# Lab 2 [Backend application performance]

|  |  |
| --- | --- |
| Lab Title: | Backend application performance |
| **Expected duration** (hours): | **1 hour 45 mins** |

|  |
| --- |
| Objectives |
| Understanding load balancers and how they provisioned in a cloud environment |
| Learn how to provision cache like Redis in a cloud environment |

|  |
| --- |
| Requirements |
| A laptop or a desktop with Windows or Mac as an operating system |
| Visual studio code or something similar as a text editor |
| A modern web browser like chrome |

## Lab Instructions:

# Understanding load balancers, caching etc. and how they provisioned in a cloud environment

This lab will help you to get familiar with load balancers, how they are being utilised and what are the steps need to be followed to create a load balancer in the cloud. We will also look into benefits of cache, and how can easily provision a cache server in the cloud for a web application.

### **Part 1 – Create a load balancer in Azure**

### **Step 1: Log in to the Azure Portal**

1. Go to the [Azure Portal](https://portal.azure.com/) and log in with your Azure account.

### **Step 2: Create a Resource Group**

1. In the Azure Portal, click on "Resource groups" in the left-hand menu.
2. Click the "+ Create" button.
3. Fill in the necessary details:

* Subscription: Select your subscription.
* Resource group: Enter a name for your resource group.
* Region: Choose a region where you want to deploy your resources.

1. Click "Review + create" and then "Create".

### **Step 3: Create Virtual Network and Subnets**

1. In the Azure Portal, search for "Virtual networks" and select it.
2. Click the "+ Create" button.
3. Fill in the necessary details:

* Subscription: Select your subscription.
* Resource group: Select the resource group you created.
* Name: Enter a name for your virtual network.
* Region: Select the same region as your resource group.

1. Click on "IP Addresses" and configure your address space and subnets.

* Address space: Enter an IP address range (e.g., 10.0.0.0/16).
* Subnets: Add subnets (e.g., frontend-subnet, backend-subnet).

1. Click "Review + create" and then "Create".

### **Step 4: Create Virtual Machines**

1. In the Azure Portal, search for "Virtual machines" and select it.
2. Click the "+ Create" button.
3. Fill in the necessary details:

* Subscription: Select your subscription.
* Resource group: Select the resource group you created.
* Virtual machine name: Enter a name for your VM.
* Region: Select the same region.
* Availability options: Choose "Availability set" or "Availability zone" to ensure high availability.
* Image: Select the OS image (e.g., Windows Server or Ubuntu).
* Size: Choose the VM size.
* Administrator account: Configure the admin username and password/SSH key.
* Inbound port rules: Allow HTTP (80) and/or HTTPS (443).

1. Click "Next: Disks" and configure storage options.
2. Click "Next: Networking" and select your virtual network and subnet.
3. Repeat the process to create additional VMs as needed.
4. Click "Review + create" and then "Create".

### **Step 5: Create a Load Balancer**

1. In the Azure Portal, search for "Load balancers" and select it.
2. Click the "+ Create" button.
3. Fill in the necessary details:

* Subscription: Select your subscription.
* Resource group: Select the resource group you created.
* Name: Enter a name for your load balancer.
* Region: Select the same region.
* Type: Choose "Public" or "Internal" based on your requirement.
* SKU: Select the appropriate SKU (Basic or Standard).

1. Click "Review + create" and then "Create".

### **Step 6: Configure the Load Balancer**

1. After the load balancer is created, go to your load balancer in the Azure Portal.
2. Click on "Frontend IP configuration" and add a frontend IP.
3. Click on "Backend pools" and add a backend pool.

* Add your virtual machines to the backend pool.

1. Click on "Health probes" and add a health probe.

* Configure the health probe (e.g., TCP, HTTP, HTTPS) and the port.

1. Click on "Load balancing rules" and add a load balancing rule.

* Configure the rule (e.g., frontend IP, backend pool, protocol, port, session persistence).

### **Step 7: Test the Load Balancer**

1. Obtain the public IP address of the load balancer from the "Frontend IP configuration" section.
2. Open a web browser and enter the public IP address.
3. You should see the web application being served from the VMs behind the load balancer.

Congrats – you have managed to create a load balancer in a cloud environment like Azure to help route your application’s traffic so that your backend servers can be best utilised.

### **Part 2 – Provision a redis cache in Azure**

### **Step 1: Log in to the Azure Portal**

1. Go to the [Azure Portal](https://portal.azure.com/) and log in with your Azure account.

### **Step 2: Create a Resource Group**

1. In the Azure Portal, click on "Resource groups" in the left-hand menu.
2. Click the "+ Create" button.
3. Fill in the necessary details:
   * Subscription: Select your subscription.
   * Resource group: Enter a name for your resource group.
   * Region: Choose a region where you want to deploy your resources.
4. Click "Review + create" and then "Create".

### **Step 3: Create a Redis Cache**

1. In the Azure Portal, search for "Azure Cache for Redis" and select it.
2. Click the "+ Create" button.
3. Fill in the necessary details:
   * Subscription: Select your subscription.
   * Resource group: Select the resource group you created.
   * DNS name: Enter a globally unique name for your Redis cache (this will be used as part of the hostname).
   * Location: Select the region where you want to deploy your Redis cache.
   * Pricing tier: Select a pricing tier that suits your needs (e.g., Basic, Standard, Premium).
4. Click "Next: Networking" to configure network settings.
   * Virtual Network: Optionally, select a virtual network to deploy the cache in.
5. Click "Next: Advanced" to configure advanced settings.
   * Persistence: Enable or disable persistence (only available for Premium tier).
   * Clustering: Enable or disable clustering (only available for Premium tier).
6. Click "Next: Tags" to add tags (optional).
7. Click "Review + create" and then "Create".

### **Step 4: Access and Configure Redis Cache**

1. After the Redis cache is created, go to your Redis cache instance in the Azure Portal.
2. In the "Overview" section, note the Hostname and Access keys. These will be used to connect to your Redis cache.
3. To configure Redis settings, go to "Settings" > "Advanced settings". Here you can configure various settings such as:
   * Maxmemory policy
   * Data persistence
   * Redis version upgrade

### **Step 5: Connect to Redis Cache**

1. Use the connection information (Hostname and Access keys) to connect to your Redis cache from your application.
2. Example connection strings for various programming languages:

**Node.js**:

const redis = require("redis");  
const client = redis.createClient({  
 host: "your-redis-cache-name.redis.cache.windows.net",  
 port: 6379,  
 password: "your-access-key"  
});  
  
client.on("connect", function() {  
 console.log("Connected to Redis");  
});  
  
client.set("key", "value", redis.print);  
client.get("key", redis.print);

### **Step 6: Monitor and Manage Redis Cache**

1. In the Azure Portal, go to your Redis cache instance.
2. Use the "Monitoring" section to view metrics such as:
   * CPU usage
   * Memory usage
   * Network traffic
3. Use the "Scaling" section to scale your Redis cache up or down based on your needs.

### **Step 7: Secure Redis Cache**

1. Go to the "Access keys" section in your Redis cache instance to regenerate keys if needed.
2. Use the "Firewalls and virtual networks" section to restrict access to your Redis cache to specific IP ranges or virtual networks.

Congrats – you have managed to create a caching server in a cloud environment like Azure to help manage your application’s performance.