loadbalancer.org

Enterprise Azure Quick Start Guide

v8.3.0

Rev. 1.0.0

Table of Contents

1. Introduction

Azure is Microsoft's cloud platform. It's a growing collection of integrated cloud services that developers and IT professionals use to build, deploy and manage applications. It allows services to be deployed as and when required. Charges are made for what is used making it an extremely flexible and cost effective solution. The Loadbalancer.org Enterprise Azure cloud based load balancer allows customers to rapidly deploy and configure a feature rich load balancing solution within the Azure cloud.

2. About Enterprise Azure

The core software is based on customized versions of Centos 6.x/RHEL 6.x, Linux 3.10.x, LVS, HA-Linux, HAProxy, Pound, STunnel & Ldirectord. Enterprise Azure can be deployed as a single instance or as an HA clustered pair of instances for high availability and resilience. For details of adding a second (slave) instance, please refer to page 25. Enterprise Azure is based on the same code base as our main hardware/virtual product. This means that Enterprise Azure supports many of the same features as the hardware & virtual based products. There are certain differences due to the way the Microsoft Azure environment works. The main differences are listed below.

MAIN DIFFERENCES TO THE NON-CLOUD PRODUCT

- The appliance can only have a single IP address, all configured virtual services must also use this address
- The network setup is customized for Microsoft Azure deployment
- Layer 4 DR mode is currently not supported
- Layer 4 NAT mode where the default gateway on the load balanced real servers is required to be the load balancer is not supported. Routing rules for the real server subnet must be changed instead. Please refer to the example on page 21 for more details on configuring this
- Layer 7 SNAT mode with TProxy enabled where the default gateway on the load balanced real servers is required to be the load balancer is not supported

WHY USE ENTERPRISE AZURE?

Microsoft enables users to configure a load balancer to load balance multiple Azure instances running in the cloud. This does provide basic load balancing functionality but is limited in several areas. Loadbalancer.org's Enterprise Azure load balancer provides the following additional features & advantages:

1. Supports comprehensive Layer 7 load balancing functionality
2. Load balances both Azure based and non-Azure based servers
3. Supports Round Robin and Least Connection connection distribution algorithms
4. Supports customizable timeouts for custom applications beyond those offered by Azure
5. Supports comprehensive back-end server health-check options
6. Enables fallback servers to be configured and invoked when all load balanced servers/services fail
7. Provides extensive real time and historical statistics reports
8. Supports session distribution based on actual server load (utilizing Loadbalancer.org's feedback agent which is available for both Linux & Windows)
9. Supports SSL Termination
10. Supports RDP Cookie based persistence
11. Supports full integration with Remote Desktop Services Connection Broker

3. Azure Terminology

	Description
Azure	Microsoft's Cloud platform
ARM	Azure Resource Manager - the latest Azure deployment model
Classic	The original Azure deployment model
Virtual Network	Provides an isolated, private environment in the cloud
Availability Set	A collection of virtual machines that are managed together to provide application redundancy and reliability

4. Azure Deployment Models

The Azure platform is currently in transition from the older Classic or Service Management Model, to the new Resource Manager deployment model. Resource Manager is designed to eventually replace the classic deployment model, which has been the model up to now for deploying virtual machine-based workloads in Azure.

The Resource Manager deployment model provides a new way to deploy and manage the services that make up your application. This new model contains important differences from the classic deployment model, and the two models are not completely compatible with each other. To simplify the deployment and management of resources, Microsoft recommends that you use Resource Manager for new resources, and, if possible, re-deploy existing resources through Resource Manager.

The table below shows the main differences between the deployment models:

Item	Azure Classic	Azure Resource Manager
Cloud Service for Virtual Machines	Cloud Service was a container for holding the virtual machines that required Availability from the platform and Load Balancing.	Cloud Service is no longer an object required for creating a Virtual Machine using the new model.
Availability Sets	Availability to the platform was indicated by configuring the same “AvailabilitySetName” on the Virtual Machines. The maximum count of fault domains was 2.	Virtual Machines that require high availability must be included in the Availability Set. The maximum count of fault domains is now 3.
Load Balancing	Creation of a Cloud Service provides an implicit native Azure load balancer for the Virtual Machines deployed.	Native Azure Load balancers must be explicitly defined and can be internal or external.

Virtual IP Address (VIP)	Cloud Services will get a default VIP (Virtual IP Address) when a VM is added to a cloud service. The Virtual IP Address is the address associated with the implicit Azure load balancer.	Public IP Address can be Static (Reserved) or Dynamic. Dynamic Public IPs can be assigned to a Load Balancer. Public IPs can be secured using Security Groups.
Reserved IP Address	You can reserve an IP Address in Azure and associate it with a Cloud Service to ensure that the IP Address is sticky.	Public IP Address can be created in "Static" mode and it offers the same capability as a "Reserved IP Address". Static Public IPs can only be assigned to a Load balancer right now.
Public IP (PIP) per VM	Public IP Addresses can also associated to a VM directly.	Public IP Address can be Static (Reserved) or Dynamic. However, only dynamic Public IPs can be assigned to a Network Interface to get a Public IP per VM right now.
Endpoints	Input Endpoints needed to be configured on a Virtual Machine to open up connectivity for certain ports. One of the common modes of connecting to virtual machines done by setting up input endpoints.	The concept of Endpoints no longer exists. Instead create a network security group. A network security group is a set of firewall rules that control traffic to and from your virtual machine.
DNS Name	A cloud service would get an implicit globally unique DNS Name. For example: mycoffeeshop.cloudapp.net	DNS Names are optional parameters that can be specified on a Public IP Address resource. The FQDN will be in the following format: <label>.<region>.cloudapp.azure.com
Network Interfaces	Primary and Secondary Network Interface and its properties were defined as network configuration of a Virtual machine.	Network Interfaces are separately defined and associated with a VM. The lifecycle of the Network Interface is not tied to a Virtual Machine.

For a more detailed comparison of Classic and Resource Manager models, please refer to this URL.

5. Accessing Microsoft Azure

To start using Microsoft Azure, you'll need an Azure account. If you don't already have one you can create one at the following URL: https://account.windowsazure.com/

6. Azure Management

Many management tasks can be carried out using the Portal. However, there are still a number of tasks that must be done via Azure PowerShell or the CLI. Azure PowerShell is a powerful scripting environment that can be used to control and automate the deployment and management of workloads in Azure. The Azure CLI provides a set of open source, cross-platform commands for working with the Azure Platform.

ACCESSING THE AZURE PORTAL

The original and new Azure portals are available at the following URL's:

- Accessing the new portal: https://portal.azure.com
- Accessing the old portal: https://manage.windowsazure.com

AZURE POWERSHELL & AZURE CLI

- How to obtain, install and configure Powershell: https://docs.microsoft.com/en-gb/powershell/azure/install-azurerm-ps?view=azurermps-4.2.0

- How to obtain, install and configure Azure CLI: https://azure.microsoft.com/en-gb/documentation/articles/xplat-cli/

> Note:
>
> The load balancer has a single public IP address in Azure so all work-load and management services are accessed via the same IP address. Make sure that when you add Virtual Services (VIPs) , you either specify the appliances own IP address or 0.0.0.0. Please refer to the examples starting on page 16 for more details.

7. Deploying Enterprise Azure From the Marketplace

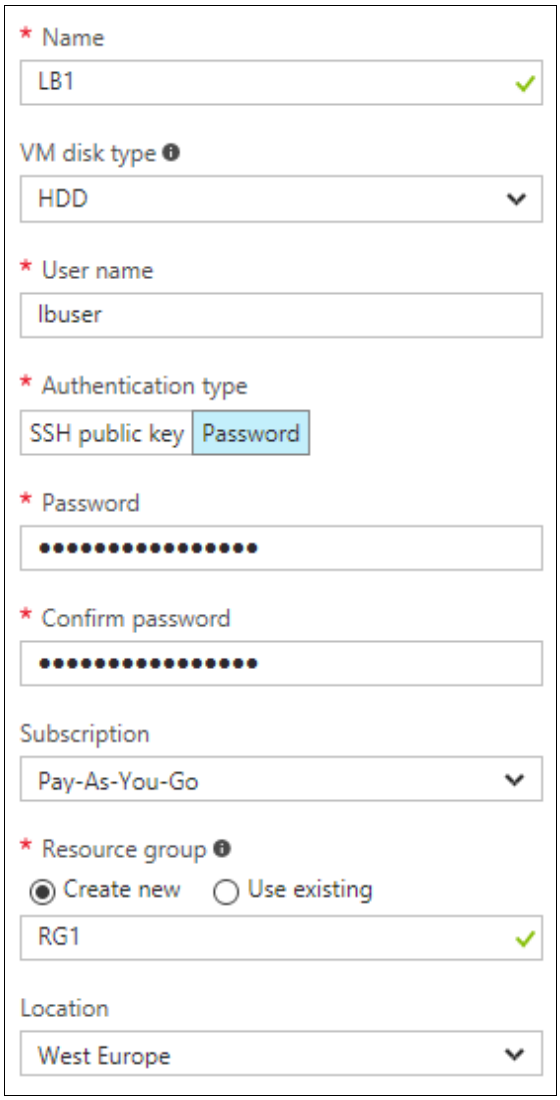
(Using Resource Manager Model)

1. Login to the Azure Management Portal: https://portal.azure.com
2. In the Azure Management Portal select the **Virtual Machines** option
3. Click **Add** under the Virtual Machines heading
4. In the *Search Compute* search box type **Loadbalancer.org** and press <ENTER>
5. Select either:

   - Loadbalancer.org Load Balancer for Azure - *for hourly billing*
   - Loadbalancer.org Load Balancer for Azure BYOL - *for purchasing & applying your own license*

6. Set the deployment model to **Resource Manager**
7. Click **Create**

Define basic settings:



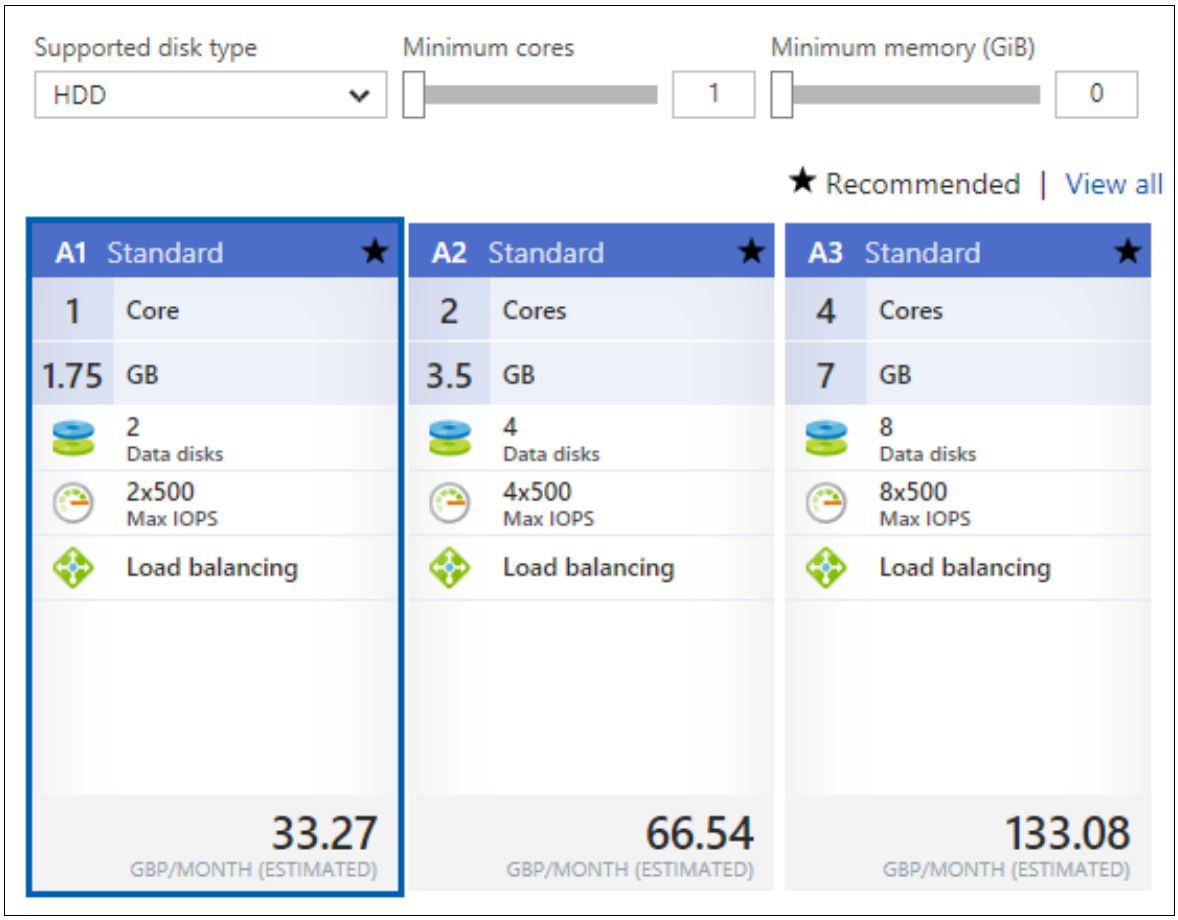
1. Enter a suitable name for the instance, e.g. **LB1**
2. Select the required disk type, either **SSD** or **HDD**
3. Enter a suitable username in the *User name* field, e.g. **lbuser**
4. Select the required *Authentication type* - a password or an SSH public key can be used

> Note:
>
> Please refer to page 14 for more details on creating and using SSH keys.

5. Enter a suitable password in the *Password* field (if using password authentication)
6. Set the *Subscription*, *Resource group* and *Location* according to your requirements
7. Click **OK**

Select an instance size:

1. Choose an appropriate instance size according to your requirements:



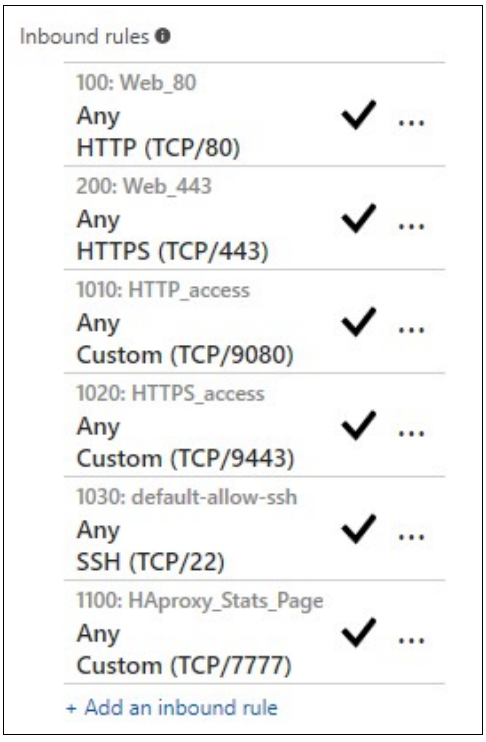
2. Click **Select**

Configure Optional Features:

1. Configure **Storage** settings according to your requirements
2. Configure **Network** settings according to your requirements

- A dynamic public IP address will assigned by default, the **Public IP address** option can be used to change this setting to suit your needs
- The **Network Security Group (Firewall)** settings that are assigned by default are shown below:

- The default inbound rules shown above are required for managing the load balancer. If you'll be deploying layer 7 services, TCP port 7777 can also be added – this allows the HAProxy statistics page to be viewed.
- Specify additional inbound rules for the ports used for your load balanced applications, e.g. TCP 80 and TCP 443 if you're load balancing web servers, TCP 3389 if you're load balancing RDP etc.
- To specify additional inbound rules, click **Add an inbound rule**. The example below shows additional ports TCP/7777 (for HAProxy stats), TCP/80 (for load balanced HTTP web server traffic) and TCP/443 (for load balanced HTTPS web server traffic) :



- Network Security Groups & Inbound/outbound rules can be assigned at an instance level and at the subnet level. Whichever way you choose to allocate your rules, make sure that the inbound rules required to manage the load balancer are included

3. Configure **Extensions** settings according to your requirements
4. Configure **High Availability** settings according to your requirements

- If you're configuring an HA pair, ensure that you select/create an Availability set
- For an HA pair, both appliances must be in the same Availability set

> Note:
> Please refer to page 25 for more details on setting up an HA pair.

5. Configure **Monitoring** settings according to your requirements
6. Click **OK**

Check all settings and accept the purchasing terms:

1. Check the Summary details are correct , e.g.

Basics	
Subscription	Pay-As-You-Go
Resource group	(new) RG1
Location	West Europe
Settings	
Computer name	LB1
Disk type	HDD
User name	lbuser
Size	Standard A1
Storage account	(new) rg1disks940
Virtual network	(new) RG1-vnet
Subnet	(new) default (10.0.0.0/24)
Public IP address	(new) LB1-ip
Network security group (firewall)	(new) LB1-nsg
Availability set	None
Guest OS diagnostics	Disabled
Boot Diagnostics	Enabled
Diagnostics storage account	(new) rg1diag424

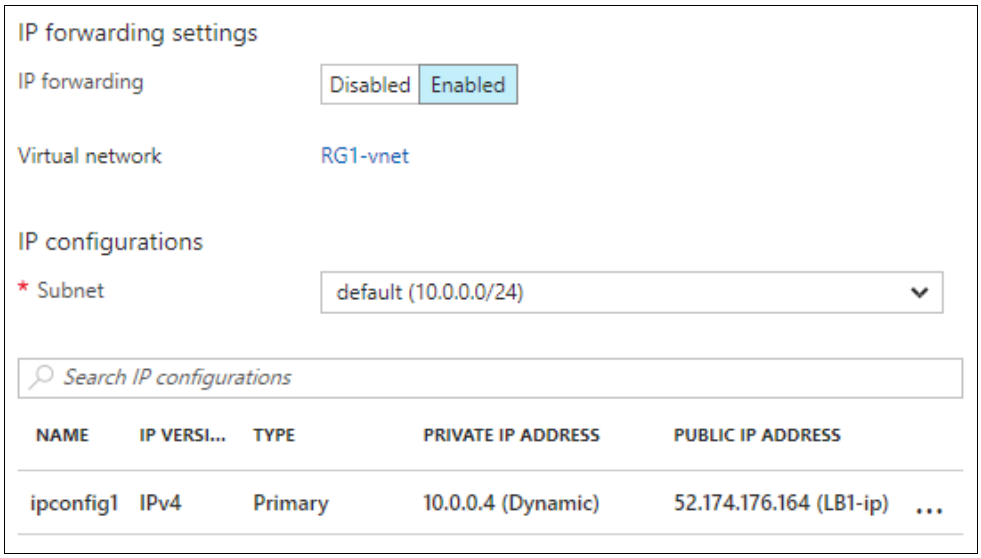
2. Read the Purchase details and terms of use, and if you're happy to proceed click **Purchase**
3. The load balancer will now be deployed

Enable IP Forwarding for Layer 4 Services:

If you'll be configuring layer 4 services, ensure that IP forwarding is enabled, this allows the VM to accept traffic that is not addressed to itself, i.e. the return traffic from the load balanced servers to the client. For an HA pair, this must be done on both appliances. To enable IP forwarding:

1. In the Azure Management Portal, select the *Virtual Machines* option, click on the newly deployed Load Balancer VM, click on *Network interfaces* and then select the network interface attached to the load balancer, then click *IP configurations*

2. Ensure that IP forwarding is enabled as shown below:



8. Accessing the Appliance

ACCESSING THE APPLIANCE USING THE WEBUI

In a browser, navigate to the Public IP address or FQDN on port 9443 , i.e.

https://<Public IP Address>:9443
or
https://<FQDN>:9443

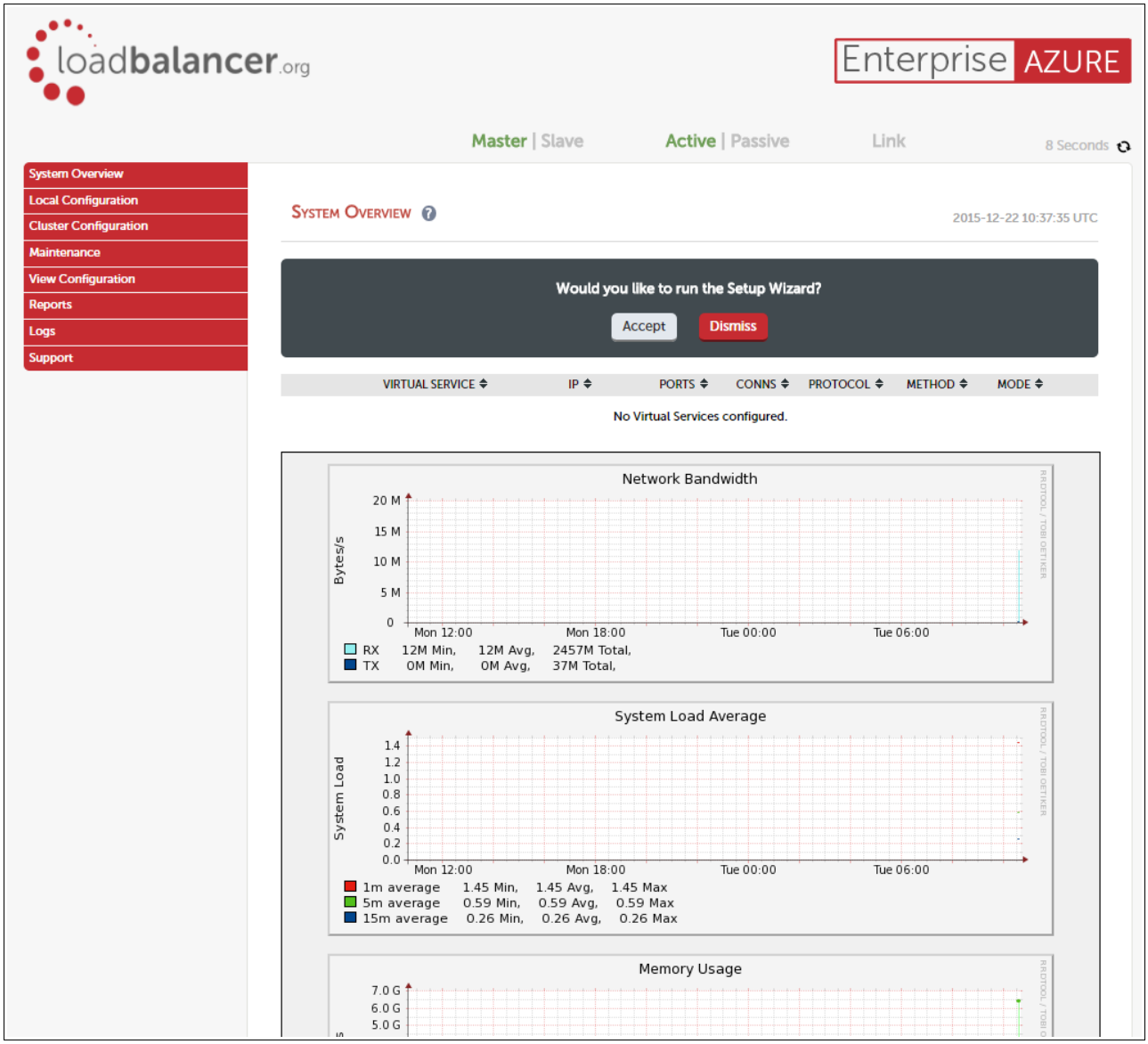
> Note:
>
> To configure an FQDN in Azure under the Resource Manager model please refer to this link.

You'll receive a warning about the certificate as it's a self signed cert not related to an Internet based CA. Confirm you want to continue and a login prompt will be displayed. Use the following default credentials:

Username: loadbalancer
Password: loadbalancer

> Note:
>
> To change the password for the 'loadbalancer' account, use the WebUI option: *Maintenance > Passwords.*

Once logged in, the WebUI is displayed:



WEBUI MENU OPTIONS

The main menu options are as follows:

System Overview – Displays a graphical summary of all VIPs, RIPS and key appliance statistics
Local Configuration – Configure local host settings such as DNS, Date & Time etc.
Cluster Configuration – configure load balanced services such as VIPs & RIPs
Maintenance – Perform maintenance tasks such as service restarts and taking backups
View Configuration – Display the saved appliance configuration settings
Reports – View various appliance reports & graphs
Logs – View various appliance logs
Support – Create a support download & contact the support team

CHECKING FOR UPDATES

Once you have access to the WebUI, we recommend that you use the online update feature to ensure that you're running the very latest version of the appliance. To check for updates, use the WebUI option: *Maintenance > Software Update* and click the **Online Update** button. If updates are available, you'll be presented with a list of changes that are included in the update. To start the update, click the second **Online Update** button at the bottom of the screen. Updates are incremental, so repeat the process until you're informed that no more updates are available.

APPLIANCE LICENSING

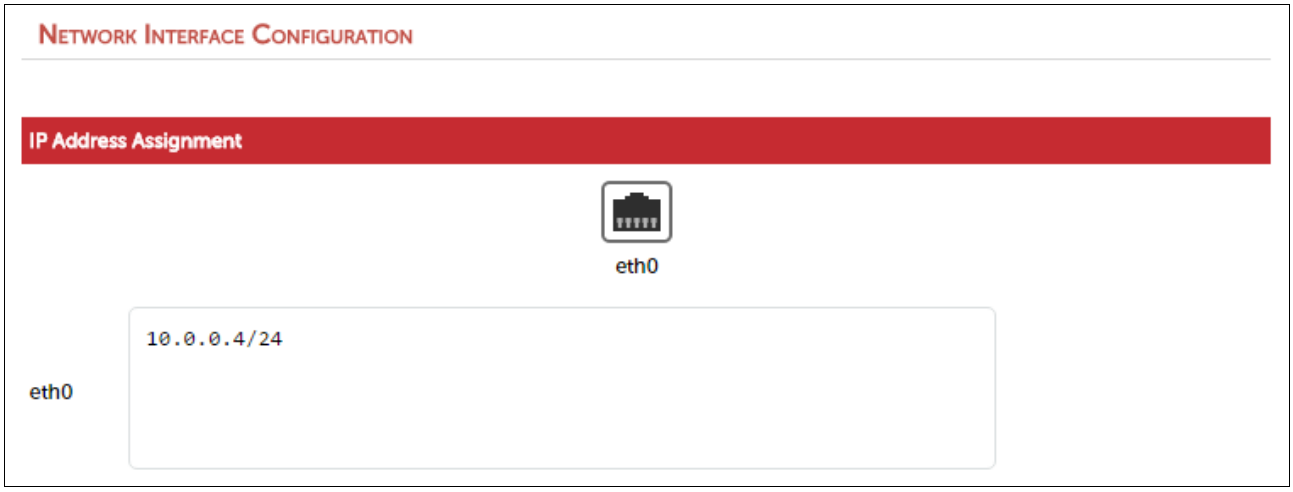
If you've deployed the BYOL version of the appliance, by default it runs as a 30 day trial and is completely unrestricted during this time. After 30 days, the appliance continues to work but it's no longer possible to make changes to the configuration. When a license is purchased, you'll be provided with a license key file by our sales team. This must then be installed on your appliance. To install the license, use the WebUI option: *Local Configuration > License Key* to browse to and select the license file provided. Once selected, click **Install License Key** to apply the license. We recommend that you should check for updates *before* applying the license key.

ENTERPRISE AZURE NON-STANDARD WEBUI MENU OPTIONS

Enterprise Azure has a number of differences to the standard hardware/virtual product range due to the way the Microsoft Azure environment works.

The menu options that are different are detailed below. For all others please refer to our main administration manual: http://pdfs.loadbalancer.org/loadbalanceradministrationv8.pdf

Local Configuration > Network Interface Configuration



Notes:

- Shows the private IP address allocated to the Virtual Machine

> Note:
>
> The IP address cannot be changed via the WebUI, this must be done via the Azure Portal.

ACCESSING THE APPLIANCE USING SSH

When the appliance is deployed, *Authentication type* must be set to either **SSH Public key** or **Password**. When set to **SSH Public Key**, a key pair must be manually generated outside of the Azure environment using tools such as ssh-keygen under Linux and PuttyGen under Windows. Once the key pair is generated, the public key must be copied into the *SSH public key* field at VM deployment, and the private key is then used on the SSH client machine to access the VM.

GENERATING SSH KEYS

The steps below show how to generate SSH key pairs using Linux and Windows.

Using Linux

STEP 1 – Generate a keypair using ssh-keygen:

All Distros:

```
# ssh-keygen -q -t rsa -b 2048 -f <output filename>
```

e.g.

```
# ssh-keygen -q -t rsa -b 2048 -f AzureKeys
```

When prompted, enter a pass-phrase, or leave empty for no pass-phrase:

```
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

2 files are created:

- **AzureKeys** – this is the Private Key file and is used on the SSH client machine
- **AzureKeys.pub** – this is the Public Key file, the contents are copied into the *SSH public key* field when the VM is deployed

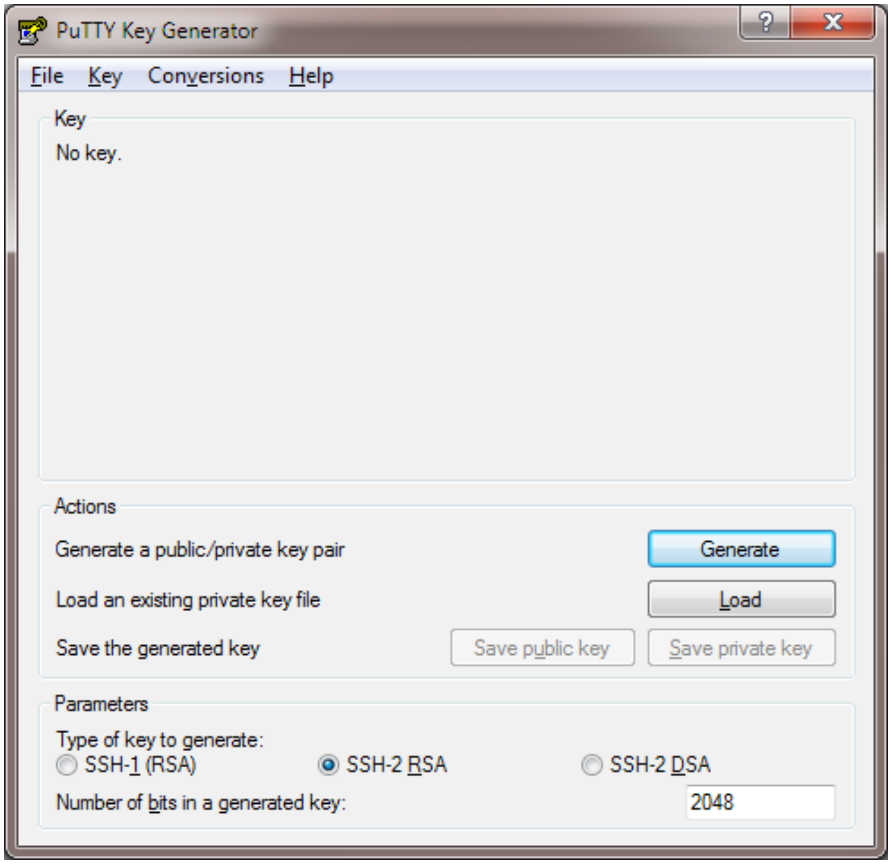
Using Windows

STEP 1 - Install PuTTY

1. Download PuTTY from this link:
   http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
2. Run the installer

STEP 2 - Use PuTTYgen to generate a Public/Private key pair:

1. Browse to the PuTTY program folder and run PuTTYgen



2. Click the **Generate** button
3. As directed, move the mouse around to create random keys
4. Once generated, click the **Save public key** and **Save private key** buttons to save the keys

ACCESSING THE APPLIANCE FROM LINUX

Start SSH specifying the private key file and login as the user defined when deploying the VM:

e.g.

Using the IP address:

```
# ssh -i /root/AzureKeys lbuser@1.2.3.4
```

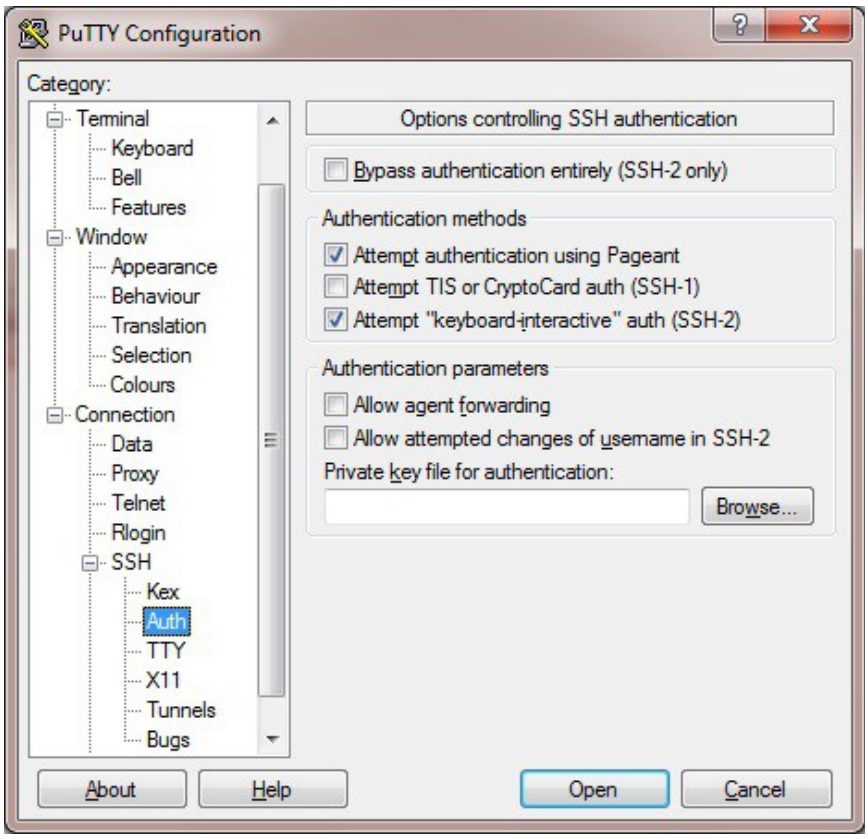
Or using the fqdn:

```
# ssh -i /root/AzureKeys lbuser@fqdn
```

> Note:
>
> To configure an FQDN in Azure under the Resource Manager model please refer to this link.

ACCESSING THE APPLIANCE FROM WINDOWS USING PUTTY

1. Run PuTTY
2. Expand the SSH section and select *Auth* as shown below



3. Click **Browse** and select the private key created earlier
4. Click **Open** to start the SSH session
5. Login using the username specified when deploying the instance, no password will be required

> Note:
>
> To enable full root access, the following command can be used once logged in to the appliance via SSH: $ sudo su

9. Configuration Examples

The following sections provide a number of examples to help illustrate how the load balancer can be deployed.

1 - LOAD BALANCING WEB SERVERS – 1 SUBNET, LAYER 7

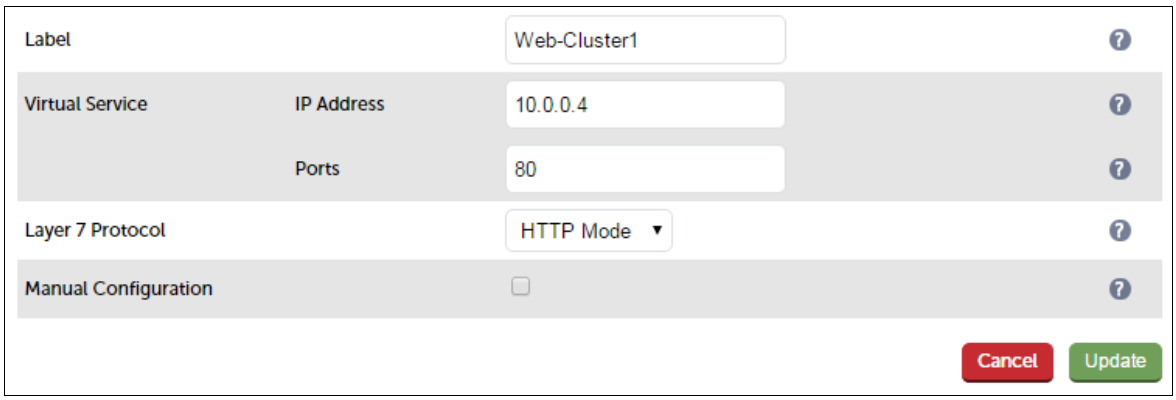
This is a simple layer 7 example using one public subnet for both the load balancer and the web servers.

a) Setting up Azure

1. Deploy the load balancer instance as described earlier on page 6
2. Deploy your required web server VM's into the same VNet & subnet as the load balancer
3. Ensure the Network Security Group includes port 80
4. Public IP addresses are not needed when deploying the web servers (real servers) instances since the load balancer is configured to send traffic to the private IP address of each web server

b) Setting up the Virtual Service

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 7 – Virtual Service* and click **Add a New Virtual Service**
2. Enter the following details:



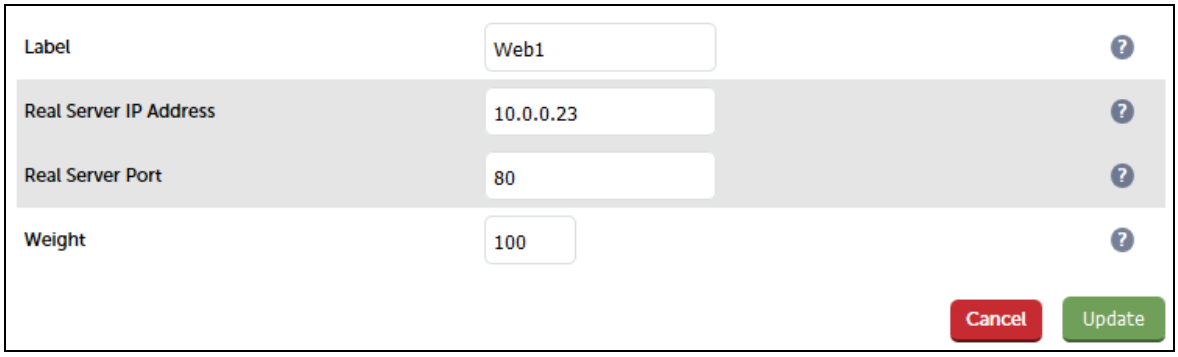
3. Enter an appropriate label for the VIP, e.g. **Web-Cluster1**
4. Set the *Virtual Service IP Address* field to an appropriate value. This can be either:

   - the appliance IP address - this can be viewed using the WebUI option: *Local Configuration > Network Interface Configuration*
     Or
   - **0.0.0.0** – this means all IP's on the local appliance, which is functionally equivalent to specifying the appliance IP address

5. Set the *Virtual Service Ports* field to **80**
6. Leave *Layer 7 Protocol* set to **HTTP mode**
7. Click **Update**

c) Setting up the Real Servers

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 7 – Real Servers* and click **Add a new Real Server** next to the newly created VIP
2. Enter the following details:



3. Enter an appropriate label for the RIP, e.g. **Web1**
4. Change the *Real Server IP Address* field to the required IP address, e.g. **10.0.0.23**
5. Click **Update**

6. Repeat the above steps to add your other Web Server(s)

d) Applying the new Layer 7 Settings

1. Once the configuration is complete, use the **Reload HAProxy** button at the top of the screen to commit the changes

e) Testing & Verification

1. To test the configuration is working, browse to the public IP address or FQDN on port 80, i.e.

   http://<Public IP Address>
   or
   http://<FQDN>

> Note:
>
> To configure an FQDN in Azure under the Resource Manager model please refer to this link.

2 - LOAD BALANCING WEB SERVERS – 1 SUBNET, LAYER 7, SSL TERMINATION

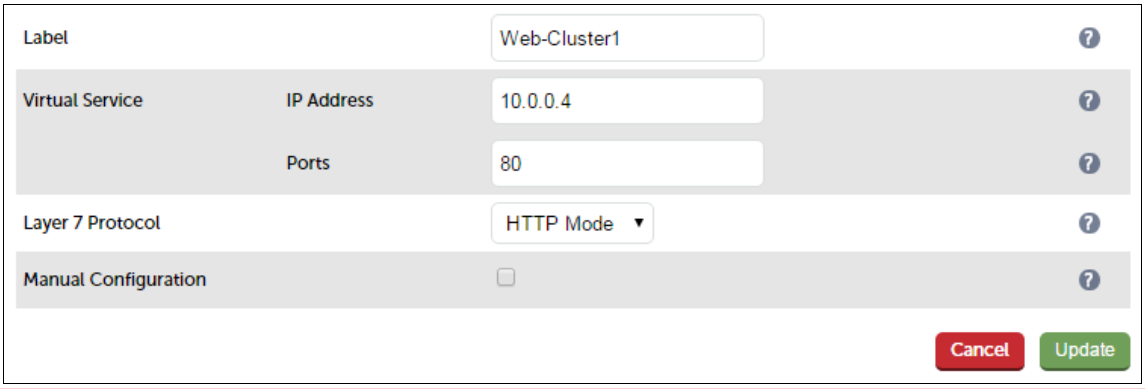
This is similar to the first example with the addition of setting up SSL termination on the load balancer. We generally recommend that SSL should be termination on the backend servers rather than the load balancer for scalability reasons, although in some cases terminating on the load balancer may be preferred.

a) Setting up Azure

1. Deploy the load balancer instance as described earlier on page 6
2. Deploy your required web server VM's into the same VNet & subnet as the load balancer
3. Ensure the Network Security Group includes port 443
4. Public IP addresses are not needed when deploying the web servers (real servers) instances since the load balancer is configured to send traffic to the private IP address of each web server

b) Setting up the Virtual Service

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 7 – Virtual Service* and click **Add a New Virtual Service**
2. Enter the following details:



3. Enter an appropriate label for the VIP, e.g. **Web-Cluster1**
4. Set the *Virtual Service IP Address* field to an appropriate value. This can be either:

   - the appliance IP address - this can be viewed using the WebUI option: *Local Configuration > Network Interface Configuration*

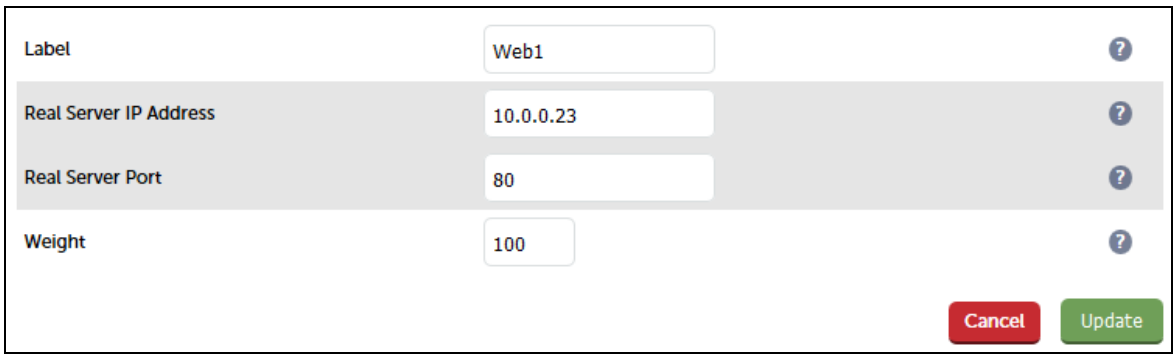
     Or

   - **0.0.0.0 –** this means all IP's on the local appliance, which is functionally equivalent to specifying the appliance IP address

5. Set the *Virtual Service Ports* field to **80**
6. Leave *Layer 7 Protocol* set to **HTTP mode**
7. Click **Update**

c) Setting up the Real Servers

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 7 – Real Servers* and click **Add a new Real Server** next to the newly created VIP
2. Enter the following details:

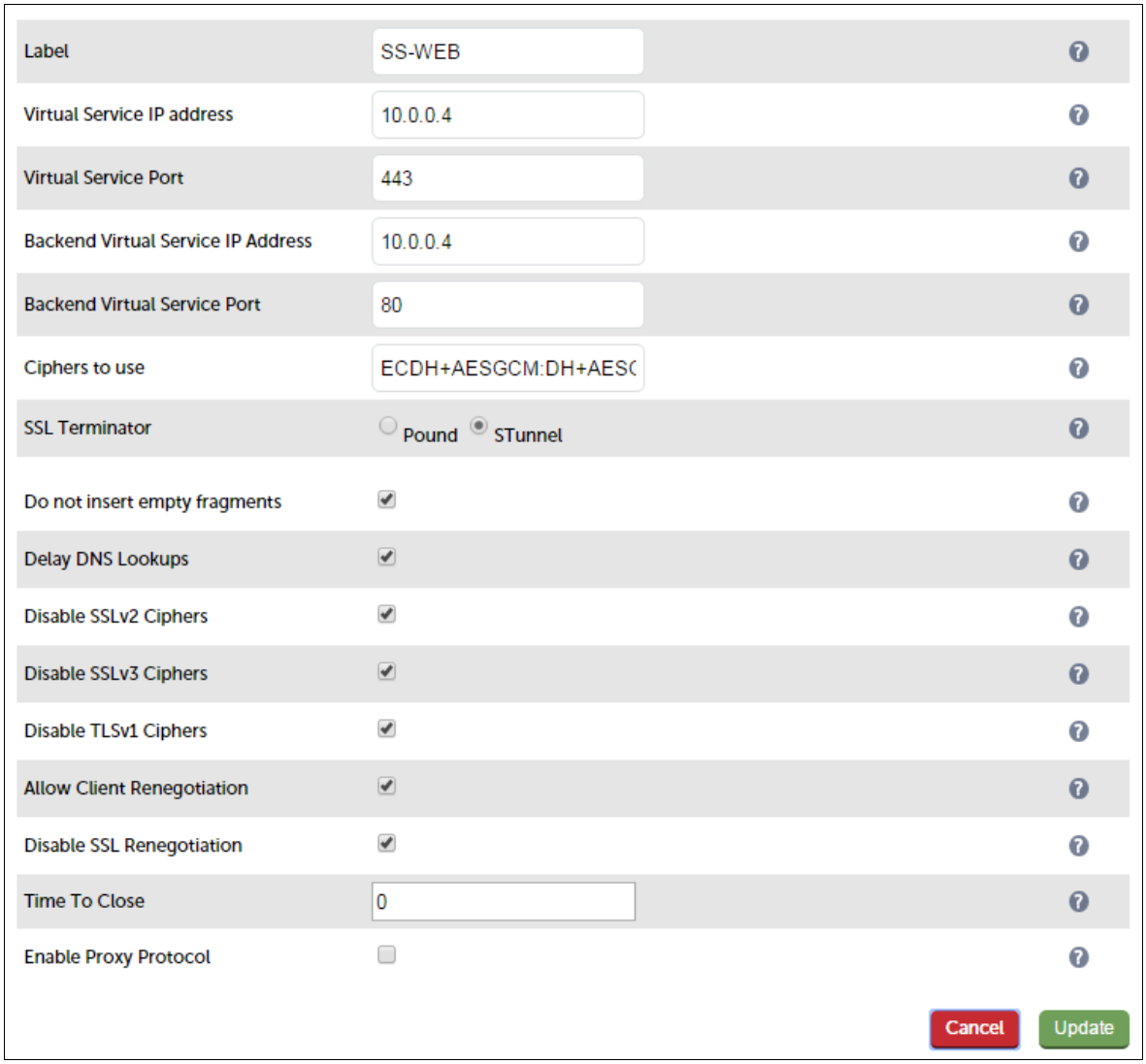


3. Enter an appropriate label for the RIP, e.g. **Web1**
4. Change the *Real Server IP Address* field to the required IP address, e.g. **10.0.0.23**
5. Click **Update**
6. Repeat the above steps to add your other Web Server(s)

d) Configuring SSL Termination

1. Using the WebUI, navigate to: *Cluster Configuration > SSL Termination* and click **Add a New Virtual Service**

2. Enter the following details:



3. Enter an appropriate label for the VIP, e.g. **SSL-WEB**
4. Set the *Virtual Service IP address* to be the same as the VIP created in step c) e.g. **10.0.0.4**
5. Set the *Virtual Service Ports* field to **443**
6. Set the *Backend Virtual Service IP address* to be the same as the VIP created in step c) e.g. **10.0.0.4**
7. Set the *Backend Virtual Service Ports* field to **80**
8. Leave all other settings at their default values
9. Click **Update**

SSL Certificate Notes:

- A default self-signed certificate will be used when setting up SSL Termination
- To change this using the WebUI, navigate to: *Cluster Configuration > SSL Termination*
- Click **[Certificate]** next to the Virtual Service
- If you already have a certificate, use the **Upload prepared PEM/PFX file** option at the bottom of the screen to upload it

- If you don't have a certificate, you can create a CSR using the **Generate SSL Certificate Request** section. This will create the CSR in the upper pane of the **Upload Signed Certificate** section based on the settings you enter. This should be copied and sent to your CA
- Once the signed certificate is received copy/paste it (along with any required intermediate certificates) the lower pane of the **Upload Signed Certificate** section, and click **Upload Signed Certificate**

e) Applying the new Settings

- Once the configuration is complete:

  1. use the **Reload HAProxy** button at the top of the screen to commit the changes
  2. use the **Restart STunnel** button at the top of the screen to commit the changes

f) Testing & Verification

- To test the configuration is working, browse to the public IP address or FQDN on HTTPS port 443, i.e.

  https://<Public IP Address>
  or
  https://<FQDN>

  > Note:
  > To configure an FQDN in Azure under the Resource Manager model please refer to this link.

3 - LOAD BALANCING WEB SERVERS – 2 SUBNETS, LAYER 4

This example uses 2 subnets - one public subnet for the load balancer and one private subnet for the web servers. The load balancer has a single network interface located in the first subnet. Routing rules for the second private subnet must be changed so that return traffic passes back via the load balancer.

a) Setting up Azure

1. Deploy the load balancer instance into the first public subnet as described earlier on page 6
2. Deploy your required web server VM's into the second private subnet
3. Configure security rules for the 2 subnets (assuming an Internet facing deployment):

   Public Subnet (load balancer) :

   - Inbound rule - from 0.0.0.0/0 to port 80
   - Outbound rule – from 0.0.0.0/0 to private subnet, port 80

   Private Subnet (web servers) :

   - Inbound rule - from 0.0.0.0/0 to port 80

4. Modify routing rules for the second subnet so that all return traffic is routed via the load balancer:

1) Start Powershell & login to Azure Resource Manager:

```
Login-AzureRmAccount
```

2) Create a route to send all return traffic via the load balancer:

```
$route = New-AzureRmRouteConfig -Name RouteToLB -AddressPrefix 0.0.0.0/0
-NextHopType VirtualAppliance -NextHopIpAddress 10.0.1.4
```

*change **10.0.1.4** to the private IP address of the load balancer VM in the public subnet*

3) Create a route table that contains the route created above:

```
$routetable = New-AzureRmRouteTable -ResourceGroupName lbtest -Location
westeurope -Name RouteTable1 -Route $route
```

*change **lbtest** to your resource group name*
*change **westeurope** to your location*

4) Store the virtual network in a variable:

```
$vnet = Get-AzureRmVirtualNetwork -ResourceGroupName lbtest -Name vnet1
```

*change **lbtest** to your resource group name*
*change **vnet1** to your vnet name*

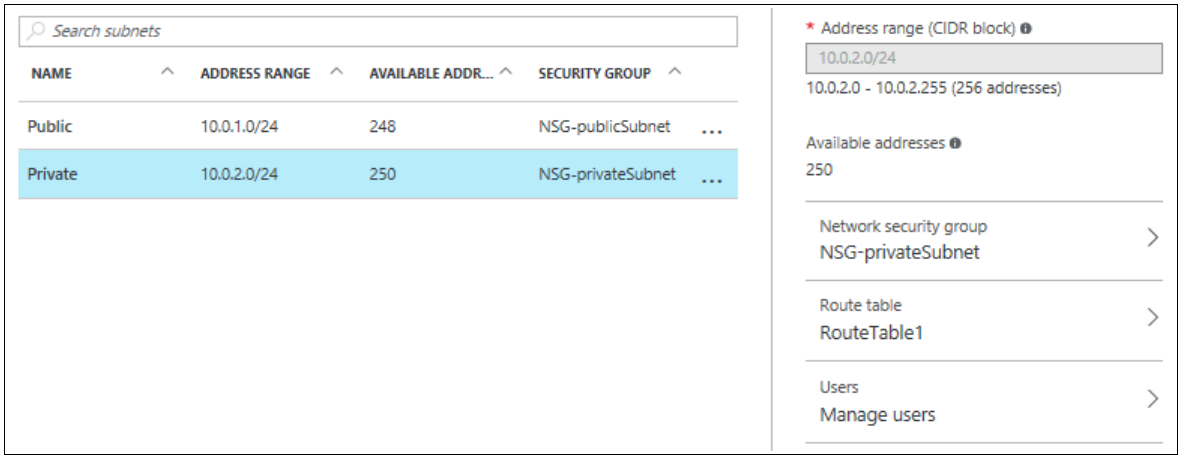
5) Save the new subnet configuration in Azure:

```
Set-AzureRmVirtualNetworkSubnetConfig -VirtualNetwork $vnet -Name Private
-AddressPrefix 10.0.2.0/24 -RouteTable $routetable
```

*change **Private** to the name of your second subnet*
*change **10.0.2.0/24** to your private subnet address*

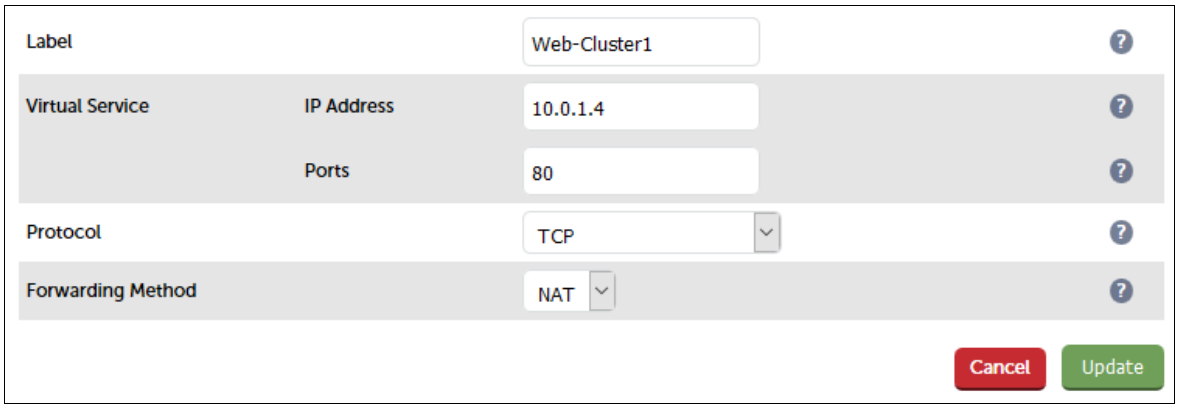
```
Set-AzureRmVirtualNetwork -VirtualNetwork $vnet
```

Verify route table association in the portal:



b) Setting up the Virtual Service

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 4 – Virtual Service* and click **Add a New Virtual Service**
2. Enter the following details:



3. Enter an appropriate label for the VIP, e.g. **Web-Cluster1**
4. Set the *Virtual Service IP Address* field to an appropriate value. This can be either:

   - the appliance IP address - this can be viewed using the WebUI option: *Local Configuration > Network Interface Configuration*

     OR

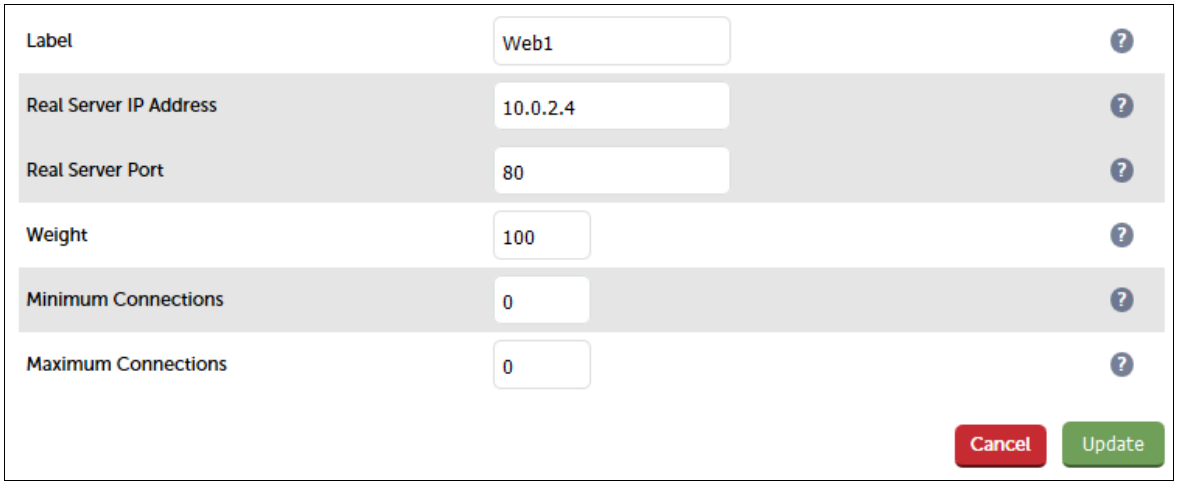
   - **0.0.0.0** – this means all IP's on the local appliance, which is functionally equivalent to specifying the appliance IP address

5. Set the *Virtual Service Ports* field to **80**
6. Leave *Protocol* set to **TCP**
7. Ensure *Forwarding Method* is set to **NAT**
8. Click **Update**

c) Setting up the Real Servers

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 4 – Real Servers* and click **Add a new Real Server** next to the newly created VIP

2. Enter the following details:



3. Enter an appropriate label for the RIP, e.g. **Web1**
4. Change the *Real Server IP Address* field to the required IP address, e.g. **10.0.2.4**
5. Set the *Real Server Port* field to **80**
6. Click **Update**
7. Repeat the above steps to add your other Web Server(s)

d) Testing & Verification

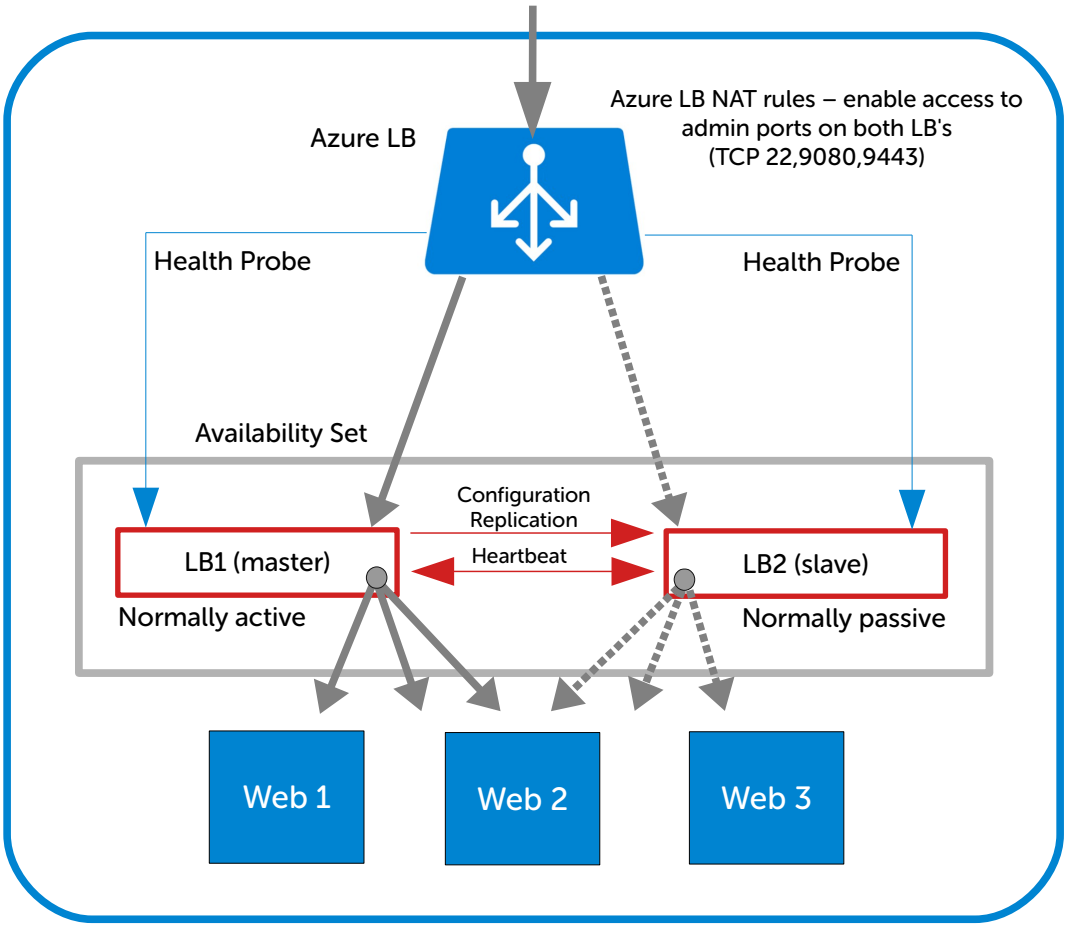
1. To test the configuration is working, browse to the public IP address or FQDN on port 80, i.e.

   http://<Public IP Address>
   or
   http://<FQDN>

> Note:
> To configure an FQDN in Azure under the Resource Manager model please refer to this link.

10. Configuring High Availability using two instances (Master & Slave)

Enterprise Azure supports HA by using two appliances configured as a clustered pair in combination with an Azure load balancer instance as shown in the diagram below.



- LB1 and LB2 are configured as a clustered pair. In this mode, one device is active (typically the master appliance) and the other is passive (typically the slave appliance)
- The probe service on TCP port 6694 is up on the active appliance (LB1) and down on the passive appliance (LB2), The active appliance responds with **200 OK**
- The Azure load balancer probes port 6694 on LB1 and LB2 and then forwards traffic to the active load balancer appliance (LB1)
- If the master appliance fails for any reason, the passive appliance will detect this, become active and bring up the probe service on port 6694. In turn, the Azure load balancer detects this and will then forward traffic to the slave device (LB2)

> Note:
>
> Appliance HA is currently only supported when only standard (i.e. non-transparent) layer 7 services are configured on the load balancer.

> Note:
>
> The following procedure assumes the first appliance is already up and running, and it will be the master unit of the clustered pair.

Step 1 – Deploy a second load balancer VM

1. Please refer to the steps starting on page 6
2. Ensure that both load balancer VM's are in the ***same Availability Set***

Step 2 – Update Network Security Group Settings

1. Ensure that your Network Security Group(s) permit the following communication between the 2 appliances:

   - TCP port 22 (SSH)
   - UDP port 6694 (heartbeat)

   > Note:
   >
   > The appliances must also be able to ping each other, this is enabled by default within the same Virtual Network.

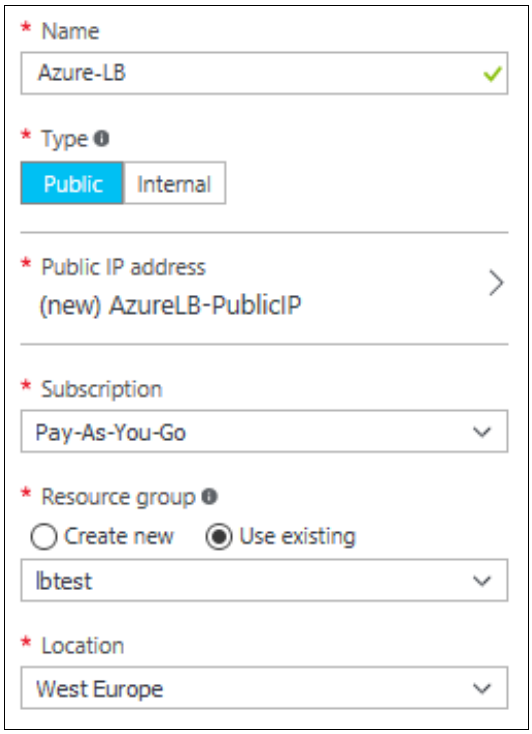
2. Ensure that your Network Security Group(s) permit the following inbound communication from the Azure load balancer to both appliances:

   - TCP port 6694 (Azure probe)

Step 3 – Configure the Azure Load balancer

1. First add the Azure Load Balancer

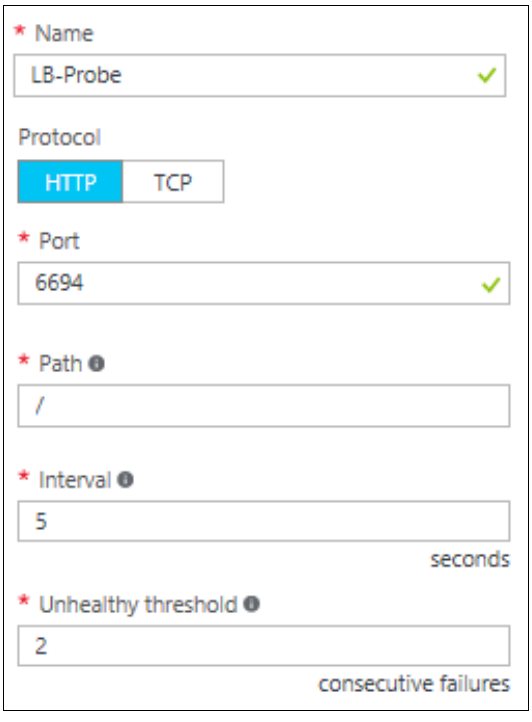
   - In the Azure Portal select *More Services* at the bottom of the main menu
   - Under *NETWORKING* select *Load balancers*
   - Click **Create Load balancers**
   - Enter an appropriate name, .e.g. **Azure-LB**
   - If deploying within a private network set *Type* to **Internal** , if it's public facing select **Public**



- Configure the IP address and other remaining settings according to your requirements

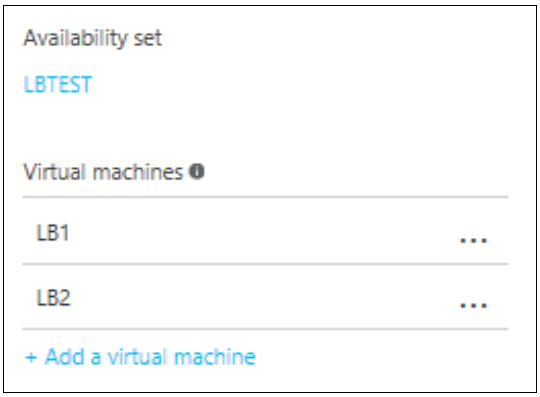
2. Next create a Health Probe

- Select the newly created Load balancer (you may need to click **Refresh** to see it)
- In the menu for the Load balancer, click *Health-probes*
- Click **Add**



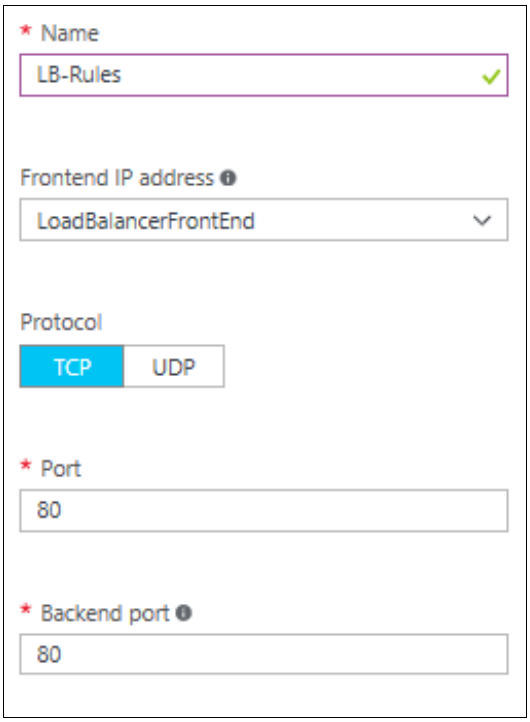
- Enter an appropriate name
- Set *Protocol* to **HTTP** - this will configure the Azure load balancer to look for a 200 OK response from each Loadbalancer.org Virtual Appliance
- Set the *Port* to **6694**
- Leave the remaining settings at their default values

3. Next create a Backend Pool as shown in the example below:

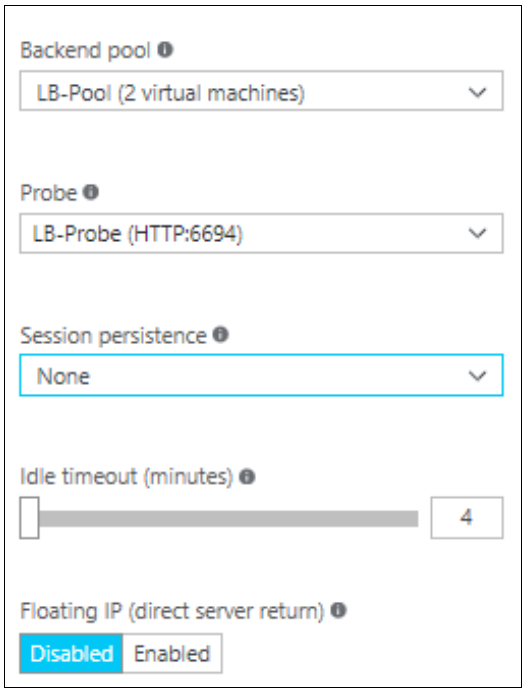


- Select the *Availability set* that contains the load balancer VM's
- Ensure that both load balancer *Virtual machines* are selected

4. Next configure Load Balancing rules as shown in the example below:



screenshot cont'd .....



- Set the *Frontend IP address* to the address of the Azure load balancer
- Set the *Port* and *Backend port* according to your requirements
- Select the *Backend pool* created previously
- Select the *Probe* created previously
- Leave *Session persistence* set to **None** – session persistence is not required since the Azure Load balancer will simply send all traffic to the working loadbalancer.org appliance, i.e the appliance that is responding to the HTTP probe on TCP port 6694

5. Next configure the Inbound NAT Rules for appliance administration related port access via the Azure load balancer frontend IP address. Note the following when creating the rules:

- Set *Service* to **Custom**
- Set *Protocol* to **TCP**
- Change the *Port* for each rule to suit your requirements
- Set *Associated to* to **Availability set**
- Set *Availability set* to the availability set that contains the load balancer VM's
- Set *Port Mapping* to **Custom**

Rules to be created:

Rule Name	Port	Target Port	Use
NAT-LB1-22	122	22	external access to SSH on LB1
NAT-LB2-22	222	22	external access to SSH on LB2
NAT-LB1-9443	19443	9443	external access to WebUI on LB1
NAT-LB2-9443	29443	9443	external access to WebUI on LB2

6. Once created, the rules are listed in Azure:

NAME	DESTINATION	TARGET	SERVICE	
NAT-LB1-22	13.81.205.27	LB1	Custom (TCP/122)	...
NAT-LB1-9443	13.81.205.27	LB1	Custom (TCP/19443)	...
NAT-LB2-22	13.81.205.27	LB2	Custom (TCP/222)	...
NAT-LB2-9443	13.81.205.27	LB2	Custom (TCP/29443)	...

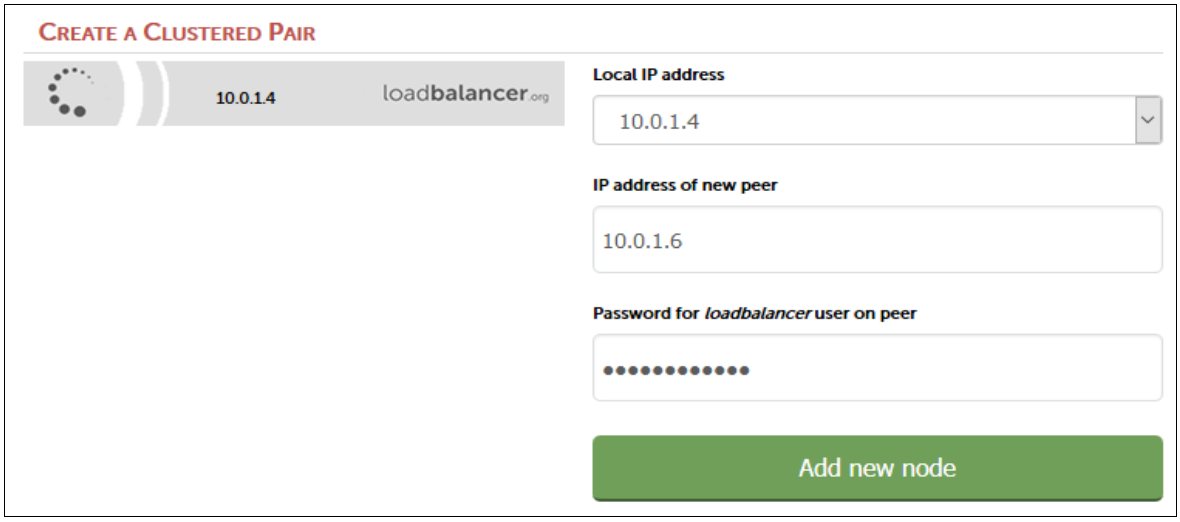
> Note:
> Add other rules if you need access to your load balanced servers via the Azure load balancer,

Step 4 – Configure the master Appliance to allow service control during failover / fail-back

1. Open the WebUI on the master unit
2. Navigate to: *Cluster configuration > Floating IP's*
3. In the *New Floating IP* field enter an IP address in the same subnet as the appliances – this address is not used for any connections, it's required to allow service control on both master & slave units
4. Click **Add Floating IP**

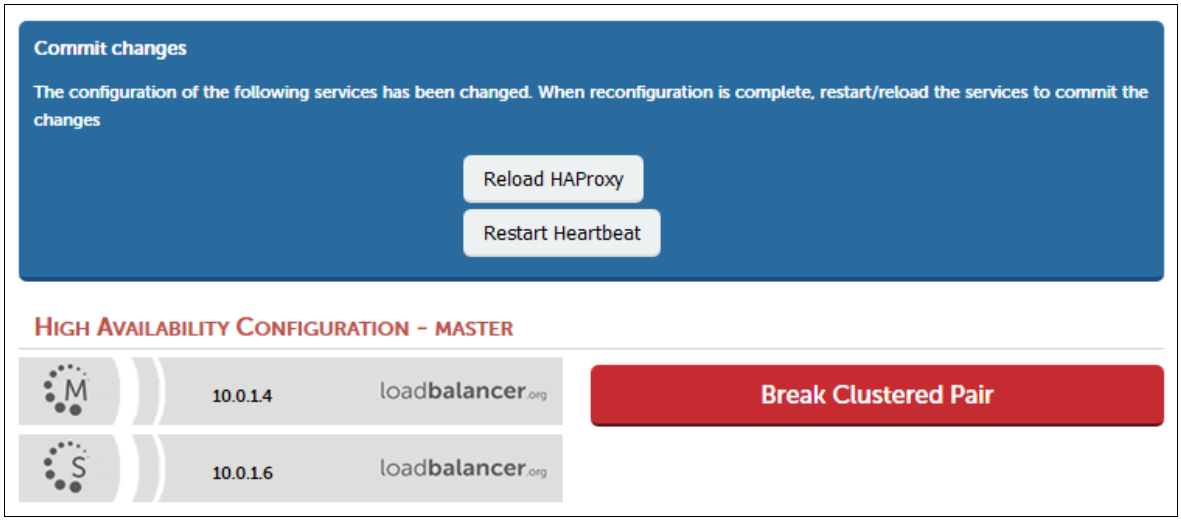
Step 5 – Configure Appliance High-Availability

1. Open the WebUI on the master unit
2. Navigate to: *Cluster Configuration > High Availability Configuration*



3. In the *IP address of new peer* field, enter the slave appliances private IP address
4. In the *Password for loadbalancer user on peer* field enter the relevant password, the default password is 'loadbalancer'
5. Click **Add new node**
6. Once the pairing configuration has finished, any service restart messages and the confirmed pair

message will be displayed as shown below:



7. Restart the services using the buttons presented, in this example HAProxy and Heartbeat

Step 6 – Verify Synchronization State

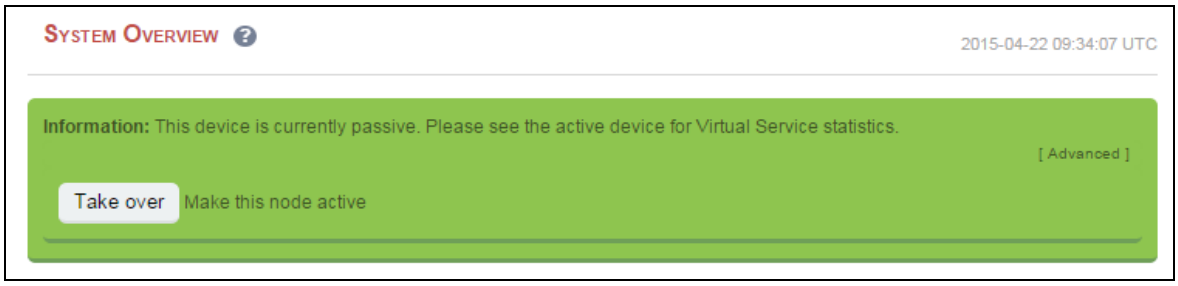
1. Once all services have restarted, the synchronization process will be complete
2. Verify that the status on the master & slave is as follows:

Master Unit:

Slave Unit:

> Note:
> If no services have been configured, 'Active' will be greyed out on both instances.

The slave can be made active by clicking **[Advanced]** in the green box, and then clicking the **Take over** button



Other states:

Master | Slave	**Active** | Passive	Link	this is a master unit, it's active, no slave unit has been defined
Master | Slave	**Active** | Passive	**Link**	this is a master unit, it's active, a slave has been defined but the link to the slave is down. ***Action***: *check & verify the heartbeat configuration*
Master | **Slave**	**Active** | Passive	**Link**	this is a slave unit, it's active (a failover from the master has occurred) and the heartbeat link to the master has been established

11. Testing & Validation

TESTING LOAD BALANCED SERVICES

For example, to test a web server based configuration, add a page to each web servers root directory e.g. ***test.html*** and put the server name on this page for easy identification during the tests.

Use two or more clients to do the testing. Open up a web browser on each test clients and enter the URL for the VIP e.g. **http://104.40.133.119**

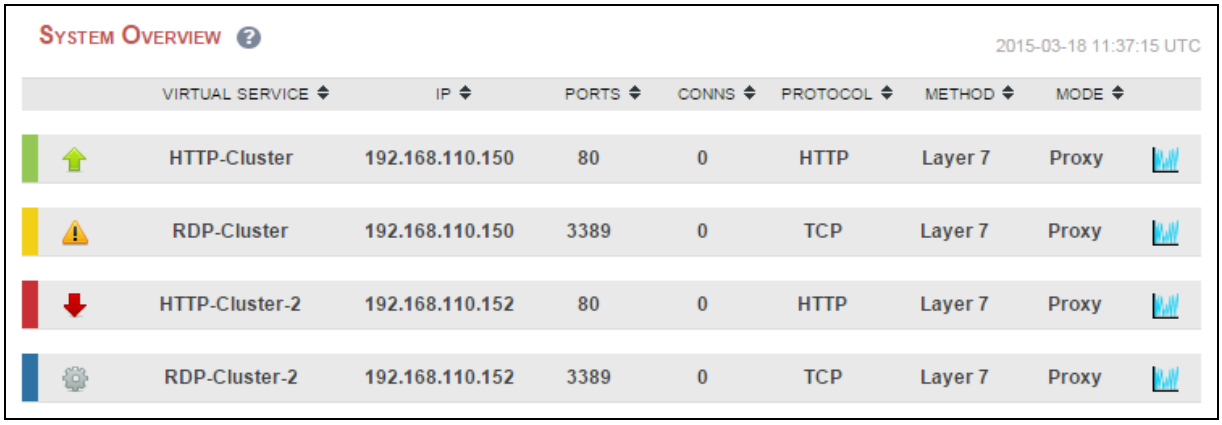
Provided that persistence is disabled, each client should see a different server name because of the load balancing algorithm in use , i.e. they are being load balanced across the cluster.

Why test using two clients? *If you use a single client it will most likely keep on hitting the same server for multiple requests. This is to do with the way that the load balancing algorithms are optimized.*

DIAGNOSING VIP CONNECTION PROBLEMS

1. ***Make sure that the device is active*** - this can be checked in the WebUI. For a single appliance, the status bar should report **Master** & **Active** as shown below:

2. ***Check that the Real Servers are up*** - Using *System Overview* make sure that none of your VIPs are colored red. If they are, the entire cluster is down (i.e. all Real Servers). Green indicates a healthy cluster, yellow indicates that your cluster may need attention (one or more of the Real Servers may be down), and blue indicates all Real Server have been deliberately taken offline (by using either Halt or Drain).



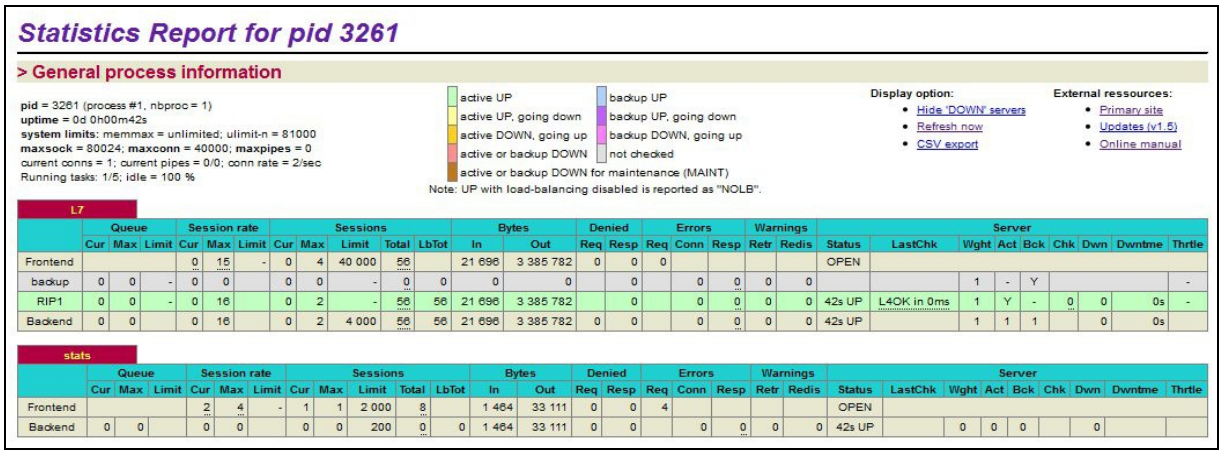
3. *Check the connection state*

For layer 4 (NAT mode) VIPs, check *Reports > Layer 4 Current Connections* to view the current traffic in detail. Any packets with state **SYN_RECV** often implies a return traffic routing issue, so make sure that the routing rules for the real server subnet have been configured correctly

For Layer 7 VIPs, check *Reports > Layer 7 Status.* The default credentials required are:

username: loadbalancer
password: loadbalancer

This will open a second tab in the browser and display a statistics/status report as shown in the example below (this is accessed on port TCP/7777 so make sure that the inbound rules allow connections on this port) :



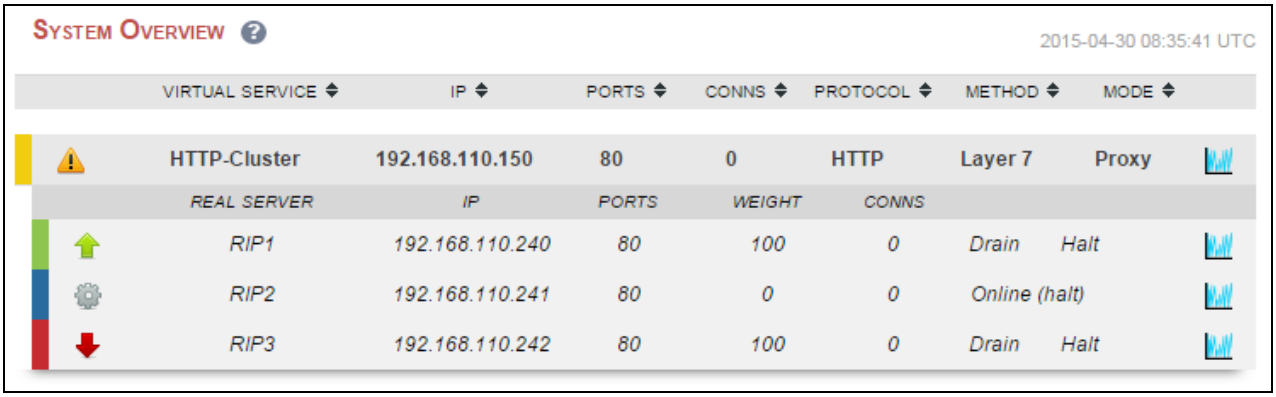
TAKING REAL SERVERS OFFLINE

1) Using the *System Overview* check that when you Halt one of the Real Servers the connections are redirected to the other server in the cluster.

2) Stop the web service/process on one of the servers, wait a few seconds (for the load balancer to detect the change) and then refresh the browsers on both clients. They should now both switch to the same server (since one has been removed from the load balancing list). Also check that the server is shown red (down) in the system overview.

3) Start the web service/process on the server, wait a few seconds and then refresh the browsers again. After a few refreshes they should again show different web servers. Also check that the server is shown green (up) in the system overview.

The *System Overview* shows the status as these tests are performed:



In this example:

RIP1 is green, this indicates that it's operating normally

RIP2 is blue, this indicates that it has been either Halted or Drained. in this example Halt has been used as indicated by *Online (Halt)* being displayed. If it had been drained it would show as *Online (Drain)*

RIP3 is red, this indicates that it has failed a health check

USING REPORTS & LOG FILES

The appliance includes several logs and reports that are very useful when diagnosing issues. Both are available as main menu options in the WebUI. Details of both can be found in the administration manual.

12. Loadbalancer.org Technical Support

If you have any questions regarding the appliance or how to load balance your application, please don't hesitate to contact our support team using the following email address: support@loadbalancer.org

For more details please refer to the administration manual:

http://pdfs.loadbalancer.org/loadbalanceradministrationv8.pdf

13. Company Contact Information

Website	URL: www.loadbalancer.org
North America (US)	Loadbalancer.org, Inc. 4250 Lancaster Pike, Suite 120 Wilmington DE 19805 USA Tel: +1 888.867.9504 Fax: +1 302.213.0122 Email (sales): sales@loadbalancer.org Email (support): support@loadbalancer.org
North America (Canada)	Loadbalancer.org Ltd 300-422 Richards Street Vancouver, BC V6B 2Z4 Canada Tel: +1 866.998.0508 Fax: +1 302.213.0122 Email (sales): sales@loadbalancer.org Email (support): support@loadbalancer.org
Europe (UK)	Loadbalancer.org Ltd. Compass House North Harbour Business Park Portsmouth, PO6 4PS UK Tel: +44 (0)330 3801064 Fax: +44 (0)870 4327672 Email (sales): sales@loadbalancer.org Email (support): support@loadbalancer.org
Europe (Germany)	Loadbalancer.org GmbH Tengstraße 27 D-80798 München Germany Tel: +49 (0)89 2000 2179 Fax: +49 (0)30 920 383 6495 Email (sales): vertrieb@loadbalancer.org Email (support): support@loadbalancer.org