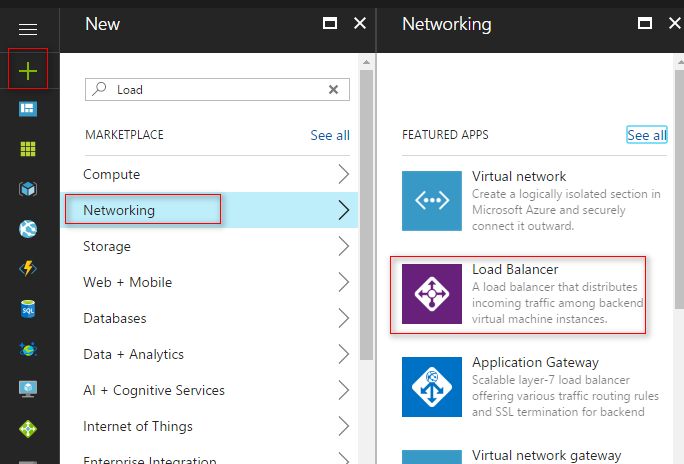
## Configuring the Load Balancer

### Lab Overview

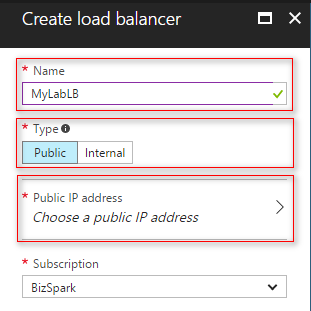
In this lab, you will create a new load balancer resource and a public static IP address. The web virtual machines created earlier will be added to the back end pool of the load balancer.

### Exercise 1: Create the Load Balancer

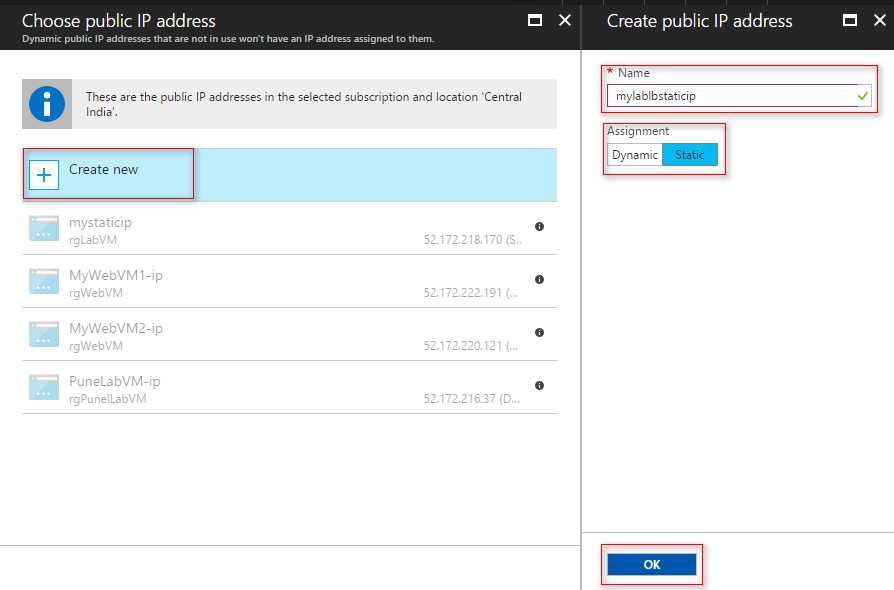
1. While logged into **LABVM**, open the Azure portal (https://portal.azure.com). Click **New** in the upper left hand corner. Go in to the Networking Category and select Load Balancer.



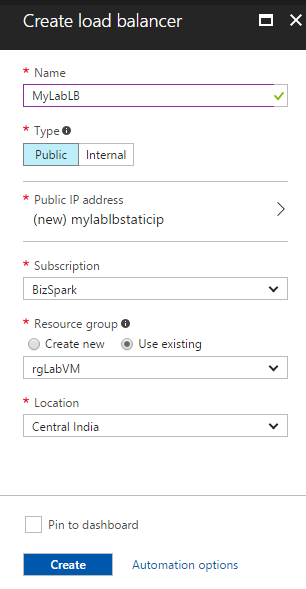
1. In the Create load balancer dialog, provide the following configuration detail,
   * Name: **MyLabLB**
   * Scheme**: Public**



* + Click on Public IP Address then click on CreateNew.
    - **Name**: mylablbstaticip
    - **Assignment**: Static
  + Click Ok Button

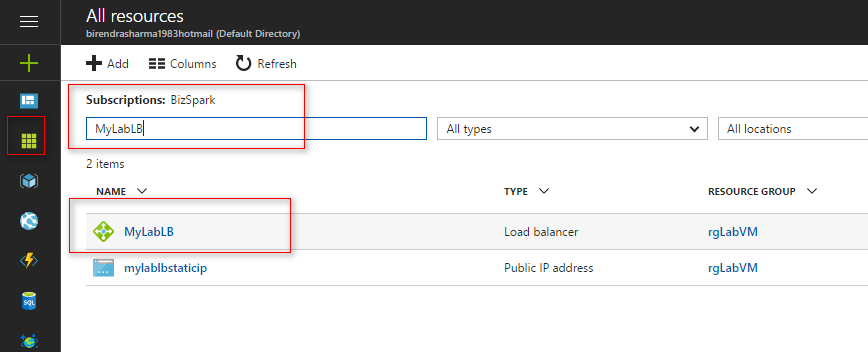


1. Ensure the correct subscription is selected.
2. Under Resource group click the dropdown list and choose **rgWebVM** from the listing of Resource Groups to choose from.
3. Ensure the **Location** is set to the same location used throughout this lab, then click **Create**.

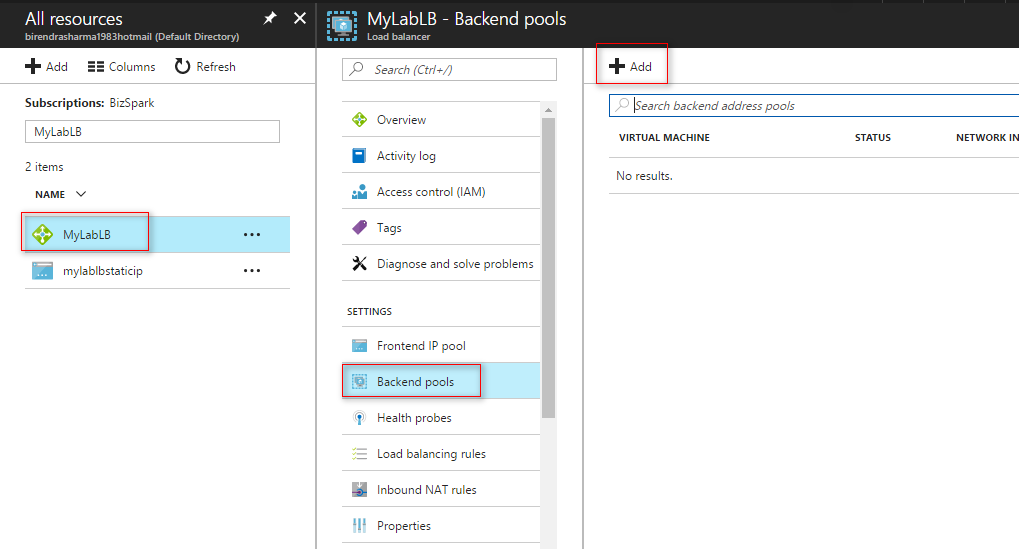


### Exercise 2: Configure the Load Balancer

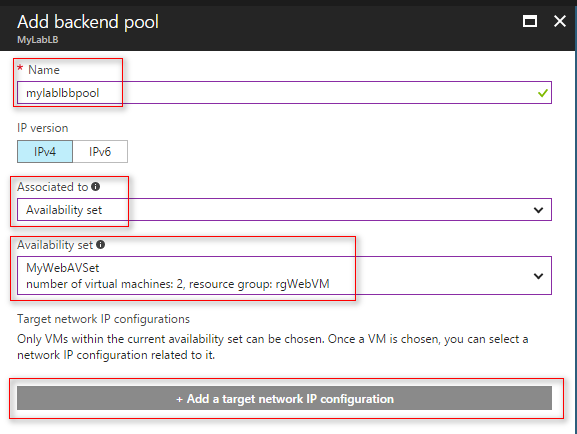
1. Goto All Resources and Search for Load Balancer by its name and click Load Balancer.



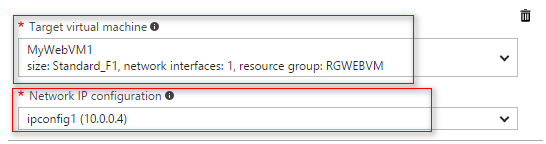
1. Under the Setting blade Click **Backend pools** and then click **Add** at the top.



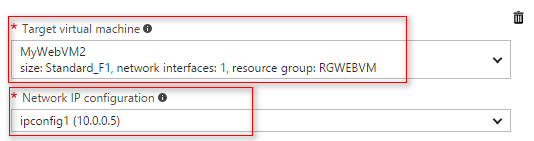
1. In the Add Backend Pool blade enter the following configuration,
   * Name: **mylablbbpool**
   * Associated to: Availability set
   * Availability set: Select Availability set that you have created during last vm.
   * Click on Add a target network IP configuration



* + Select first virtual machine

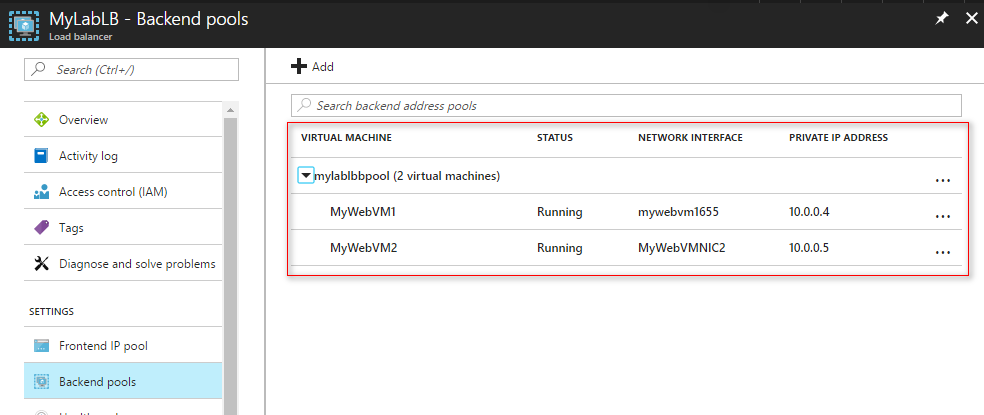


* + Click on Add a target network IP configuration again
  + Select second virtual machine

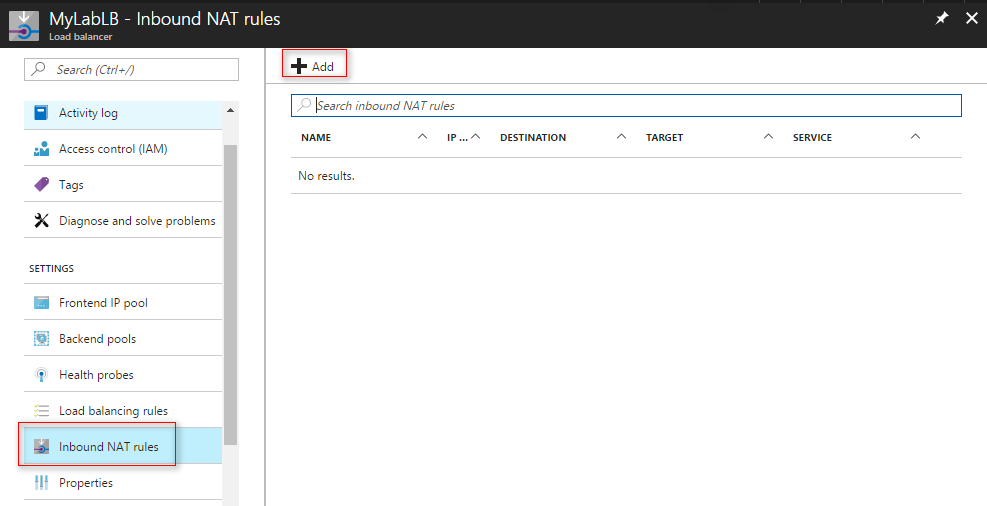


* + Click on Ok button

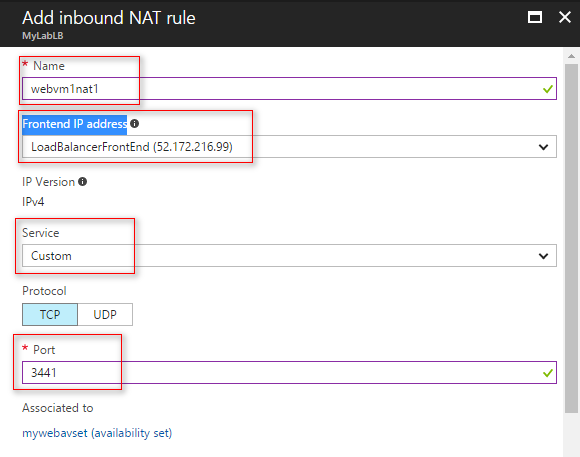
1. After successful backend pool you can see the list of vm in backend pool blade.

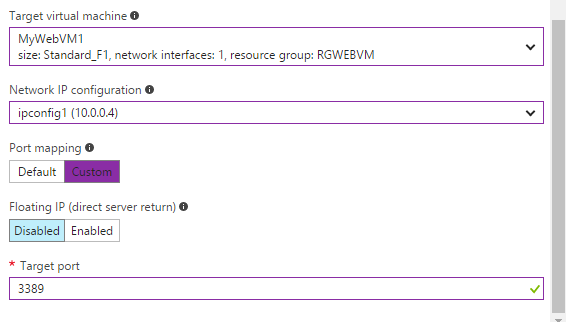


1. Back on the **Settings** blade forthe load balancer click **Inbound NAT** **rules**. On the **Inbound NAT rules** blade click **Add**.

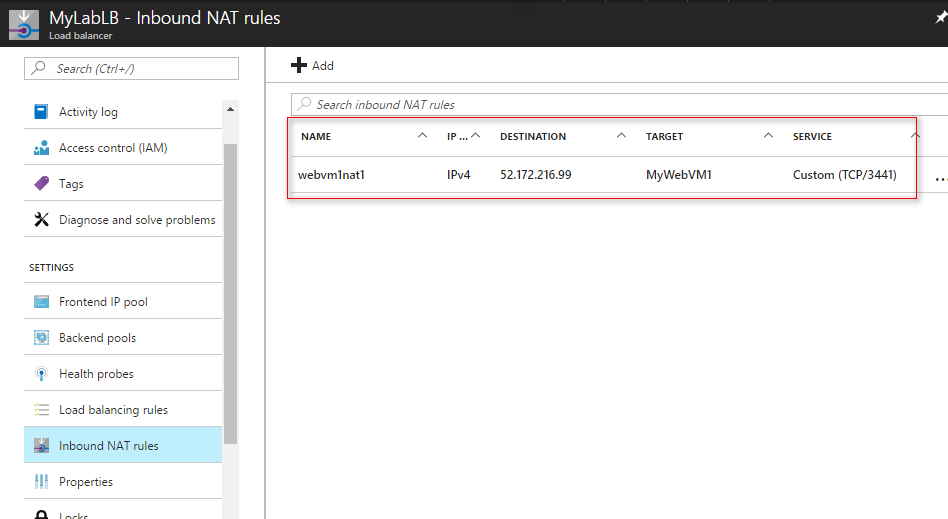


1. Configure the **Add inbound NAT rule** blade like below.
   * Name: webvm1nat1
   * Frontend IP address: LoadBalancerFrontEnd
   * Service: Custom
   * Port: 3441
   * Target virtual machine: select virtual machine1
   * Network IP configuration: select IP address of virtual machine
   * Port Mapping: Custom
   * Floating IP: Disable
   * Target Port: 3389

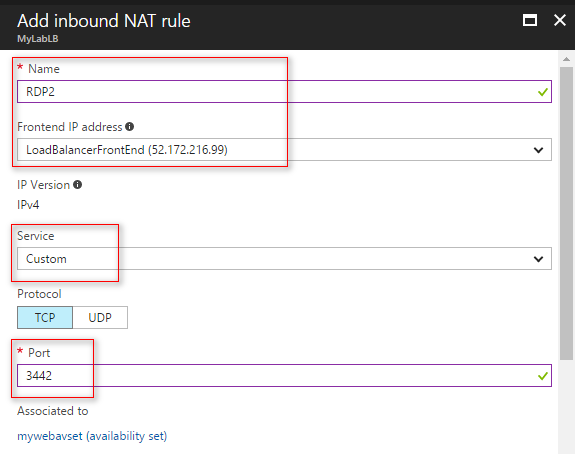


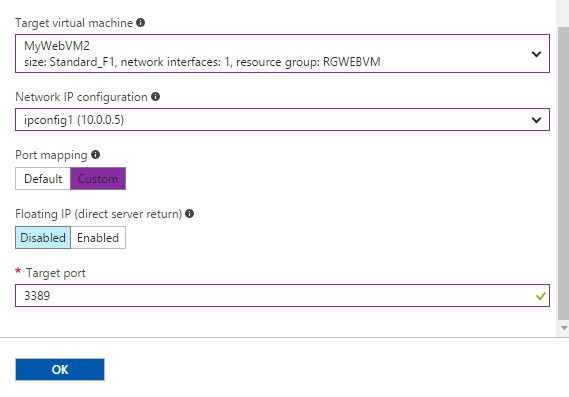


1. Click **OK** to create the rule.
2. After a couple of minutes, this operation will complete. Then proceed to the next step.

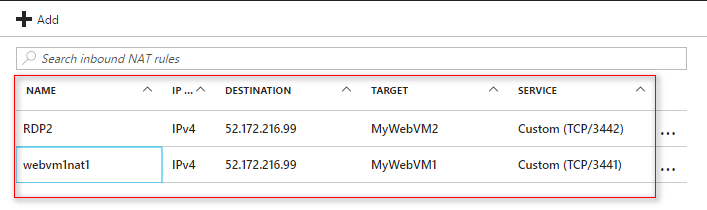


1. Click **Add** once more. Create another **inbound NAT rule** with the below configuration and click **OK** to save it.
   * Name: RDP2
   * Frontend IP address: LoadBalancerFrontEnd
   * Service: Custom
   * Port: 3442
   * Target virtual machine: select virtual machine2
   * Network IP configuration: select IP address of virtual machine
   * Port Mapping: Custom
   * Floating IP: Disable
   * Target Port: 3389

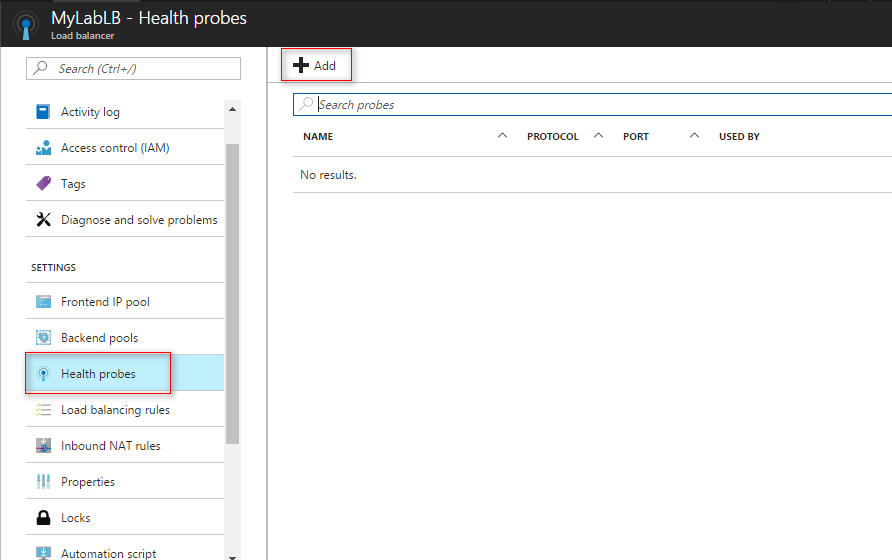




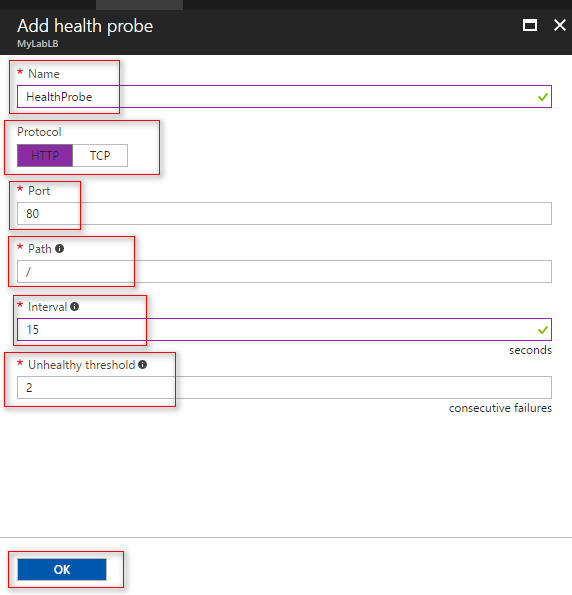
1. After completing NAT Rules you can see it on screen.

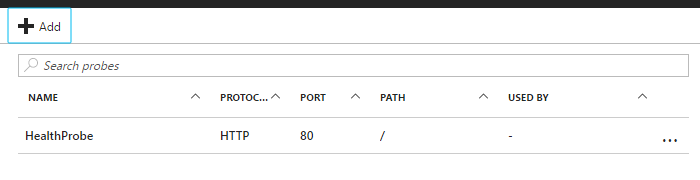


1. Next, in the **Settings** tab click on **Probes**. Click **Add.**

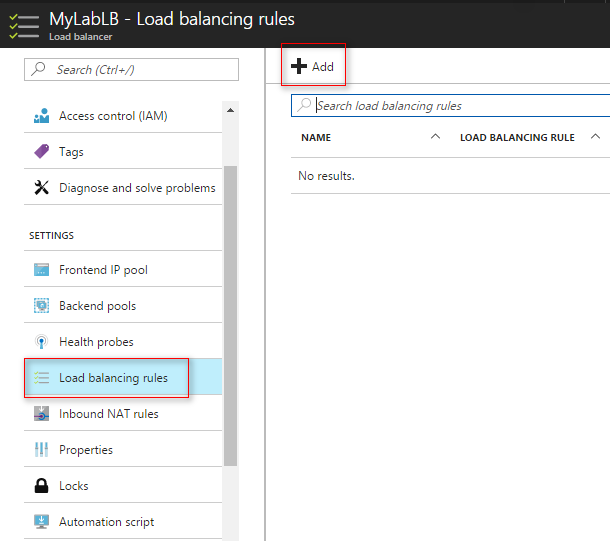


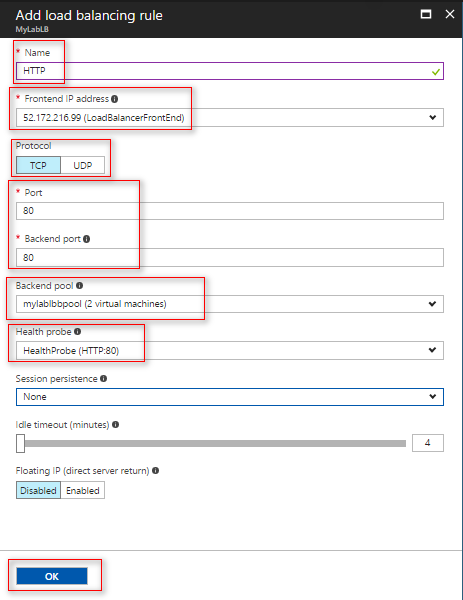
1. Use the below information to create a health probe, then click **OK** to create the probe.
   * Name: HealthProbe
   * Protocol: HTTP
   * Port: 80
   * Path: /
   * Interval: 15
   * Unhealthy Threshold: 2





1. Back on the **Settings** blade click **Load balancing rules**.
2. Click **Add** at the top of the **Load balancing rules** blade, complete the configuration as shown below, and then click **OK**.

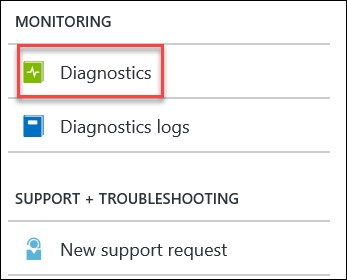




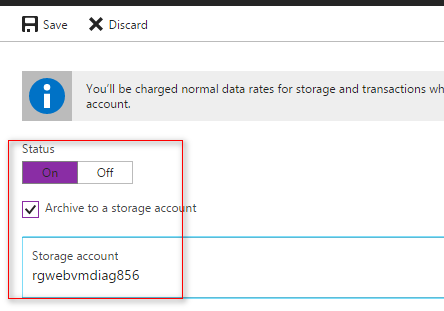
1. Close the **Load balancing rules** and **Settings** blade. The essentials panel shows you a high level view of how many virtual machines are in the backend pool, the public IP address associated with it as well as the number of NAT rules. Note the IP address as you will use it shortly to validate the load balancer.
2. Validate the load balancer is working by launching a browser instance and navigating to the public IP address of the load balancer. **Refresh** the page several times until you see a response from **WebVM-1 and WebVM-2**.

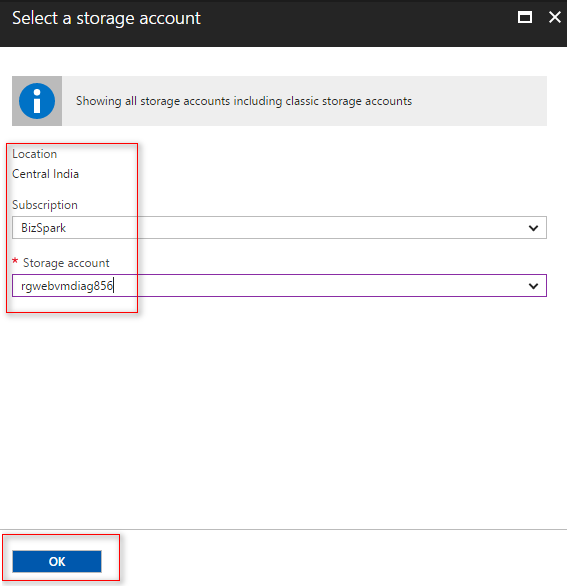
### Exercise 3: Configure Diagnostics for the Azure Load Balancer

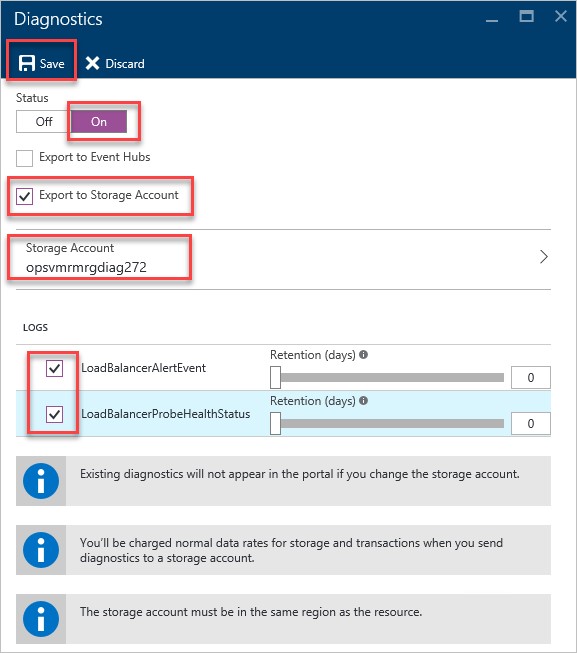
1. Open Load Balancer.
2. On the Settings Pane, click **Diagnostics**.

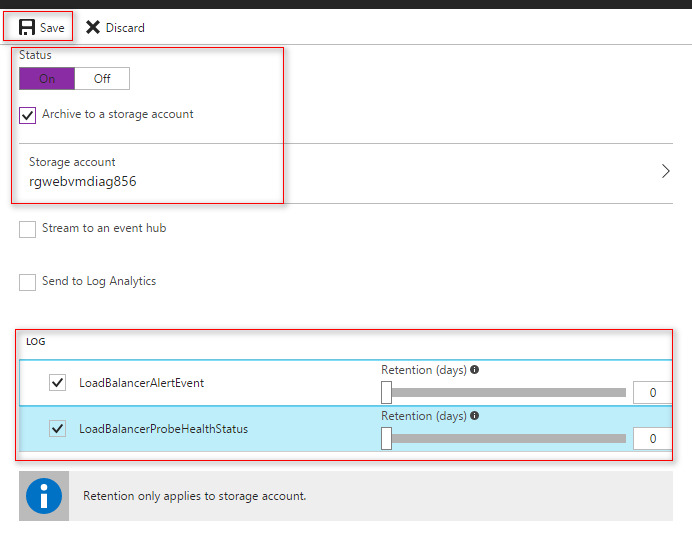


1. Enable Diagnostics on the load balancer by changing the Status to **On**, selecting a **Storage Account** (use the same one as your virtual machines for this lab), and then enable logging for the **LoadBalancerAlertEvent** and **LoadBalancerProbeHealthStatus** diagnostic sources. Click **Save** to continue.









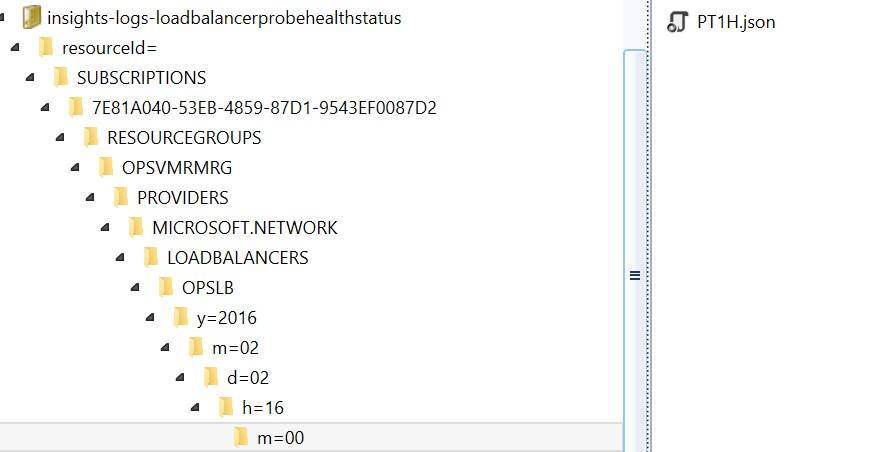
Note 1: Similar to virtual machine diagnostics, load balancer diagnostic data is store in Azure Storage Tables and is accessible using tools like Visual Studio or 3rd party management tools like Azure Management Studio. It may take up to an hour for any data to appear.

Note 2: At the time of this writing (2/2/2016) diagnostics logs are only applicable to load balancers listening on a public IP address.

1. Verify the load balancer alert works by stopping **WebVM-2**. Click **Browse**, **Virtual Machines**, **WebVM-2**, and click **Stop**.



1. Come back to this exercise in at least 1 hour so the diagnostics agent can synchronize the failure to Azure Storage. From Storage Explorer or Azure Management Studio navigate to the diagnostics storage account -> Blob Containers \ insights-logs-loadbalancerprobehealthstatus and expand the path down to the PT1H.json file.



The record will show that the “dipDownCount” is set to 1 (meaning that the private port is not responding to the load balancer health probes).

Lab

Summary



In this lab, you created a new load balancer resource and a public static IP address. The web virtual machines created earlier were added to the back end pool of the load balancer.