Assignment 5

1. Set up a domain, setup a server on a VM and use the DNS server for traffic

   Resources :
   https://www.youtube.com/watch?v=57ZwdztCx2w&pp=ygUSYXp1cmUgcHJpdmF0ZSBsaW5rhttps://learn.microsoft.com/en-us/azure/virtual-network/virtual-networks-overview

2. Create and test Azure Application gateway

   Resources :
   https://www.youtube.com/watch?v=-SRk0hHa-S0https://learn.microsoft.com/en-us/azure/virtual-network/virtual-networks-overview

3. THIS SI THE DESCRIPTN OF THE ROW 23

   Resources :
   https://learn.microsoft.com/en-us/azure/virtual-network/virtual-networks-overview

To set up a domain, configure a server on a Virtual Machine (VM), and use a DNS server for traffic, you'll need to follow several steps involving domain registration, VM setup, and DNS configuration. Here's a detailed guide to help you through the process:

Step 1: Register a Domain Name

1. **Choose a Domain Registrar**:
   - Select a domain registrar to purchase and manage your domain name (e.g., GoDaddy, Namecheap, Google Domains).
2. **Search and Register a Domain**:
   - Search for an available domain name that you want to register.
   - Follow the registrar's instructions to complete the registration process.
   - Provide your contact information and make the necessary payment.
3. **Verify Ownership**:
   - Confirm your ownership of the domain as per the registrar's verification process (e.g., email verification).

Step 2: Set Up a Virtual Machine (VM)

1. **Create a Virtual Machine in Azure** (assuming Azure as the cloud provider):
   - Log in to the Azure portal.
   - Click on "Create a resource" > "Virtual Machine".
   - Configure the VM settings, including:
     - **Basics**: Choose subscription, resource group, VM name, region.
     - **Image**: Select an operating system image (e.g., Ubuntu Server, Windows Server).
     - **Size**: Choose an appropriate VM size.
     - **Settings**: Configure networking, disks, and other optional features.

        - **Review + create**: Review your settings and click "Create" to deploy the VM.
2. **Connect to the VM**:
    - Once the VM is deployed, connect to it using SSH (Linux) or RDP (Windows) from the Azure portal or your local machine.
3. **Set Up Server Software**:
    - Install and configure the necessary server software (e.g., Apache, Nginx for web servers; BIND, Microsoft DNS for DNS servers) based on your requirements.

Step 3: Configure DNS for the Domain

1. **DNS Configuration on Domain Registrar**:
    - Log in to your domain registrar's website where you registered your domain.
    - Navigate to the DNS management or DNS settings section.
2. **Create DNS Records**:
    - Add DNS records to point your domain to your server's IP address. Typically, you'll need at least:
        - An `A` record pointing `@` (root domain) or `www` to your server's public IP address.
        - Optionally, `CNAME` records for subdomains (e.g., `www`, `mail`, `ftp`) if needed.
3. **Name Server Configuration**:
    - Optionally, configure name servers (NS records) if you are using custom or external DNS servers.

Step 4: Verify and Test

1. **Verify DNS Propagation**:
    - After configuring DNS records, allow time for DNS propagation (typically a few minutes to a few hours).
    - Use online tools like `dig` command or DNS lookup tools to verify DNS resolution for your domain.
2. **Test Website Access**:
    - Open a web browser and navigate to your domain name (e.g., `http://yourdomain.com`).
    - Ensure your server's website or application is accessible.

Step 5: Manage and Maintain

1. **Monitor and Manage**:
    - Regularly monitor your server and domain for performance, security, and updates.
    - Set up monitoring and logging for your server and DNS configurations.
2. **Scale and Backup**:
    - Depending on your needs, scale your server resources or add redundancy.
    - Implement regular backups for your server and critical data.

Summary

By following these steps, you can successfully set up a domain, configure a server on a VM, and use DNS to manage traffic to your server. This setup ensures that your website or application is accessible via your domain name on the internet. Adjustments may be needed based on your specific requirements and the DNS registrar's interface you choose to use.

To create and test an Azure Application Gateway, which acts as a web traffic load balancer, follow these steps. The Application Gateway allows you to manage traffic to your web applications based on various criteria, including URL path, host headers, and more.

Step 1: Create a Resource Group

1. **Navigate to the Azure Portal**:
   - Open a web browser and go to the Azure portal.
2. **Create a Resource Group**:
   - Click on "Create a resource" in the left-hand menu.
   - Search for "Resource group" and select it.
   - Click on "Create".
   - Fill in the details:
     - **Subscription**: Select your Azure subscription.
     - **Resource group**: Enter a name for your resource group.
     - **Region**: Choose the appropriate region.
   - Click on "Review + create", then "Create".

Step 2: Create a Virtual Network (Optional)

If you haven't created a Virtual Network (VNet) yet, you can follow the steps outlined in the previous responses to create a VNet with subnets. Ensure that you have a subnet prepared where you want to deploy the Application Gateway.

Step 3: Create an Application Gateway

1. **Create the Application Gateway**:
   - Click on "Create a resource" in the left-hand menu.
   - Search for "Application Gateway" and select it.
   - Click on "Create".
2. **Configure the Application Gateway**:
   - **Basics**:
     - **Subscription**: Select your Azure subscription.
     - **Resource group**: Choose the resource group created in Step 1.
     - **Application Gateway name**: Enter a name for your Application Gateway.
     - **Region**: Choose the region where you want to deploy the Application Gateway.
     - **Tier**: Select the appropriate tier (Standard or WAF, depending on your requirements).

        - **Virtual network**: Select the VNet where you want to deploy the Application Gateway.
        - **Subnet**: Select the subnet within the VNet where the Application Gateway should be deployed.
        - **Frontends**: Configure the frontend IP and port (typically HTTP or HTTPS).
3. **Backend Pool**:
    - Configure the backend pool by adding VMs or IP addresses of the servers that will receive traffic from the Application Gateway.
4. **HTTP Settings and Routing Rules**:
    - Configure HTTP settings (such as cookie-based affinity, backend protocol, and port) and routing rules to define how traffic should be directed.
5. **Review and Create**:
    - Review your configuration settings.
    - Click on "Review + create".
    - After validation passes, click on "Create".

Step 4: Test the Application Gateway

1. **Access the Application Gateway**:
    - Once the Application Gateway is deployed successfully, note down its public IP address or DNS name (if configured).
    - Ensure that the Application Gateway's frontend configuration matches the DNS name you intend to use.
2. **Verify Backend Access**:
    - Open a web browser and navigate to the public IP address or DNS name of the Application Gateway.
    - You should see your web application or a response from your backend servers.
3. **Test Load Balancing and Routing**:
    - Test different URLs or paths defined in your routing rules to ensure traffic is directed correctly to different backend servers or pools.
4. **Monitor and Troubleshoot**:
    - Use Azure Monitor to monitor the performance and health of your Application Gateway.
    - Check logs and metrics to troubleshoot any issues related to traffic routing or backend server connectivity.

Step 5: Additional Configuration (Optional)

1. **SSL Offloading**:
    - Configure SSL termination at the Application Gateway to offload SSL/TLS processing from backend servers.
2. **WAF Configuration (if applicable)**:
    - If using the Web Application Firewall (WAF) tier, configure security rules and policies to protect your web applications from common threats.

Summary

By following these steps, you can create and test an Azure Application Gateway to effectively manage and distribute web traffic to your backend servers or web applications. Adjust

configurations based on your specific requirements for traffic management, security, and performance.