced features and high availability).
* **Name:** Provide a name for load balancer.
* **Type:** Select "Public" if we want our application to be accessible over the internet or "Internal" if we want it to be accessible only within our VNet.
* **Public IP Address:** Associate the public IP address we created earlier if we are using a public load balancer.

**Configure the Frontend IP Configuration:**

* Associate the frontend IP address with the load balancer. This is where the incoming traffic will arrive.

**Configure Backend Pools:**

* Create a backend address pool that will contain the IP addresses of the VMs that will handle the incoming traffic. Add the NIC IP configurations of VMs to this backend pool.

**Configure Health Probe:**

* Set up a health probe to monitor the health of the VMs. The probe dynamically adds or removes VMs from the load balancing rotation based on their responses to health checks.
* **Protocol:** HTTP, HTTPS, TCP, or other protocols depending on application.
* **Port:** The port used for the health check.
* **Interval:** How often the health check is performed.
* **Unhealthy threshold:** The number of failed checks before a VM is considered unhealthy.

**Configure Load Balancing Rules:**

* Define a load balancing rule that specifies how traffic is distributed to the VMs.
* **Frontend IP configuration:** Select the frontend IP address.
* **Backend pool:** Select the backend pool created.
* **Protocol and port:** Choose the protocol (TCP, UDP) and the frontend port.
* **Backend port:** The port on the backend pool instances that will receive the traffic.
* **Health probe:** Associate the health probe configured.
* **Session persistence:** Optionally configure session persistence if needed for application.

**Assign NSGs :**

If need to restrict network traffic to and from VMs, we can create and assign NSGs to subnets or individual NICs.

**Review and Create:**

Review load balancer configuration and create the load balancer.