

Networking Lab 10

Standard Load Balancer

Author:

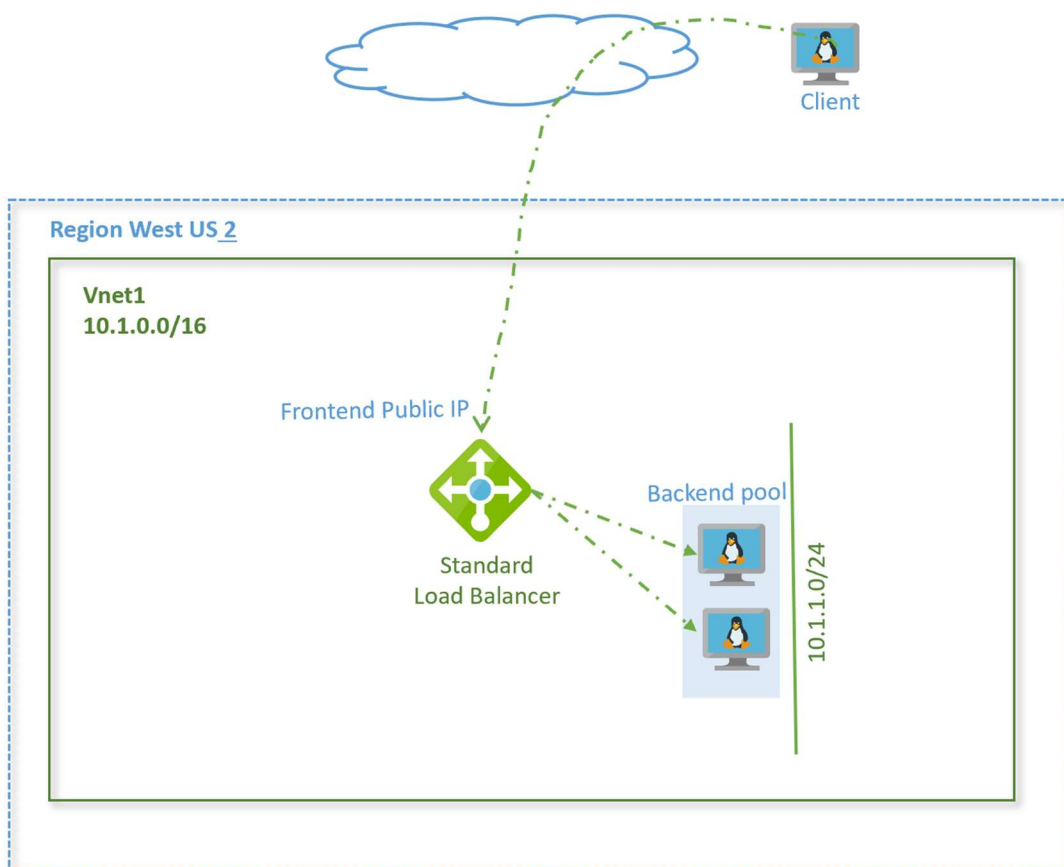
Binal Shah

Principal Cloud Solution Architect, Microsoft

Lab Overview

In this lab, we will see how to use a standard load balancer to load balance traffic to backend virtual machines.

Lab Diagram



Standard Load Balancer

A Standard load balancer load balances TCP and UDP traffic. We will spin up a load balancer in virtual network vnet1. Below is a summary of steps to create a load balancer

- Create virtual machines to load balance traffic to.
- Create a load balancer frontend IP.
- Create backend pool and add virtual machines to it.
- Create a health probe to check the health of the backend servers.
- Create a load balancing rule to listen on a port and load balance to backend pool.

Create backend servers

We need two virtual machines for this lab. We have already created vm vnet1-vm-web1 in lab 1. We will create another virtual machine vnet1-vm-web2.

1. Open command shell.
2. Define the following variables and run the command to create the virtual machine.

```
ResourceGroup=rg-lab
VmName=vnet1-vm-web2
VnetName=vnet1
SubnetName=vnet1-subnet2
Admin=azureuser
Password=Azure123456!
```

```
az vm create --resource-group $ResourceGroup --name $VmName --image UbuntuLTS --vnet-name $VnetName --subnet $SubnetName --nsg "" --asgs web --public-ip-address "" --admin-username $Admin --admin-password $Password
```

Configure the web servers

1. From cloud shell, ssh into the management vm vnet1-vm-mgmt1. You will need the public IP of the vm. You can get it from the 'Overview' page of the virtual machine.
2. From the management vm, ssh into the vm vnet1-vm-web1 using its private IP.
3. Once logged in, run the following commands:

```
sudo apt-get -y update
sudo apt-get -y install apache2
echo '<!doctype html><html><body><h1>Web Server 1</h1></body></html>' | tee
/var/www/html/index.html
sudo service apache2 status
```

Verify the output of the last command – it should show the server state as Active.

apache2.service - The Apache HTTP Server

Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)

Drop-In: /lib/systemd/system/apache2.service.d

└─apache2-systemd.conf

Active: active (running) since Sat 2019-10-26 02:03:57 UTC; 13min ago

Process: 27378 ExecStop=/usr/sbin/apachectl stop (code=exited, status=0/SUCCESS)

4. Log out from vm vnet1-vm-web1.
5. SSH to vm vnet1-vm-web2 and run the following commands:

```
sudo apt-get -y update
sudo apt-get -y install apache2
echo '<!doctype html><html><body><h1>Web Server 2</h1></body></html>' | tee
/var/www/html/index.html
sudo service apache2 status
```

Once again verify the web server status shows as **active(running)**.

6. Log out from vm vnet1-vm-web2.

Create a Standard Load Balancer

Create a Standard Load Balancer that helps load balance virtual machines.

1. From Azure portal, on the top search bar, type *Load Balancer* and select **Load Balancers** from the search results.
2. Click **+Add** on the Load Balancer page.
3. In the **Basics** tab of the **Create load balancer** page, enter or select the following information, accept the defaults for the remaining settings, and then select **Review + create**:

Setting	Value
Subscription	Select your subscription
Resource group	Select rg-lab
Name	<i>slb1-frontend1</i>
Region	Select US West 2
Type	Select Public
SKU	Select Standard

Public IP address	Select Create new
Public IP address name	Type <i>slb1-pip</i> in the text box.
Availability zone	Select Zone redundant .

- Once you see validation passed, click **Create**.

Create Load Balancer resources

Next, configure Load Balancer settings for a backend address pool, a health probe, and specify a rule.

Create a backend address pool

To distribute traffic to the VMs, create the backend address pool to include virtual machines for load-balancing internet traffic.

- From Azure portal, on the top search bar, type *slb1* and select **slb1-frontend1** from the search results.
- Under **Settings**, click **Backend pools**, then click **Add**.
- Name: *bepool1*
- Virtual Network: **vnet1**
- IP version: **IPv4**
- Under Virtual Machines, select virtual machine **vnet1-vm-web1**. Select IP address of the VM.
- Repeat step 4 to add virtual machine **vnet1-vm-web2** to the backend pool.
- Now, select **Add**.

Create a health probe

To allow the Load Balancer to monitor the status of your app, you use a health probe.

Create a health probe *health-http* to monitor the health of the VMs.

- Select **Settings** → **Health Probes**, then click **Add**.
- Use these values to create the health probe:

Setting	Value
Name	Enter <i>health-http</i>
Protocol	Select HTTP

Port	Enter <i>80</i>
------	-----------------

4. Leave the rest of the values default and select **OK**.

Create a Load Balancer rule

Create a Load Balancer rule *slb1-rule1* to listen on port 80 and send load-balanced network traffic to the backend pool *bepool1*.

1. Under **Settings**, click **Load balancing rules**, then click **Add**.
2. Use these values to configure the load-balancing rule:

Setting	Value
Name	Enter <i>slb1-rule1</i>
Protocol	Select TCP
Port	Enter <i>80</i>
Backend port	Enter <i>80</i>
Backend pool	Select <i>bepool1</i>
Health probe	Select <i>health-http</i>

4. Leave the rest of the defaults and select **OK**.

Test the Load Balancer

1. Find the public IP address for the Load Balancer on the **Overview** screen. From the Load Balancer page, click the frontend **slb1-frontend**.
2. Copy the public IP address, and then paste it into the address bar of your browser. The default page of the web server is displayed on the browser.
3. Shut down vm *vnet1-vm-web1*. You should see traffic now start going to the second vm *vnet1-vm-web2*.