**Research Document: Creating and Verifying Internal and External Load Balancers**

**Overview**

Load balancers distribute incoming network traffic across multiple servers to ensure no single server becomes overwhelmed. This document covers the steps to create and verify both internal and external load balancers.

**1. Load Balancer Types**

* **Internal Load Balancer (ILB)**: Balances traffic within a virtual network, not exposed to the internet.
* **External Load Balancer (ELB)**: Balances traffic coming from the internet.

**2. Preparation**

1. **Virtual Network (VNet)**: Ensure you have a VNet with the necessary subnets (Web, App, DB).
2. **Virtual Machines (VMs)**: Deploy VMs within the Web and App subnets to be used with the load balancers.

**3. Create Internal Load Balancer (ILB)**

1. **Navigate to Load Balancers**:
   * Go to the Azure portal, search for and select "Load balancers".
2. **Create Load Balancer**:
   * Click on "Create".
   * Choose the following settings:
     + **Subscription**: Your Azure subscription.
     + **Resource group**: Select an existing one or create a new one.
     + **Name**: InternalLoadBalancer.
     + **Region**: Same as your VNet.
     + **Type**: Internal.
     + **SKU**: Standard (recommended for production).
     + **Virtual Network**: Select your VNet.
     + **Subnet**: Choose the subnet for your internal load balancer (e.g., AppSubnet).
3. **Frontend IP Configuration**:
   * Add a frontend IP configuration:
     + **Name**: ILBFrontend.
     + **Private IP Address**: Static or dynamic, based on your requirement.
4. **Backend Pools**:
   * Add a backend pool:
     + **Name**: ILBBackendPool.
     + **Add VMs**: Select VMs from the App tier subnet.
5. **Health Probes**:
   * Create a health probe:
     + **Name**: ILBHealthProbe.
     + **Protocol**: HTTP.
     + **Port**: 80.
     + **Path**: /healthcheck.
6. **Load Balancing Rules**:
   * Create a load balancing rule:
     + **Name**: ILBRule.
     + **Frontend IP**: ILBFrontend.
     + **Backend Pool**: ILBBackendPool.
     + **Protocol**: TCP.
     + **Port**: 80.
     + **Backend Port**: 80.
     + **Health Probe**: ILBHealthProbe.

**4. Create External Load Balancer (ELB)**

1. **Navigate to Load Balancers**:
   * Go to the Azure portal, search for and select "Load balancers".
2. **Create Load Balancer**:
   * Click on "Create".
   * Choose the following settings:
     + **Subscription**: Your Azure subscription.
     + **Resource group**: Select an existing one or create a new one.
     + **Name**: ExternalLoadBalancer.
     + **Region**: Same as your VNet.
     + **Type**: Public.
     + **SKU**: Standard (recommended for production).
3. **Frontend IP Configuration**:
   * Add a frontend IP configuration:
     + **Name**: ELBFrontend.
     + **Public IP Address**: Create a new or use an existing public IP.
4. **Backend Pools**:
   * Add a backend pool:
     + **Name**: ELBBackendPool.
     + **Add VMs**: Select VMs from the Web tier subnet.
5. **Health Probes**:
   * Create a health probe:
     + **Name**: ELBHealthProbe.
     + **Protocol**: HTTP.
     + **Port**: 80.
     + **Path**: /healthcheck.
6. **Load Balancing Rules**:
   * Create a load balancing rule:
     + **Name**: ELBRule.
     + **Frontend IP**: ELBFrontend.
     + **Backend Pool**: ELBBackendPool.
     + **Protocol**: TCP.
     + **Port**: 80.
     + **Backend Port**: 80.
     + **Health Probe**: ELBHealthProbe.

**5. Verification**

1. **Verify Internal Load Balancer**:
   * SSH/RDP into a VM in the App tier subnet.
   * Test accessing the ILB frontend IP from within the VNet.
   * Ensure traffic is distributed across the backend VMs.
2. **Verify External Load Balancer**:
   * Access the public IP of the ELB from an external network.
   * Ensure traffic is distributed across the Web tier VMs.
   * Verify the backend VMs respond to requests through the ELB.

**6. Final Checks**

* **Health Probes**: Ensure health probes are correctly configured and backend VMs are healthy.
* **Logging and Monitoring**: Enable logging and monitoring for load balancers to track performance and troubleshoot issues.