loadbalancer.org

Enterprise AWS Quick Start Guide

v8.2.5

Rev. 1.0.0

Copyright © 2002 – 2017 Loadbalancer.org, Inc

Table of Contents

1. Introduction....4
2. About Enterprise AWS....4
   - Main Differences to the Non-Cloud Product....4
   - Why use Enterprise AWS?....5
3. Amazon Terminology....5
4. Getting Started....5
5. Deployment Concepts....6
   - Overview....6
   - AWS Topology Options....6
     - Single Availability Zone....6
     - Dual Availability Zones....7
   - Creating a VPC....8
   - VPC IP Address Types....9
   - IP address Allocation Options & Requirements....10
     - Internal (Private Network) Deployments....10
     - Public facing Deployments....10
   - VPC Network Interfaces (ENI)....11
   - 10GB Support....11
   - Instance Type....11
6. Deploying Enterprise AWS....11
   - Create & Configure a VPC....11
   - Accessing & Deploying the AMI....13
   - Checking your Subscriptions....19
7. Accessing the Appliance....19
   - Using the WebUI....19
     - WebUI Menu Options....20
   - Checking For Updates....21
   - Appliance Licensing....21
   - Enterprise AWS Non-standard WebUI Menu Options....21
   - Accessing the Appliance using SSH....25
     - Using Linux....25
     - Using Windows....26
8. Configuration Examples....28
   - Deployment Notes....28
   - 1 - Web Servers – 1 subnet, 1 load balancer network interface, layer 7....29
   - 2 - Web Servers - 1 subnet, 1 load balancer network interface, layer 4....31
   - 3 - Web Servers – 2 subnets, 2 load balancer network interfaces, layer 7....34
   - 4 - Web Servers - 2 subnets, 1 load balancer network interface, layer 7, transparent....37
   - 5 - Web Servers - 1 subnet, 1 load balancer network interface, layer 7, SSL termination....39
   - 6 - RD Session Hosts - 2 subnets, 1 load balancer network interface, layer 7....42
   - 7 - Web Servers - 2 subnets, 1 load balancer network interface, layer-4....44
9. Configuring High Availability using two Instances (Master & Slave)....47
10. Configuring High Availability using two Instances across Availability Zones....51

11. Testing & Validation....55
Testing Load Balanced Services....55
Diagnosing VIP Connection Problems....55
Taking Real Servers Offline....57
Using Reports & Log Files....57
12. Loadbalancer.org Technical Support....57
13. Appendix....58
1 - IAM Role Configuration....58
2 - Configuring the load balancer to auto add/remove auto-scaled Real Servers....60
3 - Configuring Auto-Scaling to auto deploy a new LB.org Instance on Failure....60
4 - Company Contact Information....66

1. Introduction

Amazon Web Services offers a broad set of global cloud-based services. These services help organizations move faster, lower IT costs, and scale. 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.

Enterprise AWS allows customers to rapidly deploy and configure a load balancing solution within the Amazon cloud. The latest Loadbalancer.org AWS appliance enables both Layer 4 and layer 7 virtual services to be quickly and easily configured.

2. About Enterprise AWS

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 AWS 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 47. From v8.2.2 it's also possible to deploy 2 instances in different AZ's for high availability, this is achieved using a primary/secondary master model rather than the master/slave model, please refer to pages 7 and 51 for more details.

Enterprise AWS is based on our main hardware/virtual product and has almost identical features. There are certain differences due to the way the Amazon EC2 environment works, these are listed below.

MAIN DIFFERENCES TO THE NON-CLOUD PRODUCT

- The network setup is customized for Amazon EC2 deployment
- Layer 4 Direct Routing (DR) mode is ***not supported***
- Dual interface layer 4 NAT mode where each interface of the load balancer is connected to a different subnet and the default gateway of the real servers is configured to be the load balancer is ***not supported***

  ==> Single interface mode should be used instead, and a default route with the target set as the load balancer instance should be added to the routing table of the subnet where the real servers are located – please refer to page 37 for an example

  *==> Also, for a clustered pair of load balancers (master & slave) the AWS routing table for the Real Server subnet must be dynamically changed when failover from the active to passive device occurs. This can be achieved using the WebUI option: Cluster Configuration > Heartbeat Advanced, and the AWS command **ec2-replace-route** as detailed on page 47*
- Dual interface layer 7 SNAT mode with TProxy where each interface of the load balancer is connected to a different subnet and the default gateway of the real servers is configured to be the load balancer is ***not supported***

  ==> Single interface mode should be used instead, and a default route with the target set as the load balancer instance should be added to the routing table of the subnet where the real servers are located – please refer to page 44 for an example

  *==> Also, for a clustered pair of load balancers (master & slave) the AWS routing table for the Real Server subnet must be dynamically changed when failover from the active to passive device occurs. This can be achieved using the WebUI option: Cluster Configuration > Heartbeat Advanced, and the AWS command **ec2-replace-route** as detailed on page 47*
- The WebUI is not accessible on HTTP port 9080, only HTTPS port 9443

WHY USE ENTERPRISE AWS?

Amazon enables users to setup *Elastic Load Balancing* for load balancing other EC2 instances running in the cloud. This does provide basic load balancing functionality but is limited in several areas. Loadbalancer.org's Enterprise AWS load balancer provides the following additional features & advantages:

1. Load balances virtually any TCP or UDP based protocol
2. Ability to deploy a clustered pair of instances for High Availability: one active, one passive
3. Load balances both EC2 based and non-EC2 based servers
4. Supports customizable timeouts for custom applications beyond those offered by AWS
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 source IP based persistence
10. Supports RDP Cookie based persistence
11. Supports full integration with Remote Desktop Services Connection Broker
12. Supports multiple load balanced services running on multiple IP addresses

3. Amazon Terminology

	Description
Amazon AWS	Amazon Web Services
Amazon S3	Amazon Simple Storage Service
Amazon EC2	Amazon Elastic Compute Cloud
Amazon VPC	Amazon Virtual Private Cloud
Amazon AMI	Amazon Machine Image
Amazon EBS	Elastic Block Store
EIP	Elastic IP Address
ENI	Elastic Network Interface

4. Getting Started

To start using AWS, you'll need an Amazon account. If you don't already have one you can create one at the following URL: http://aws.amazon.com/console/

5. Deployment Concepts

OVERVIEW

Instances must be deployed within a VPC (Virtual Private Cloud). The simplest way to create and configure a VPC is to use the wizard available in the AWS/VPC console.

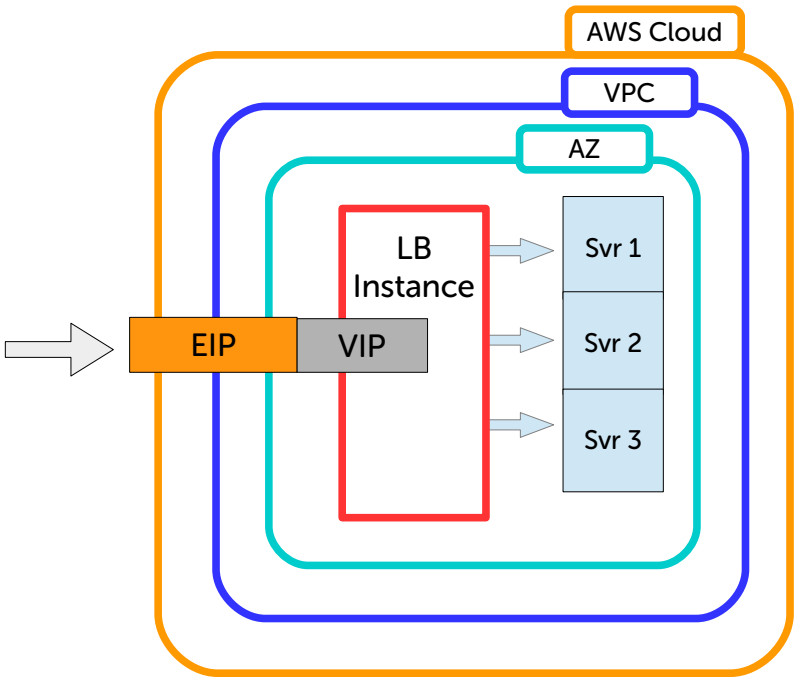
AWS TOPOLOGY OPTIONS

There are several ways in which the load balancer can be deployed. The options available depend on whether you intend to deploy one or two (for HA) load balancer instances, and whether you are deploying to single or dual availability zones. The options are explained below.

SINGLE AVAILABILITY ZONE

Single Unit

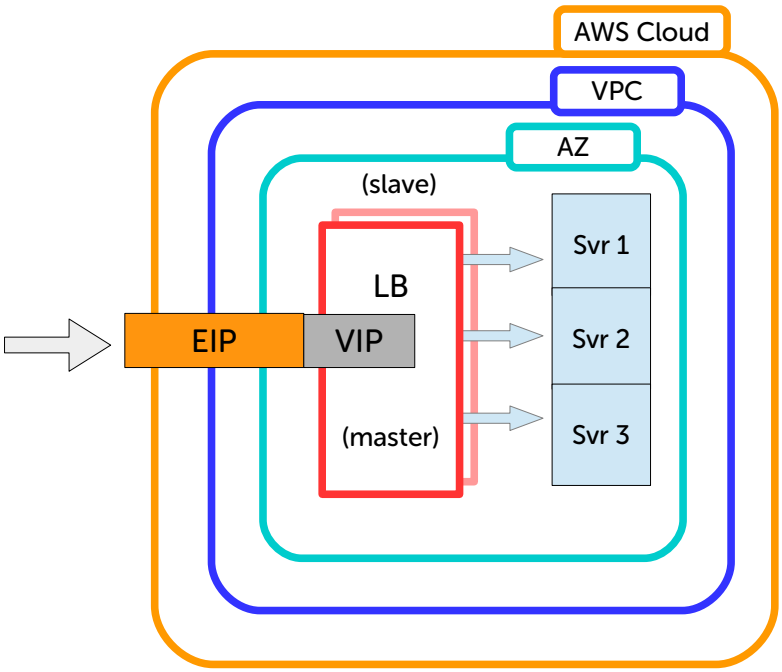
A single instance is deployed.



- If the instance fails for any reason, load balanced services will no longer be available.

2 Instances in Active/Passive mode

Here, two instances are deployed as a clustered pair. This is Loadbalancer.org's traditional HA mode where one appliance is the master and the second is the slave.



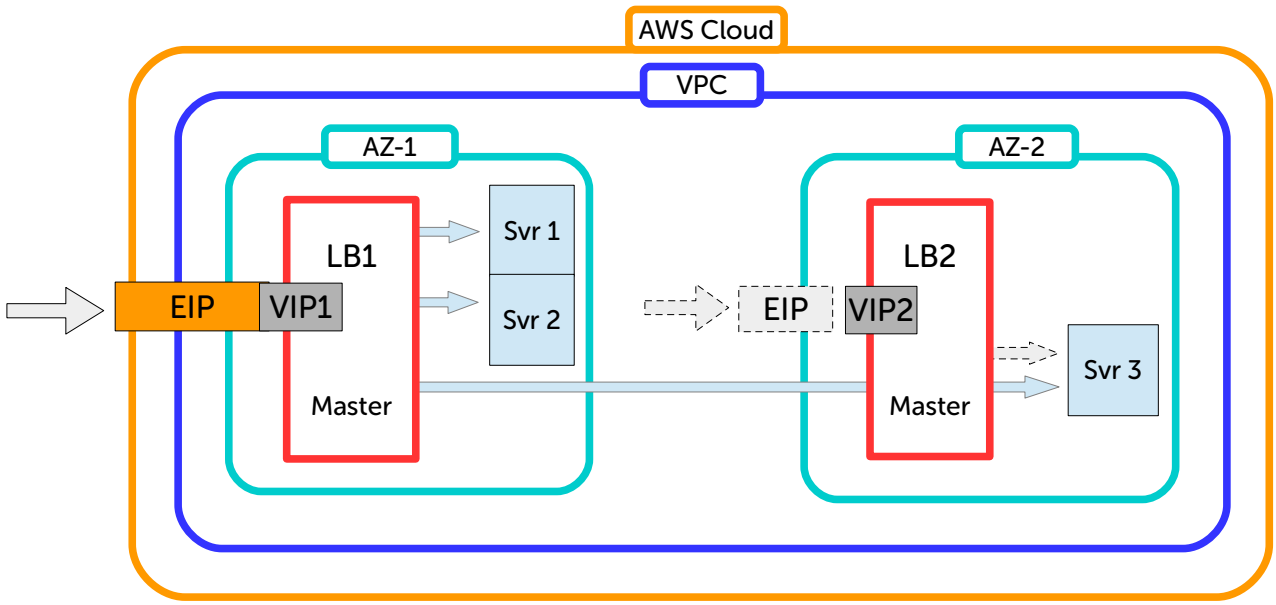
- Under normal conditions the master is active and the slave is passive. If the master fails, the load balanced services (VIPs) will be automatically brought up on the slave. When failover occurs, the EIP is still associated with the same private IP address, but it's now active on the slave
- For a correctly configured pair, changes made to load balanced services on the master will be automatically replicated to the slave
- Both master and slave appliances must be deployed in the same subnet/Availability Zone to allow VIP(s) to be brought up on either appliance
- Please refer to page 47 for detailed steps on configuring this mode

DUAL AVAILABILITY ZONES

2 Instances in AZ HA Mode

This mode enables two appliances to be configured in different subnets/Availability Zones. In this mode, the same VIP(s) are configured on both instances and are always locally active, but only one is made available via the associated EIP. Regular checks ensure that the EIP is up, and if it's not, the EIP is automatically associated with the other instance thereby ensuring availability.

There are several options regarding placement of the load balanced servers (RIPs), the example below shows one possible scenario.



- In this mode, VIPs are configured independently on both load balancer instances using a private address in the respective subnet
- Both VIP1 on LB1 and VIP2 on LB2 are locally active, but the EIP is only associated with one of the instances, in the example above the EIP is normally associated with LB1
- LB2 regularly checks that the EIP in up via LB1, and if not, the EIP is associated with LB2 after the check timeout has been reached
- The WebUI can be used to force VIP2 on LB2 to be associated with the EIP rather than VIP1 on LB1
- In the above example, should AZ-1 fail, then LB1, Svr1 & Svr2 will also go down. This will trigger LB2 to associate the EIP with VIP2/Svr3, and services will continue to be available
- Please refer to page 51 for detailed steps on configuring this mode

CREATING A VPC

The simplest way to create a VPC in AWS is to use the wizard. When using the wizard there are 4 types that can be selected as detailed in the table below:

Type	Description	Creates
VPC with a Single Public Subnet	Instances run in a private, isolated section of the AWS cloud with direct access to the Internet. Network access control lists and security groups can be used to provide strict control over inbound and outbound network traffic to your instances.	A /16 network with a /24 subnet. Public subnet instances use Elastic IPs or Public IPs to access the Internet.
VPC with Public and Private Subnets	In addition to containing a public subnet, this configuration adds a private subnet whose instances are not addressable from the Internet. Instances in the private subnet can establish outbound connections to the Internet via the	A /16 network with two /24 subnets. Public subnet instances use Elastic IPs to access the Internet. Private subnet instances access the Internet via a Network

	public subnet using Network Address Translation (NAT).	Address Translation (NAT) instance in the public subnet. (Hourly charges for NAT instances apply.)
VPC with Public and Private Subnets and Hardware VPN Access	This configuration adds an IPsec Virtual Private Network (VPN) connection between your Amazon VPC and your data center - effectively extending your data center to the cloud while also providing direct access to the Internet for public subnet instances in your Amazon VPC.	A /16 network with two /24 subnets. One subnet is directly connected to the Internet while the other subnet is connected to your corporate network via IPsec VPN tunnel. (VPN charges apply.)
VPC with a Private Subnet Only and Hardware VPN Access	Your instances run in a private, isolated section of the AWS cloud with a private subnet whose instances are not addressable from the Internet. You can connect this private subnet to your corporate data center via an IPsec Virtual Private Network (VPN) tunnel.	A /16 network with a /24 subnet and provisions an IPsec VPN tunnel between your Amazon VPC and your corporate network. (VPN charges apply.)

Note:

For more details on Amazon's VPC, please refer to their comprehensive user guide available at the following URL: http://awsdocs.s3.amazonaws.com/VPC/latest/vpc-ug.pdf.

VPC IP ADDRESS TYPES

There are 3 IP address types as detailed below:

Private

The internal RFC 1918 address of an instance that is only routable within the EC2 Cloud. Network traffic originating outside the EC2 network cannot route to this IP, and must use the Public IP or Elastic IP Address mapped to the instance.

Public

Internet routable IP address assigned by the system for all instances. Traffic routed to the Public IP is translated via 1:1 Network Address Translation (NAT) and forwarded to the Private IP address of an instance. The mapping of a Public IP to Private IP of an instance is the default launch configuration for all instance types. Public IP Addresses are released when instances are stopped or terminated. When an instance is powered on again or restarted, it is allocated a different public IP address. If you require a persistent public IP address that can be associated to and from instances as you require, use an Elastic IP address instead.

Elastic (EIP)

Internet routable IP address allocated to an AWS EC2 account. Similar to EC2 Public Address, 1:1 NAT is used to map Elastic IP Addresses with their associated Private IP addresses. Unlike a standard EC2 Public IP Address, Elastic IP Addresses are allocated to accounts and can be remapped to other instances when desired.

Note:

Virtual Services (VIPs) can be created on the same IP address as the load balancer's network interface (ENI). However, if configured in this way, it won't be possible to add a slave unit to create an HA clustered pair.

IP ADDRESS ALLOCATION OPTIONS & REQUIREMENTS

Depending on the deployment scenario, there are certain requirements & constraints that apply.

INTERNAL (PRIVATE NETWORK) DEPLOYMENTS

In this scenario **Virtual Services (VIPs)** can be configured as follows:

- **For a Single Appliance**
  - Using the primary private IP address of the instance
    OR
  - Using additional secondary private IP(s)
- **For a Clustered Pair (Master & Slave)**
  - Using additional secondary private IP(s) (to allow the VIP to 'float' between master & slave)

PUBLIC FACING DEPLOYMENTS

In this scenario **Virtual Services (VIPs)** can be configured as follows:

- **For a Single Appliance**
  - Using the primary private IP address of the instance, then associating an EIP with this address to enable public access
    OR
  - Using additional secondary private IP(s), then associating an EIP with this address to enable public access
    AND associating an additional EIP with the primary IP address on the instance – this ***must*** be done
- **For a Clustered Pair (Master & Slave) & Dual AZ (Primary & Secondary Master) Deployments**
  - Using additional secondary private IP(s) (to allow the VIP to 'float' between master & slave) then associating an EIP with this address to enable public access
    AND associating an additional EIP with the primary IP address of each instance – this ***must*** be done

> Note:
>
> In all cases, the appliance requires Internet access to be able to successfully make EC2 API calls. The EC2 API enables the appliance to automatically interact with EC2. If Internet access is not available, please note the following points:
>
> - Don't assign an IAM role, this will ensure the appliance does not attempt to make EC2 API calls
> - EIPs must be manually allocated and associated using the EC2 Console
> - Secondary IPs must be manually added using the EC2 Console
> - It won't be possible to create an HA clustered pair (master & slave)

> Note:
>
> Provided that Internet access is available, secondary IP addresses will be added automatically to the AWS instance when VIPs are added on any valid IP (other than the primary IP) using the Enterprise AWS appliance WebUI.

> Note:
>
> For an HA pair (2 load balancer instances), you'll need 3 EIPS's – 1 for the primary interface on each instance, and 1 for the VIP. For more details on configuring an HA pair, please refer to page 47.

VPC NETWORK INTERFACES (ENI)

By default, a single ENI (Elastic Network Interface) is allocated when an instance is launched. A private IP address within the IP address range of its VPC is auto assigned to the ENI. Multiple private IP addresses can be assigned to each ENI, the limit is determined by instance type as defined at the following link:

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html#AvailableIpPerENI

10GB SUPPORT

For the load balancer to support 10GB, SR-IOV (single root I/O virtualization) must be enabled. This can be done with one of following commands. The instance needs to be stopped to run the command. When using instances with enhanced networking they should be placed in the same placement group.

modify-instance-attribute (AWS CLI)

```
$ aws ec2 modify-instance-attribute --instance-id instance_id --sriov-net-support simple
```

ec2-modify-instance-attribute (Amazon EC2 CLI)

```
$ ec2-modify-instance-attribute instance_id --sriov simple
```

These commands can be run from any machine that has the AWS or ec2 tools installed and security access configured. Once enabled the load balancer supports 6.5GB/s at layer 7 and 9GB/s at layer 4.

INSTANCE TYPE

When deploying a new instance, the default type is t2.medium. This can be changed as required. Please refer to the following URL for a quick comparison of the various types available:

http://www.ec2instances.info/

6. Deploying Enterprise AWS

CREATE & CONFIGURE A VPC

For a manually created VPC, the key steps are:

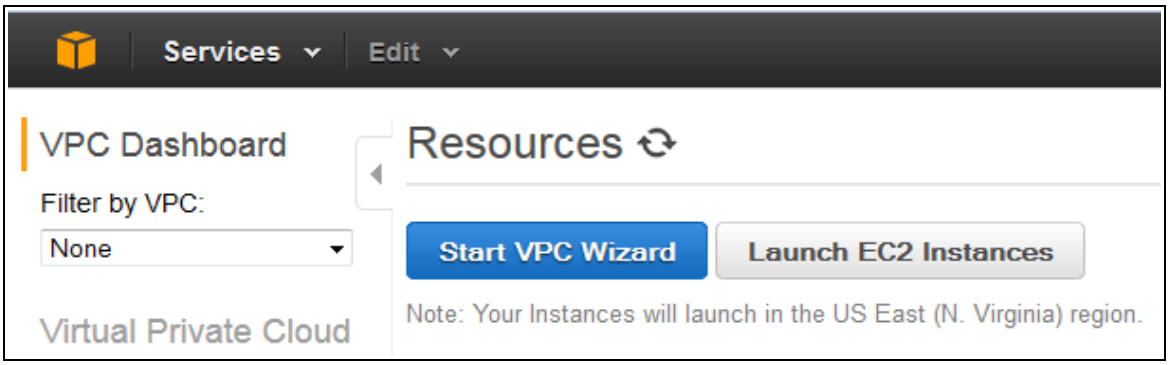
1. Create a VPC - this is an isolated portion of the AWS cloud
2. Create and attach an Internet gateway - this connects the VPC directly to the Internet and provides access to other AWS products

3. Create an Amazon VPC subnet - this is a segment of a VPC's IP address range that you can launch Amazon EC2 instances into
4. Set up routing in the VPC - this enables traffic to flow between the subnet and the Internet
5. Set Up a Security Group for the VPC - this controls the inbound and outbound traffic

However, as mentioned previously the easiest way to configure a VPC is by using the *VPC Wizard*. The wizard covers steps 1-4.

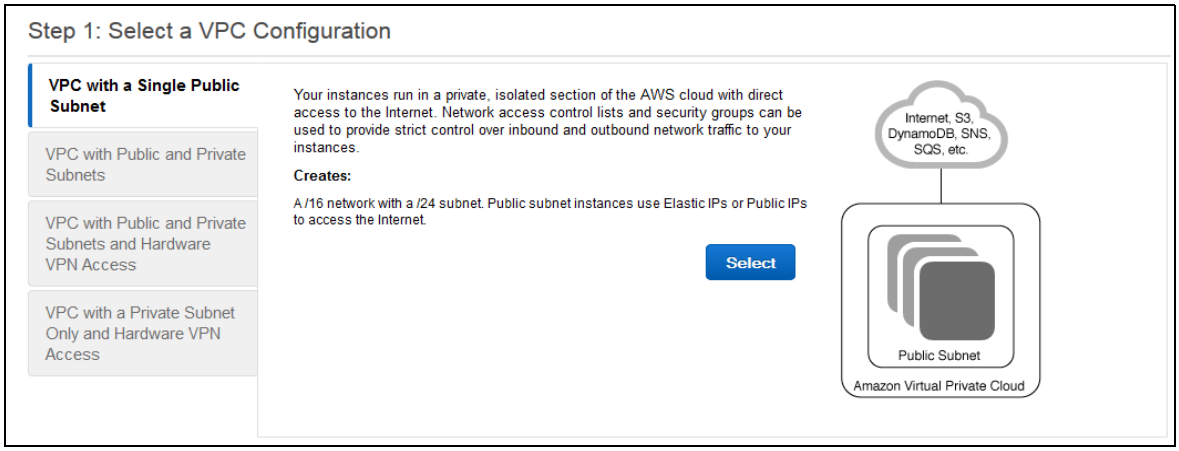
To create a VPC using the wizard:

1. In the VPC dashboard, click **Start VPC Wizard**

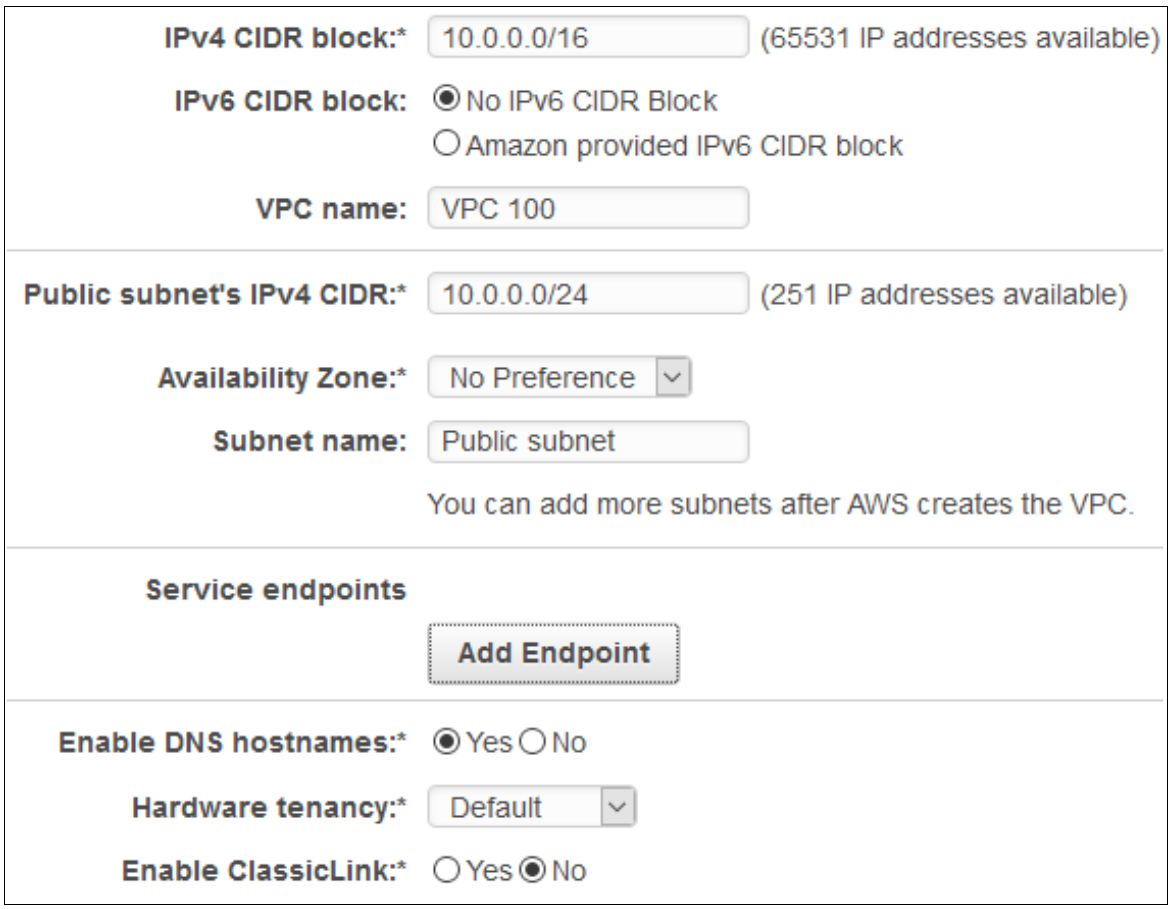


2. Select the first option – ***VPC with a Single Public Subnet***

> Note:
>
> This wizard option is appropriate in most cases. It creates a VPC with a single public subnet and auto configures the gateway, subnets and routing table. Additional subnets can be added later if required.



3. Enter a VPC name and modify the other settings as required as show in the example below:



4. Click **Create VPC**

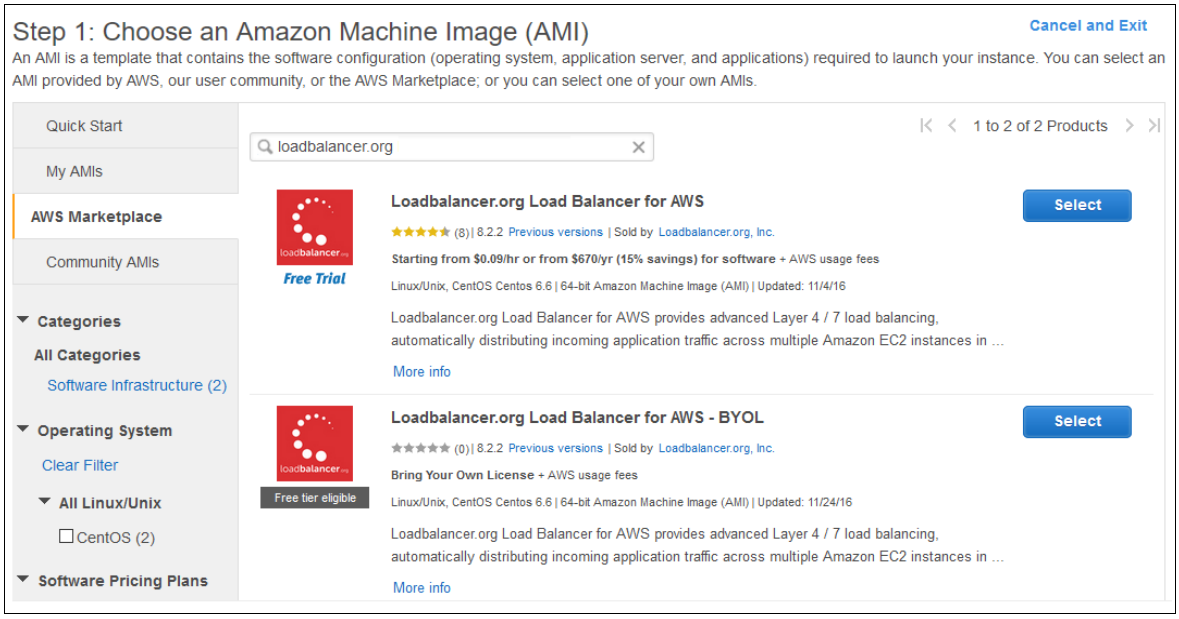
> Note:
>
> For more details on Amazon's VPC, please refer to this URL.

ACCESSING & DEPLOYING THE AMI

> Note:
>
> Configuring an **IAM role** for the instance is optional. However, we always recommend that one is assigned. This allows the instance to make AWS API calls to automatically configure the required AWS settings. If not set, these AWS settings would need to be manually configured. For more details please refer to section 1 in the Appendix on page 58.

To access and deploy the AMI:

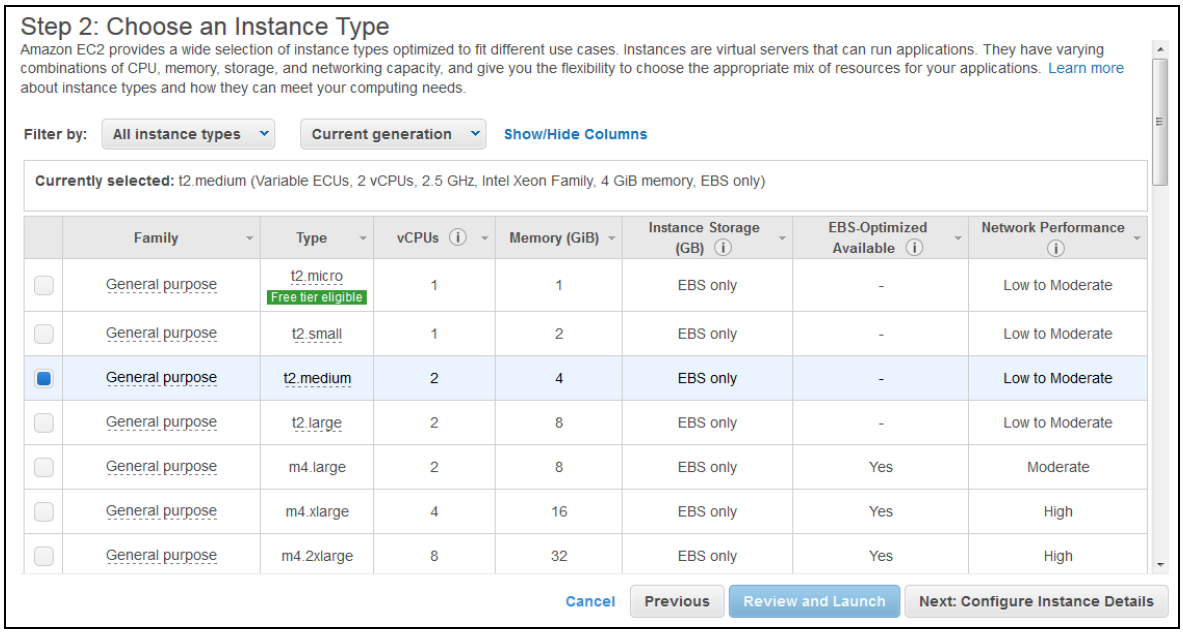
1. In the EC2 Dashboard, click **Launch Instance**
2. Select *AWS Marketplace*
3. Search for "Loadbalancer.org"



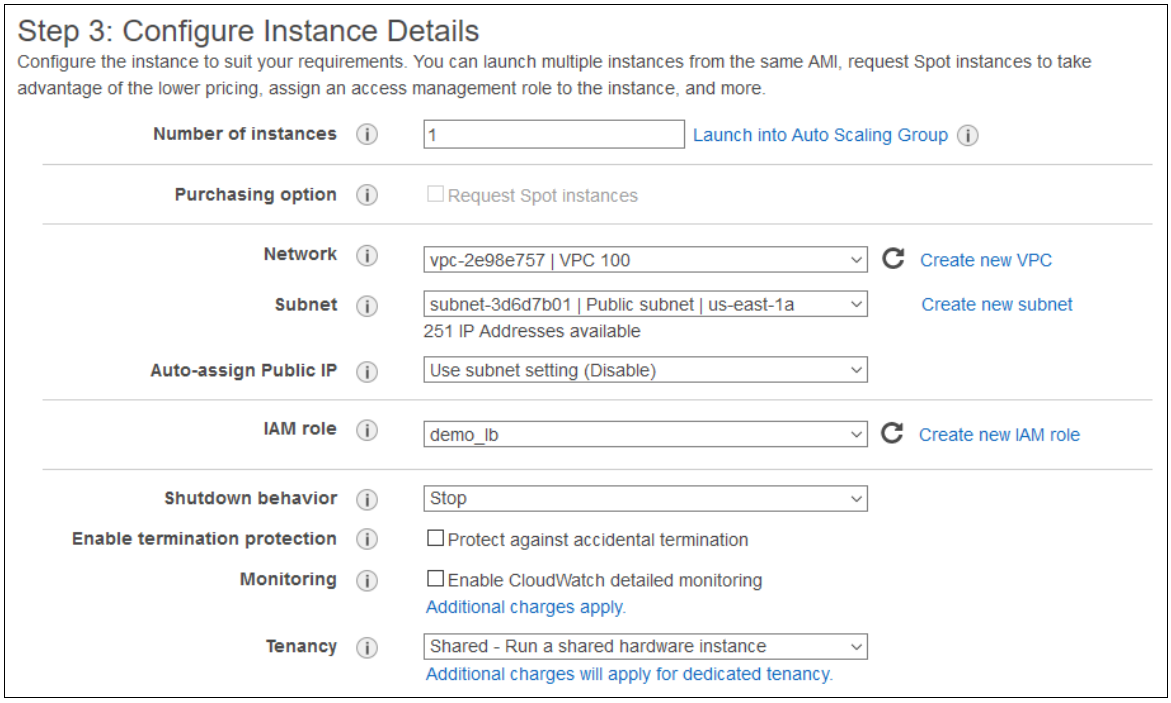
4. Click **Select** next to the required AMI, either:

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

5. Review pricing details and if happy to proceed click **Continue**
6. Select the required instance type – **t2.medium** is the default



7. Click **Next: Configure Instance Details**



8. Change **Network** to the required VPC

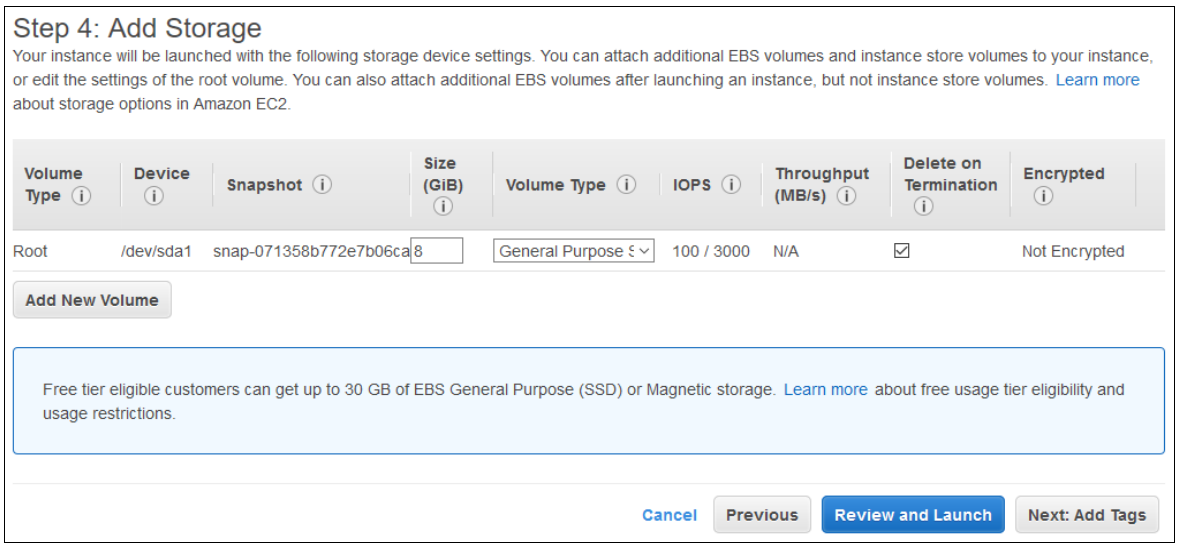
   - If the VPC was created with the wizard, the public subnet's auto-assign Public IP option will be disabled. To automatically allocate a public IP address, change **Auto-assign Public IP** to "Enable"

9. Select a suitable **IAM Role**. The role can simply have "***Amazon EC2 Full Access***" for the "***Amazon EC2***" AWS Service Role or for more granular configuration please refer to page 58 in the appendix

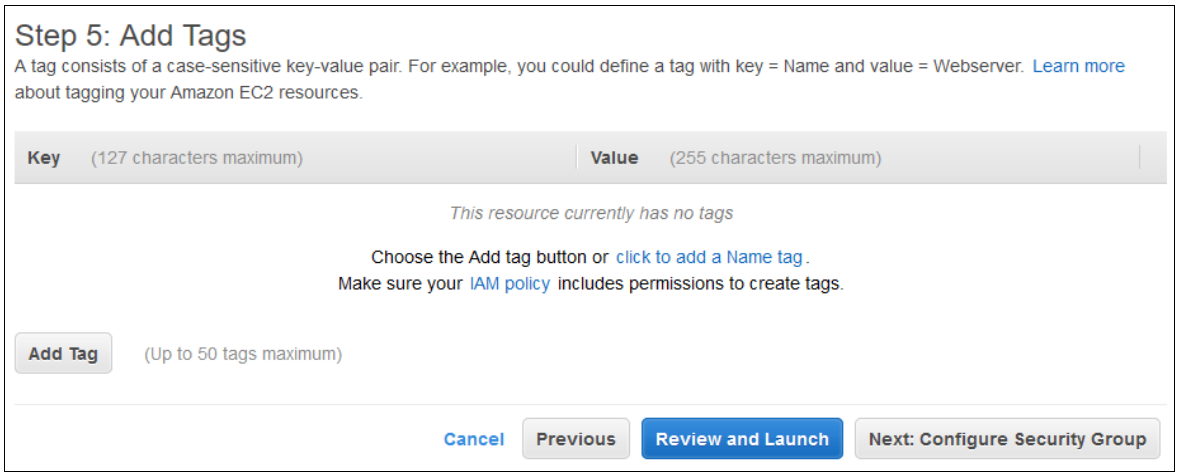
> Note:
>
> Configuring an **IAM role** for the instance is optional. However, we always recommend that one is assigned. This allows the instance to make AWS API calls to automatically configure the required AWS settings. If not set, these AWS settings would need to be manually configured. For more details please refer to section 1 in the Appendix on page 58.

Network Interfaces - typically there is no need to add additional interfaces. Load balancing real servers in different subnets is configured by changing AWS routing rules. The routing rules required depend on where the real servers and located (same or different subnet as the load balancer) and the load balancing mode selected. Please refer to the configuration examples 4 & 7 on pages 37 and 44 respectively for more details.

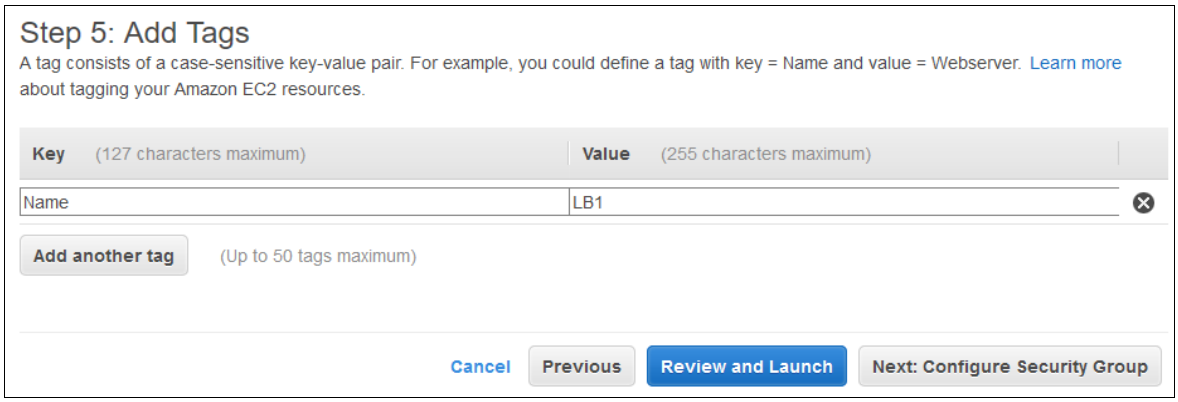
10. Configure the remaining options according to your requirements
11. Click **Next: Add Storage**



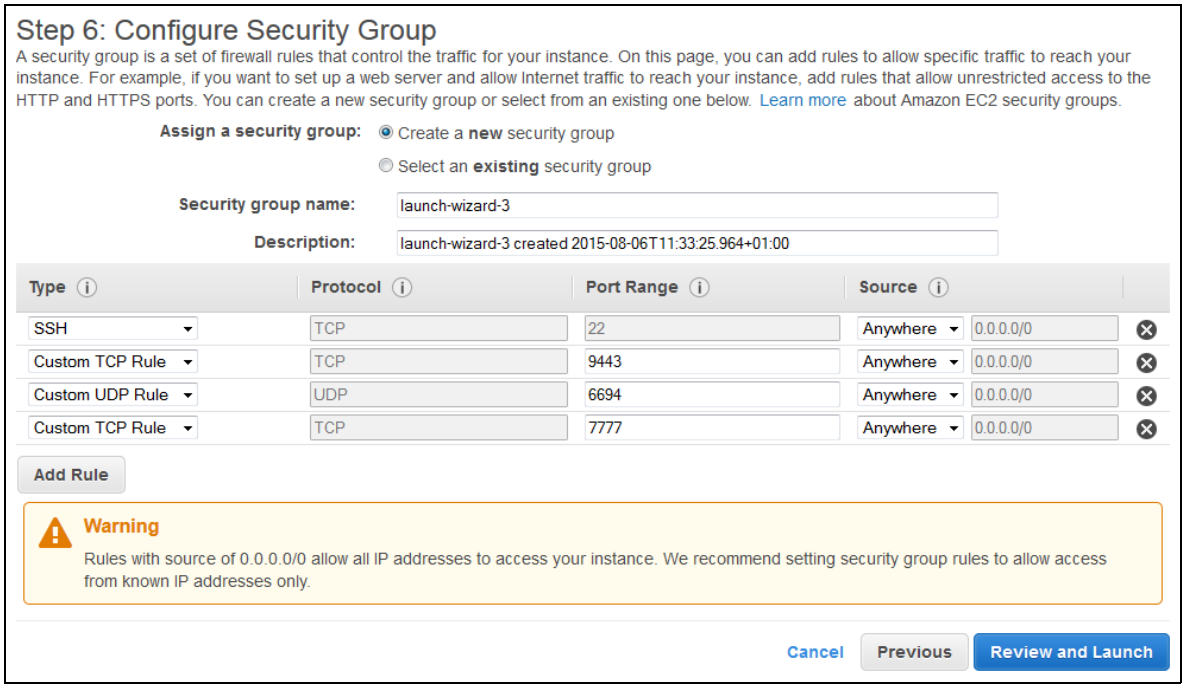
12. Set the required options – the defaults are appropriate in most cases, click **Next: Add Tag**



13. Define the required tags for the instance. For example, to define a tag with key = *Name* and value = *LB1*, click **Add Tag** and enter the values as shown below:



14. Click **Next: Configure Security Group**



- At least the rules shown above and listed below must be configured. These are required to enable management & monitoring access to the load balancer.

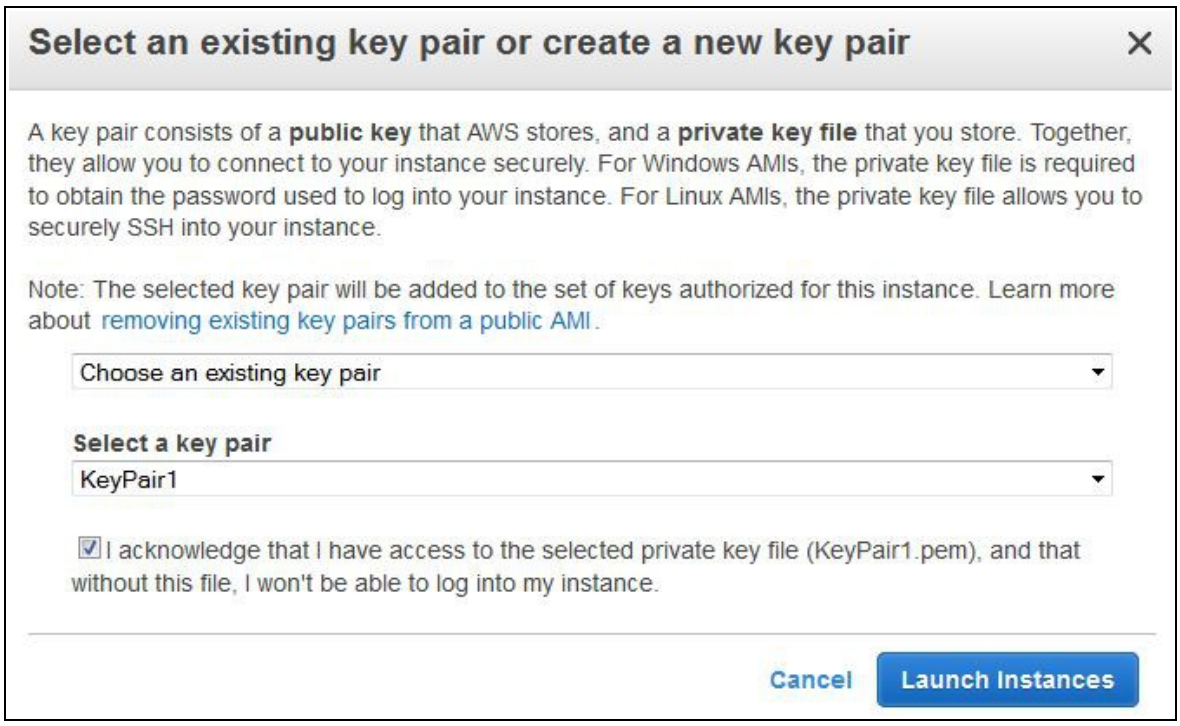
  Management (SSH) - TCP port 22
  Management (WebUI) - TCP port 9443
  Heartbeat between master and slave appliances - UDP port 6694
  Monitoring (HAProxy Statistics Page) - TCP port 7777

> Note:
>
> By default, rules with source of 0.0.0.0/0 allow all IP addresses access to the instance. For the management and monitoring addresses shown above, these should be locked down to allow access only from known / trusted IP's.

- Additional rules must be added to provide access to the application(s) being load balanced. These should also be locked down to know IPs / IP ranges where possible.

  e.g. If you're load balancing HTTP & HTTPS traffic, add TCP ports 80 & 443
  e.g. If you're load balancing RDP traffic, add TCP port 3389
  e.g. If you're load balancing SIP traffic, add TCP/UDP ports 5060/5061 (the exact ports required for SIP depend on the specific VoIP system being load balanced)
  etc.

15. Click **Review and Launch**
16. Check all settings and click **Launch**

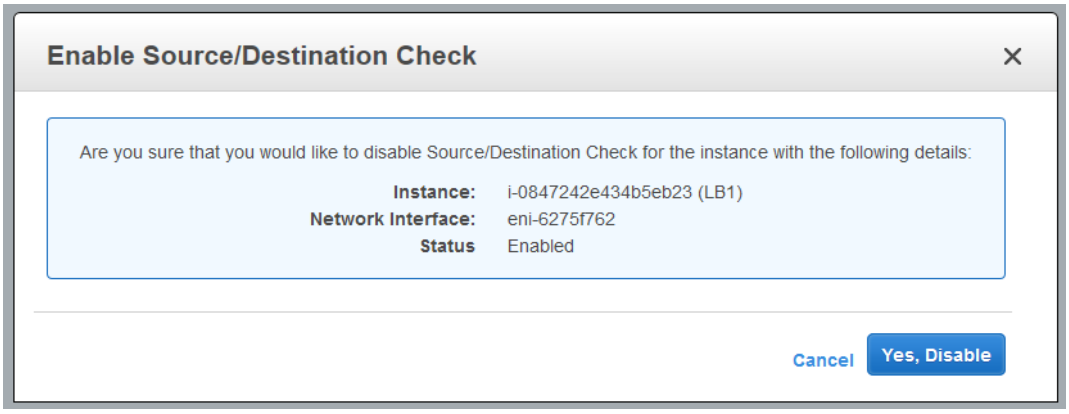


17. If creating a new pair use the **Download Key Pair** button to save the private key

> Note:
>
> This private key is used for secure access to the load balancer instance via SSH once it's up and running.

18. If using an existing key pair, check (tick) the acknowledgment check-box
19. Click the **Launch Instances** button, the instance will now launch

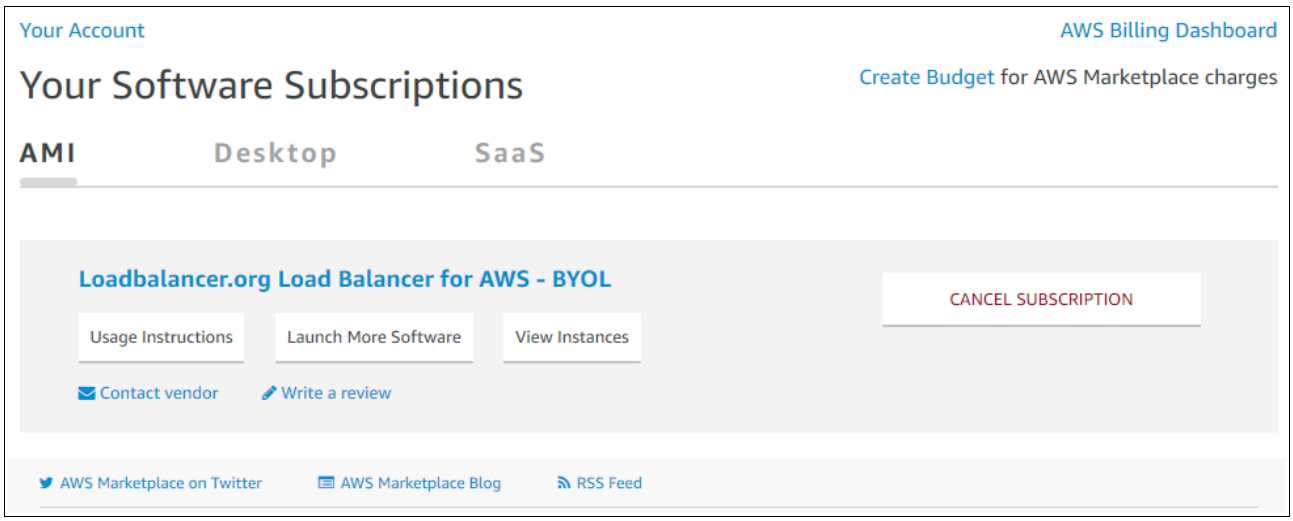
> Important:
>
> Once the instance is running, right-click the instance and select: ***Networking > Change Source/Dest. Check*** and ensure this is disabled by clicking the **Yes, Disable** button



> This **MUST** be done to enable the load balancer to function correctly.

CHECKING YOUR SUBSCRIPTIONS

Current subscriptions can be viewed and canceled using the *Your Marketplace Software* option in the awsmarketplace console as shown below:



7. Accessing the Appliance

USING THE WEBUI

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

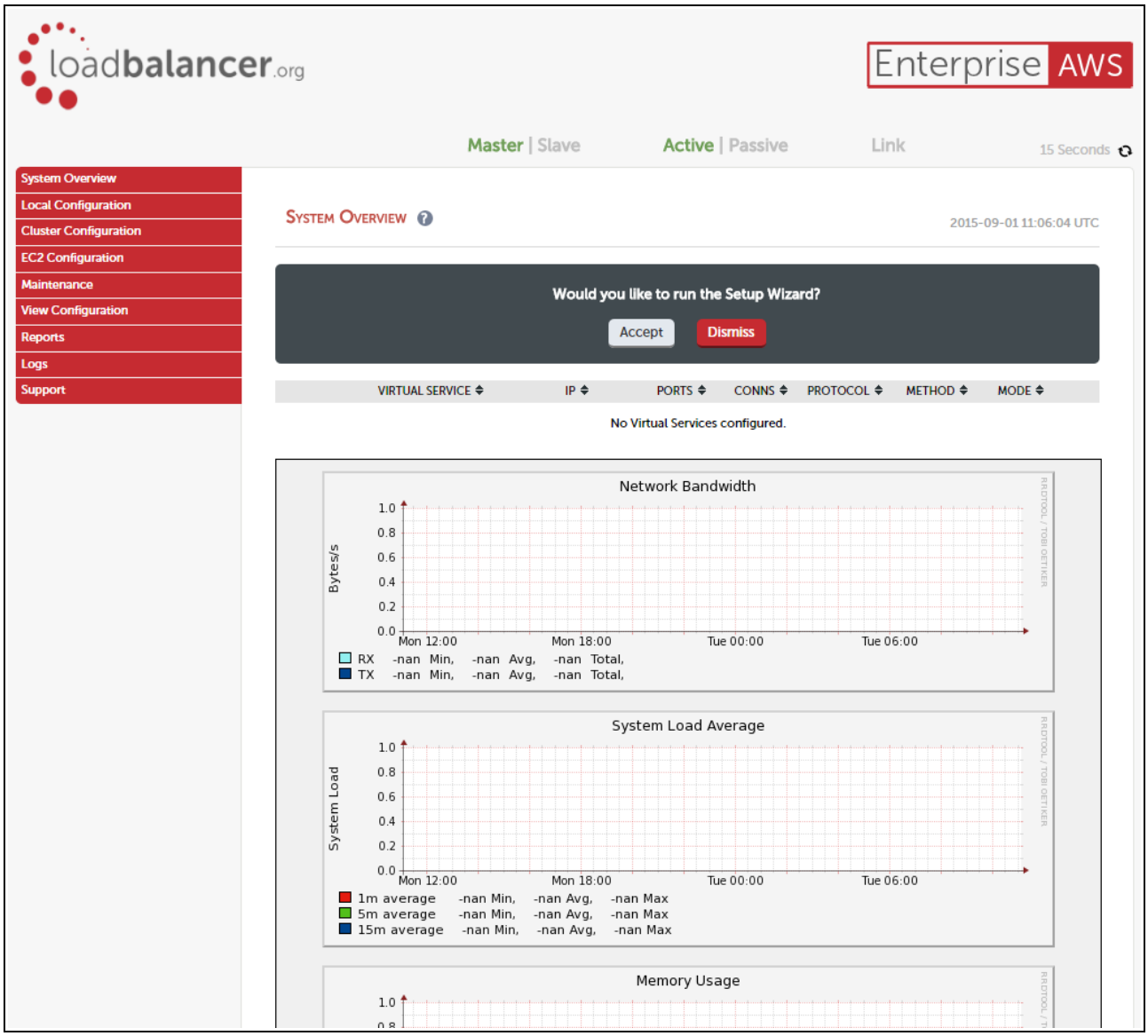
https://<Public DNS name>:9443
or
https://<Public IP address>:9443

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: <EC2 Instance-ID>

> 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
EC2 Configuration – Configure Elastic IP to local IP associations & dual AZ HA settings
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

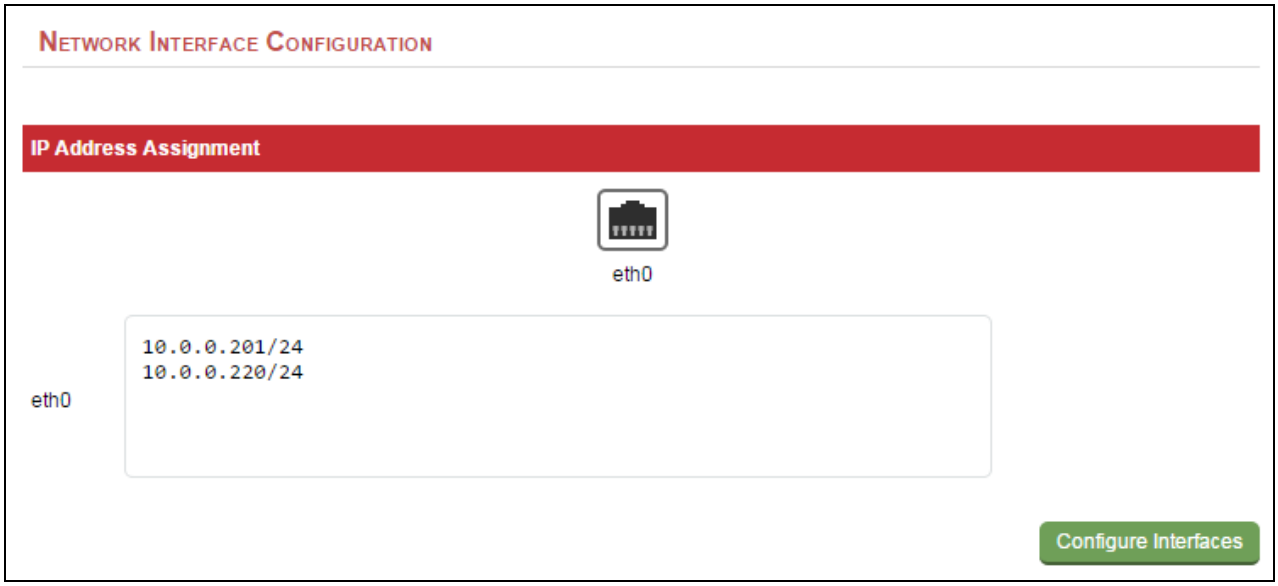
If you've deployed the BYOL version of the appliance, by default it runs as a 30 day trial and is completely unrestricted during this time. After 30 days, the appliance continues to work but it's no longer possible to make changes to the configuration. When a license is purchased, you'll be provided 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 AWS NON-STANDARD WEBUI MENU OPTIONS

Enterprise AWS has a number of differences to the standard hardware/virtual product range due to the way the Amazon EC2 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

1) Local Configuration > Network Interface Configuration



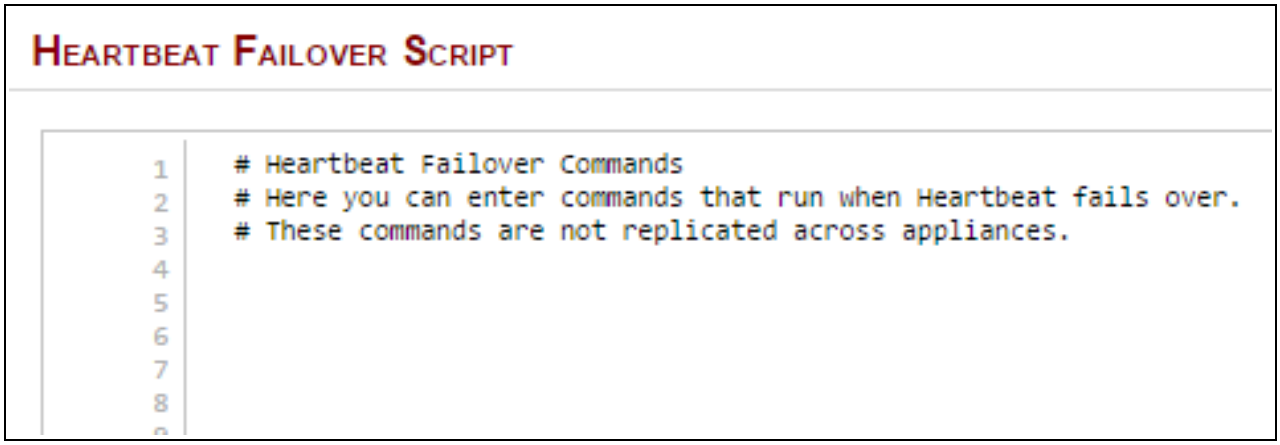
Notes:

- Shows the private IP addresses allocated to the instance
- The first address in the list is auto-allocated when launched

> Note:
>
> It's not possible to change the auto-allocated IP addres

- Multiple IP addresses can be assigned as shown
- Additional IP addresses added here after the first one in the list are shown as "Secondary Private IP's" in the AWS/EC2 Dashboard
- Click **Configure Interfaces** to apply any changes

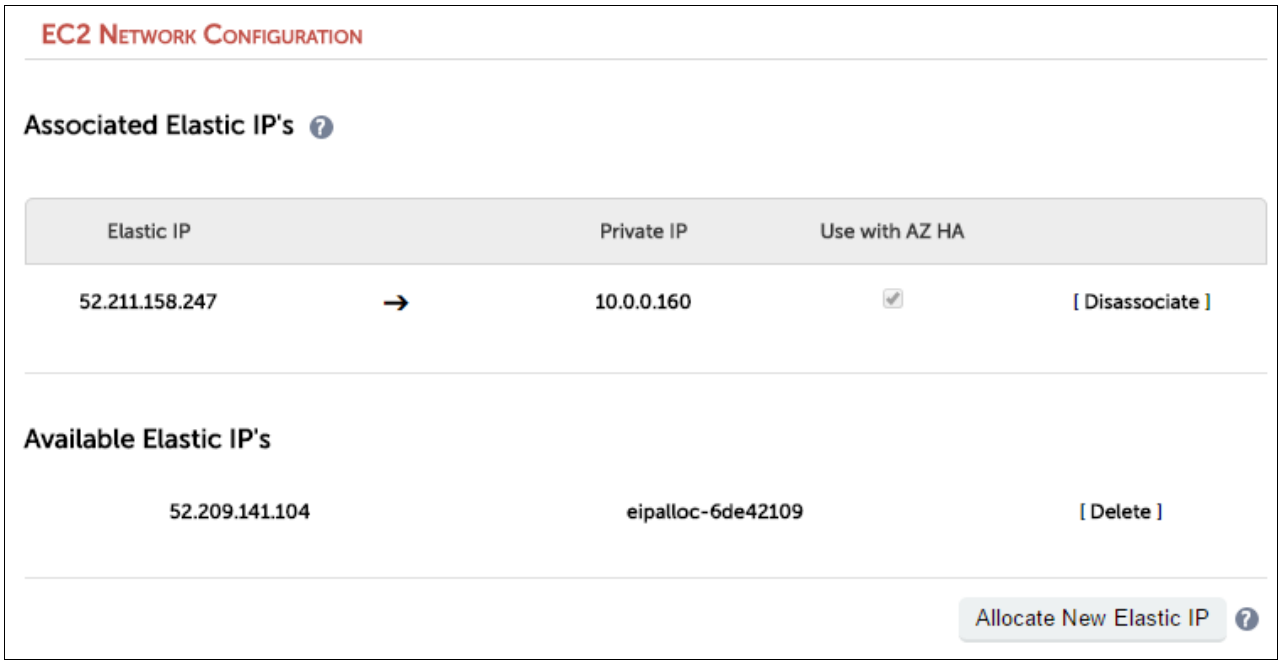
2) Cluster Configuration > Heartbeat Advanced



Notes:

- Enables commands to be run at failover from master to slave appliance if configured. This includes Amazon CLI tools commands. For more information of the various CLI commands available please refer to this AWS link
- Please refer to page 47 for more details on configuring 2 appliances in a master/slave HA configuration

3) EC2 Configuration > EC2 Network Configuration



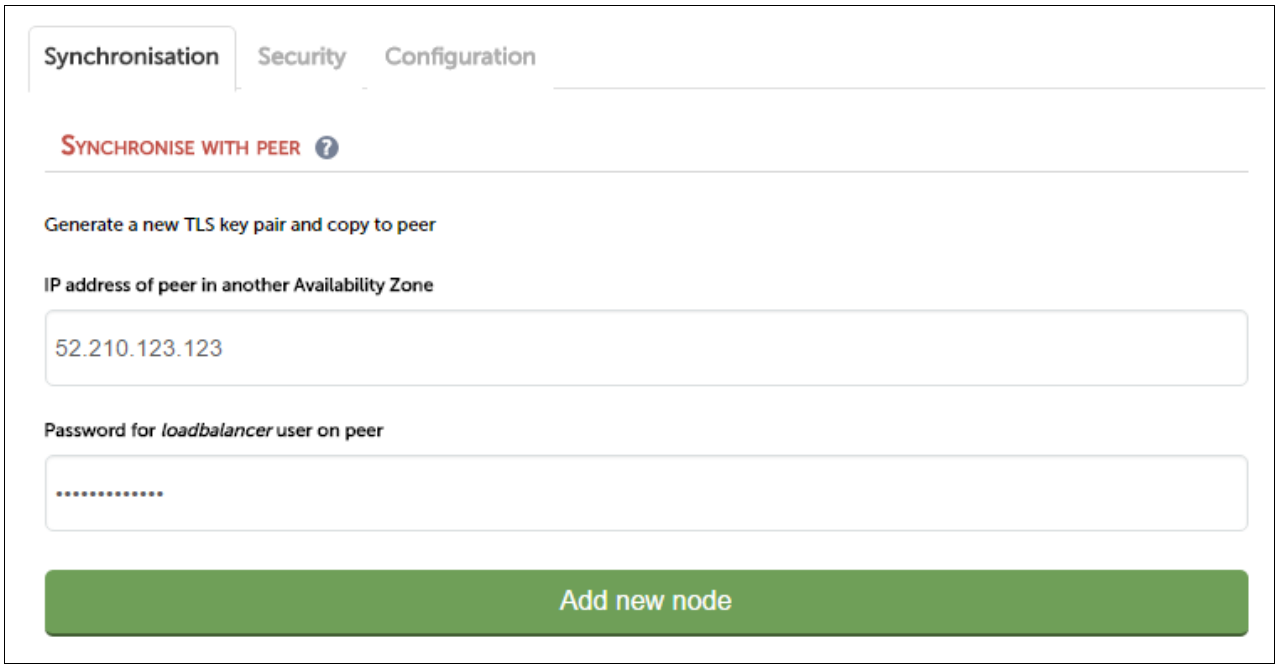
Notes:

- This menu option is used to define how Elastic IP's relate to private IP's
- Row-1 above shows that EIP 52.211.158.247 is associated with private IP 10.0.0.160. If you want to undo the association click **[Disassociate]**
- Row-2 above shows that EIP 52.209.141.104 is currently not associated with any Private IPs, it can be deleted by clicking **[Delete]**
- New EIP's can be allocated by clicking **Allocate New Elastic IP**. Newly created EIP's will be displayed in the ***Available Elastic IP's*** list. New addresses will also be displayed in the AWS console. Similarly, if new EIP's are created in the AWS console, they will be displayed here
- Please refer to page 51 for more details on configuring dual AZ HA mode

4) EC2 Configuration > EC2 Zone HA Configuration

This menu option is used to configure Zone HA , where 2 master instances are deployed, each in a different AZ in a primary/secondary configuration. Please refer to page 51 for more details on configuring dual AZ HA mode

Synchronization Tab



Notes:

- This is used to configure a primary/secondary pair. The IP address if the secondary instance and the password for the *loadbalancer* user must be entered, then when **Add new node** is clicked, new keys and signed certificates will be generated and and synchronized with the node specified. These keys are used to verify the peer when monitoring an Elastic IP across Availability Zones.
- Please refer to page 51 for more details on configuring dual AZ HA mode

Security Tab

Notes:

- This is used to verify that the various keys & certificates have been generated and also allows them to be deleted
- If deleted, the keys & certificates will need to be re-generated using the Synchronisation Tab as described above
- Please refer to page 51 for more details on configuring dual AZ HA mode

Configuration Tab

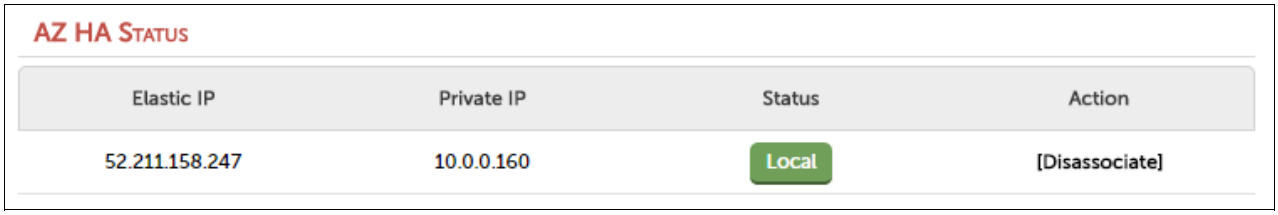
Notes:

- *Port* - This is the port the service will listen on and connect to on the peer. The appliances in each Availability Zone should use the same port
- *Check Interval* - This is the interval between health checks. It also sets the timeout value for when a health check is considered failed
- *Failure Count* - This sets the desired number of health check failures before moving the Elastic IP address. The recommended value is 3 as this helps rule out temporary issues
- *Max Association Retry* - This sets the desired number times to retry associating the elastic IP with the private IP address before giving up. Each association after the 200th association costs $0.10.

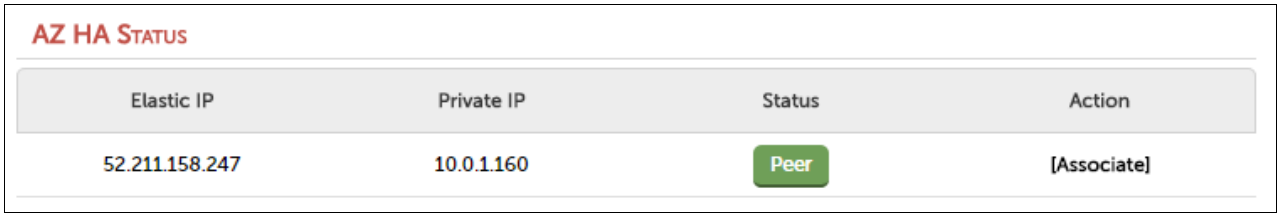
5) EC2 Configuration > EC2 Zone HA Status

This menu option is used to display the Zone HA status.

The primary master instance:



The secondary master instance:



Notes:

- The VIP (10.0.0.160) on the primary master instance is currently associated with the EIP
- Please refer to page 51 for more details on configuring dual AZ HA mode

ACCESSING THE APPLIANCE USING SSH

To access the appliance using SSH, the private key from the key pair that was selected when the instance was launched must be used. Under Linux, the key can be used immediately, for PuTTY under Windows, the key must first be converted to a format required by PuTTY as detailed below.

> Note:
>
> For SSH access make sure that TCP port 22 is included in the security group for the load balancer.

USING LINUX

First change the permission of the private key file to allow only the owner read access:

```
# chmod 400 /path-where-saved/private-key-file.pem
```

Now start SSH specifying the private key file, login as '**lbuser**'

e.g.

Using the IP address:

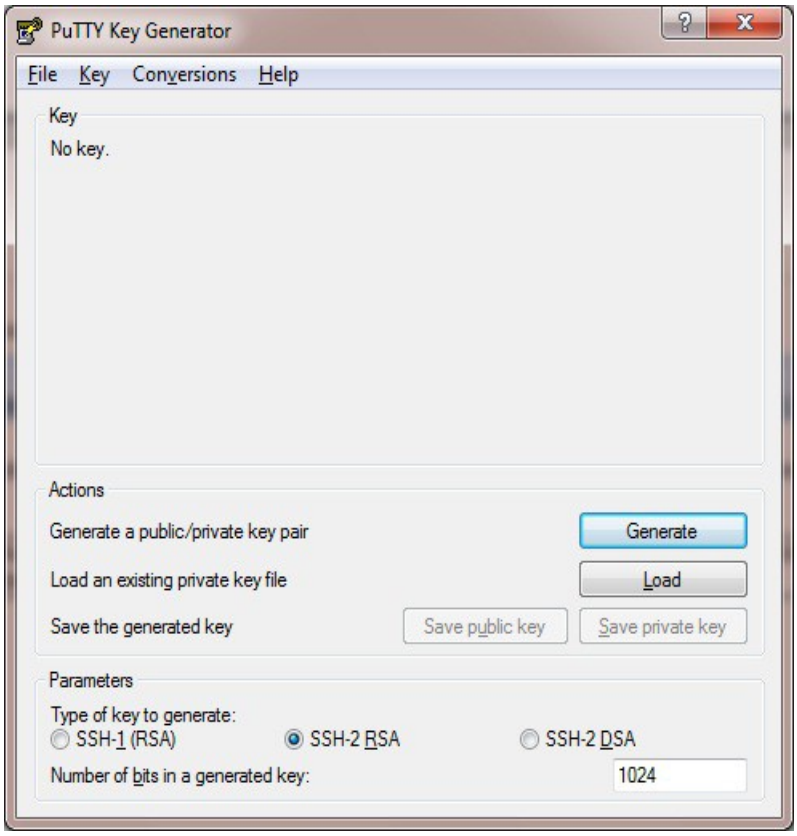
```
# ssh -i /path-where-saved/private-key-file.pem lbuser@1.2.3.4
```

Or using the fqdn:

```
# ssh -i /path-where-saved/private-key-file.pem lbuser@fqdn
```

USING WINDOWS

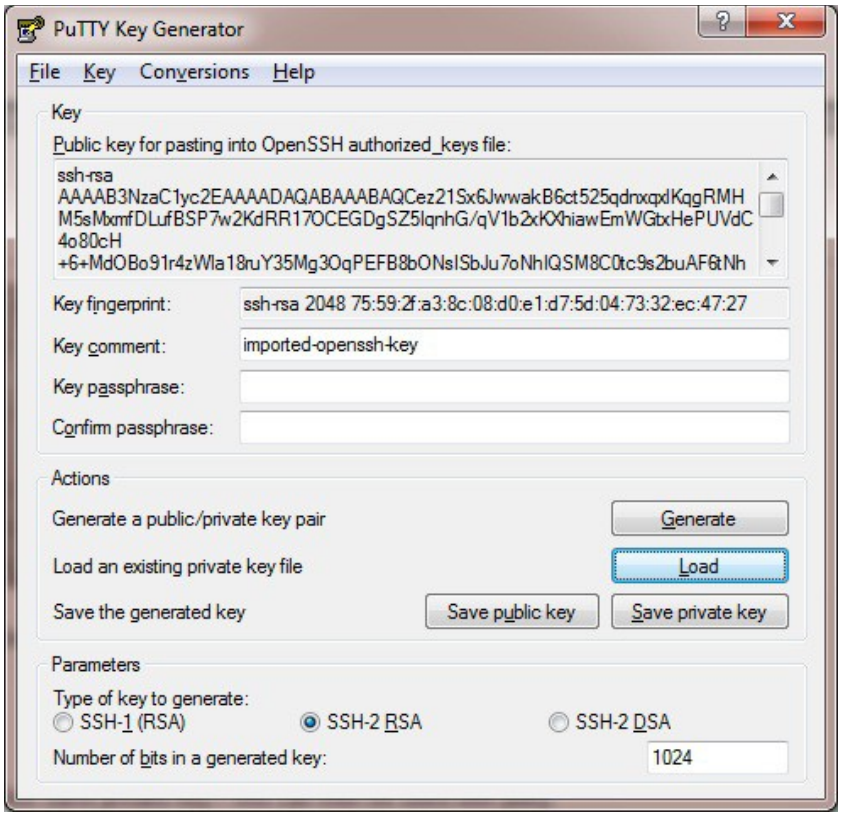
For PuTTY, the private key must be converted into an appropriate format. To do this the PuTTYgen utility (included with PuTTY) must be used. Start PuTTYgen:



Click **Load**, change the file-type to all files and select the pem file saved earlier when creating your Key Pair.

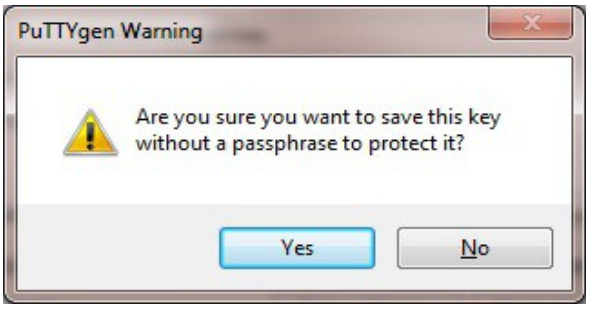
You should see the following message:

Click **OK**



Now Click **Save private key** – this can then be used with PuTTY.

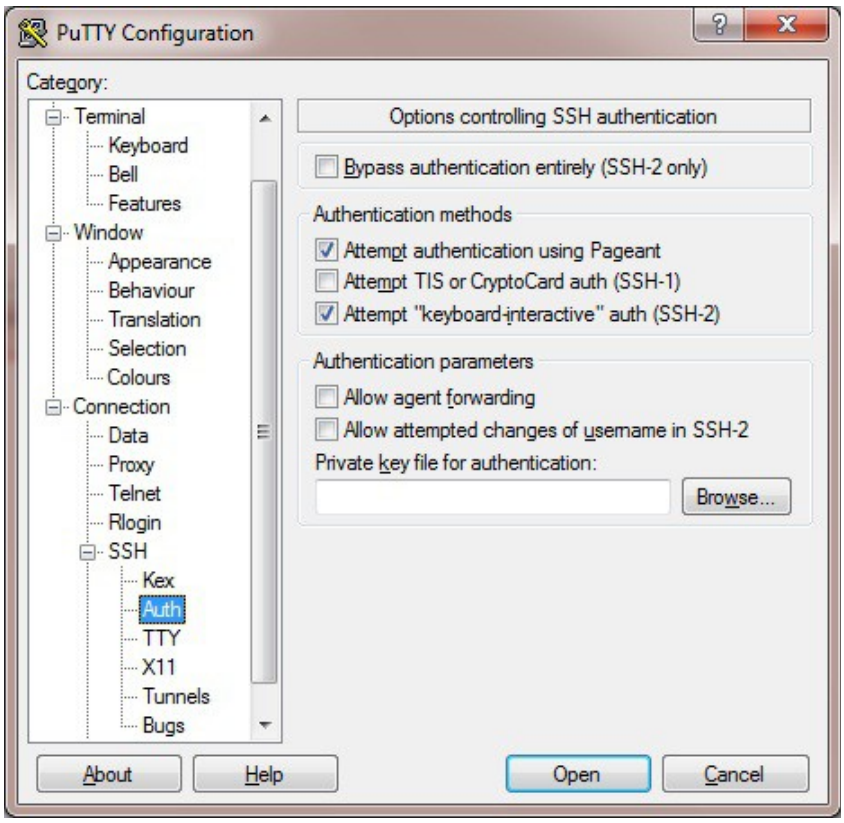
You can also choose to enter an additional pass-phrase for improved security, if you don't, the following message will be displayed:



Click **Yes** and save the file with the default .ppk extension

Now close PuTTYgen and start PuTTY

Expand the SSH section as shown below:



Click **Browse** and select the new .ppk file just created

When you open the SSH session, login as '**lbuser**' – no password will be required.

8. Configuration Examples

The following sections provide a number of examples to help illustrate how the load balancer can be deployed. In many cases, either example 1 or example 2 can be used. Both of these examples use a single subnet for the load balancer and the load balanced back-end (real) servers. The simplest is example 1 which uses a layer 7 configuration and requires no changes to the back-end servers. Example 2 uses a layer 4 configuration and requires the default gateway of the back-end servers to be the load balancer.

It's important to consider that when configured at layer 7, the load balancer is not transparent which means that the source IP address of packets reaching the real servers will be the load balancer's own IP address. At layer 4, the load balancer is transparent which means that the source IP address of packets reaching the real servers is the client IP address.

Examples 3 – 7 illustrate how the load balancer can be configured to support other scenarios, e.g. when the real servers are located in a different subnet.

DEPLOYMENT NOTES

IP Addresses

If VIPs are configured on the same IP address as any of the network interfaces, it won't be possible to add a slave unit to create an HA clustered pair.

If EIP(s) are associated with secondary IP's, you **MUST** also associate an EIP with the primary IP on the instance. Please refer to page 10 for more details.

Availability Zones

Load balanced real servers can be located in any availability zone within the region. For servers that are located in a different zone to the load balancer, simply ensure that the routing of the associated subnet is modified to include a default route (0.0.0.0/0) who's target is set to be the ENI on the load balancer. This is exactly the same approach for servers that are located in different subnets within the same zone. Please refer to configuration examples 4 & 6 on pages 37 and 42 respectively for details on setting this up.

From v8.2.2 it's also possible to place one load balancer instance in AZ-1 and a second instance in AZ-2, then create a Primary/Secondary pair HA pair. Please refer to pages 7 and 51 for more information.

Routing Table Target Configuration

To be able to set the load balancer's ENI as a route target, make sure that the ***Source/Destination Check*** is disabled for the load balancer instance. If this is not disabled, the load balancer's ENI will not be displayed as an option in the target list.

Real Server Internet access via the Load Balancer Instance

If your real servers are located in private subnets behind the load balancer and need Internet access for software installation, updates etc. , this can be achieved by enabling *autonat* on the load balancer.

To enable autonat:

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 4 – Advanced Configuration > Auto-NAT*
2. Set *Auto-NAT* to **eth0**, i.e. the load balancer's ENI

1 - WEB SERVERS – 1 SUBNET, 1 LOAD BALANCER NETWORK INTERFACE, LAYER 7

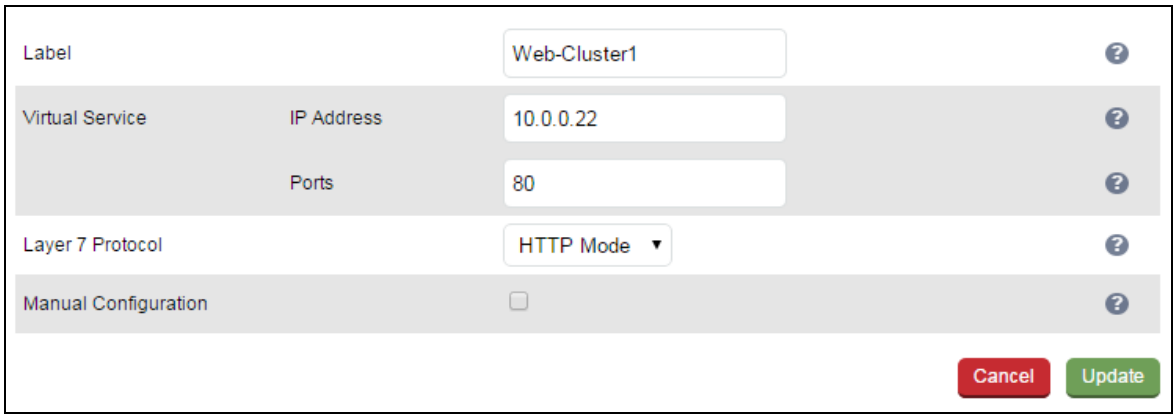
This is a simple layer 7 example using one subnet for both the load balancer and the web servers. The load balancer has a single network interface.

a) Setting up AWS

1. Deploy the load balancer instance as described starting on page 11
2. Deploy your required web server instances into the same VPC & subnet as the load balancer

b) Setting up the Virtual Service

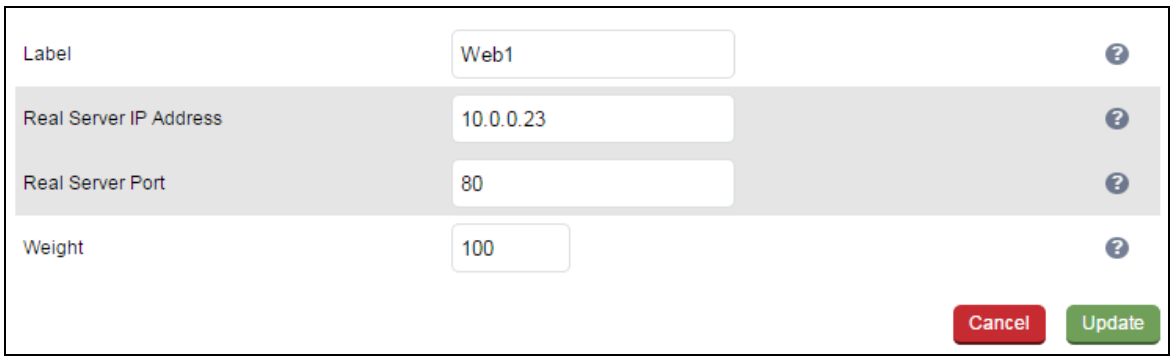
1. Using the WebUI, navigate to: *Cluster Configuration > Layer 7 – Virtual Service* and click **Add a New Virtual Service**
2. Enter the following details:



3. Enter an appropriate label for the VIP, e.g. **Web-Cluster1**
4. Set the *Virtual Service IP address* field to the required IP address, e.g. **10.0.0.22**
5. Set the *Virtual Service Ports* field to the required port, e.g. **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:



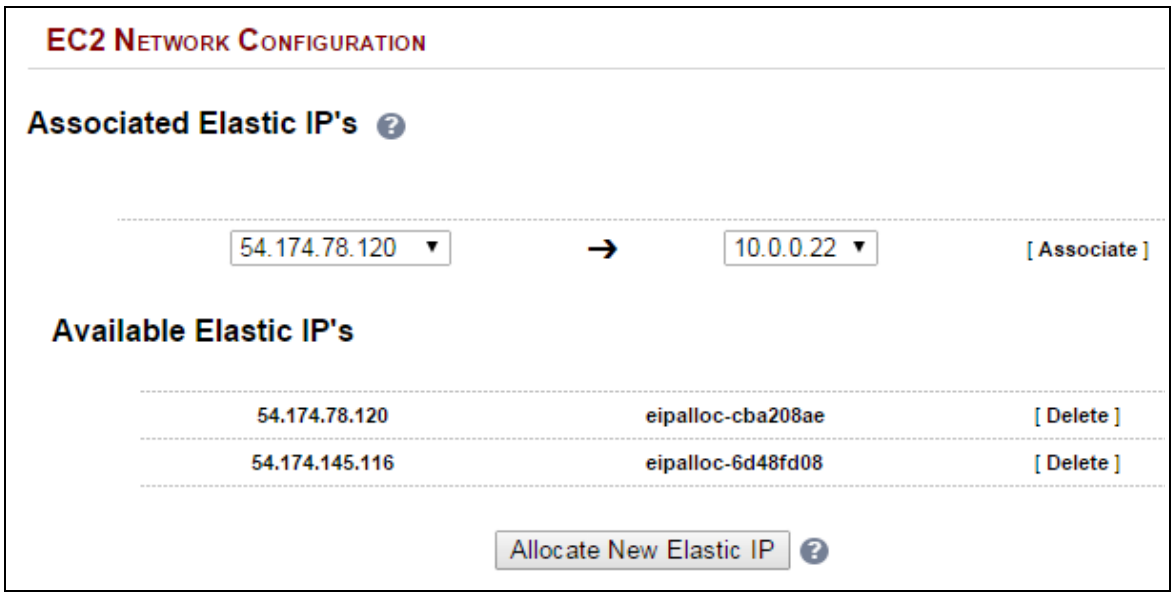
3. Enter an appropriate label for the RIP, e.g. **Web1**
4. Set the *Real Server IP Address* field to the required IP address, e.g. **10.0.0.23**
5. Set the *Real Server Port* field to the required port, e.g. **80**
6. Click **Update**
7. Repeat the above steps to add your other web server(s)

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 apply the changes

e) Associating the VIP with an Elastic IP Address (If access from the Internet is required)

1. Using the WebUI, navigate to: *EC2 Configuration > EC2 Network Configuration*



2. Under the Associated Elastic IP's section click **[Associate]** next to the VIPs private IP address (10.0.0.22 in this case), if no Elastic IP's are available, use the **Allocate New Elastic IP** button to add one

> Note:
>
> If EIP(s) are associated with secondary IP's, you **MUST** also associate an EIP with the primary IP on the instance. Please refer to page 10 for more details.

2 - WEB SERVERS - 1 SUBNET, 1 LOAD BALANCER NETWORK INTERFACE, LAYER 4

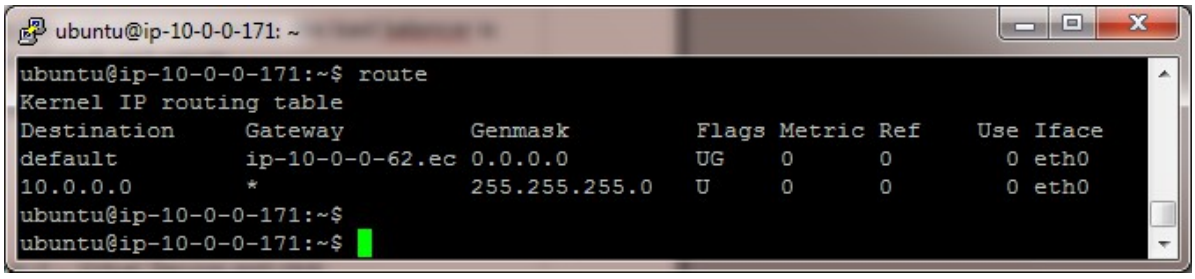
This is a simple layer 4 example using one subnet for both the load balancer and the web servers. The load balancer has a single network interface. The default gateway on the web servers must be set to be the load balancer – this ensures that return traffic goes back to the client via the load balancer, which is a requirement of layer 4 NAT mode.

a) Setting up AWS

1. Deploy the load balancer instance as described starting on page 11
2. Deploy your required web server instances into the same VPC & subnet as the load balancer
3. The default route of the Real Servers must be changed to be the load balancer (10.0.0.62). The example command below is for an Ubuntu Linux host:

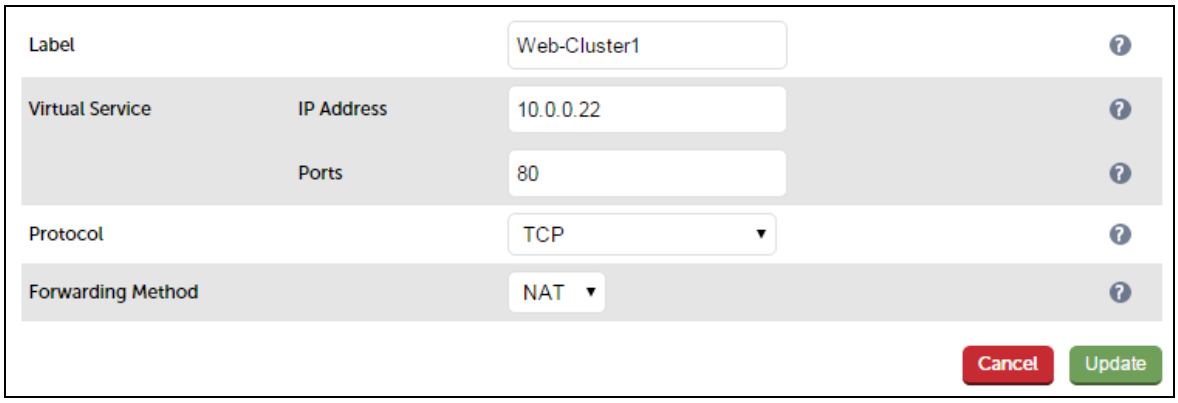
```
$ sudo ip route replace default via 10.0.0.62
```

4. The screen shot below shows that the default route is now set as the load balancer



b) Setting up the Virtual Service

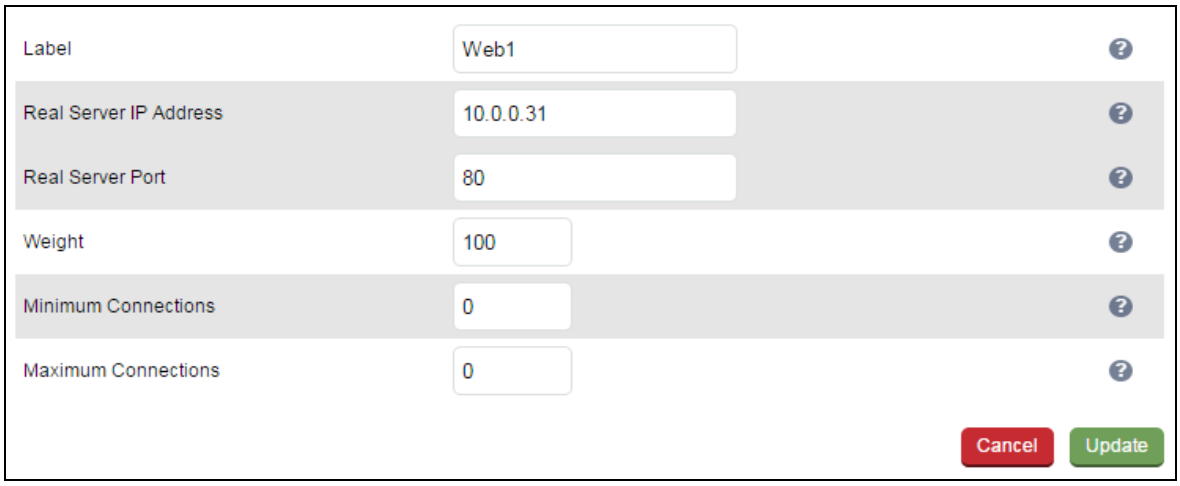
1. Using the WebUI, navigate to: *Cluster Configuration > Layer 4 – Virtual Service* and click **Add a New Virtual Service**
2. Enter the following details:



3. Enter an appropriate label for the VIP, e.g. **Web-Cluster1**
4. Set the *Virtual Service IP address* field to the required IP address, e.g. **10.0.0.22**
5. Set the *Virtual Service Ports* field to the required port, e.g. **80**
6. Leave *Protocol* set to **TCP**
7. Click **Update**

c) Setting up the Real Servers

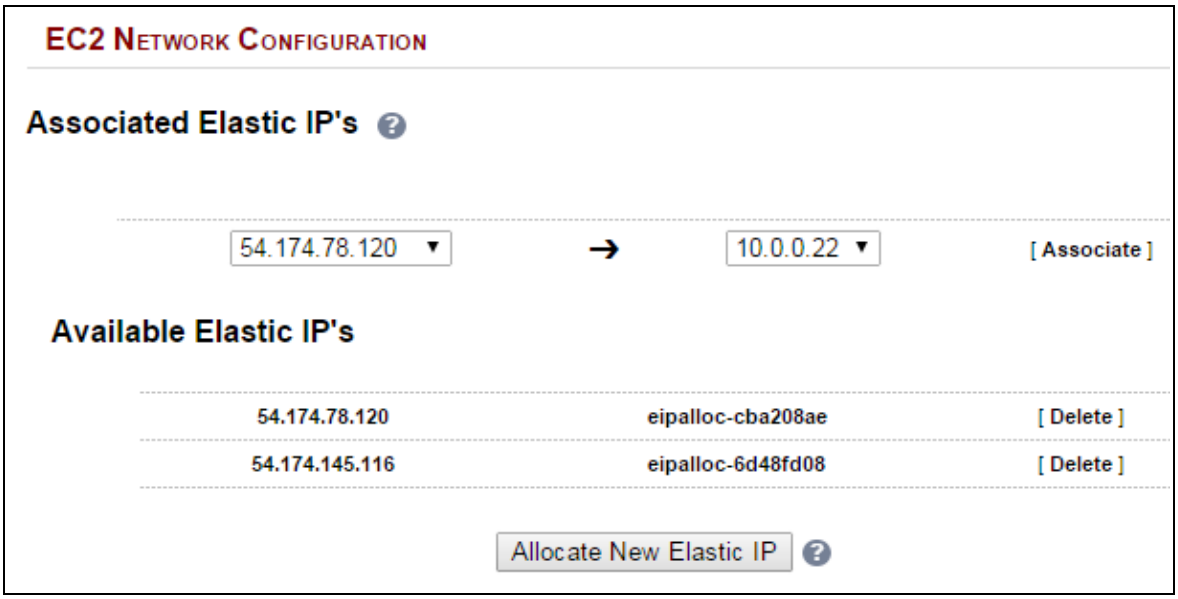
1. Using the WebUI, navigate to: *Cluster Configuration > Layer 4 – Real Servers* and click **Add a new Real Server** next to the newly created VIP
2. Enter the following details:



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. **10.0.0.31**
5. Set *Real Server Port* to **80**
6. Click **Update**
7. Repeat the above steps to add your other web servers(s)

d) Associating the VIP with an Elastic IP Address (If access from the Internet is required)

1. Using the WebUI, navigate to: *EC2 Configuration > EC2 Network Configuration*



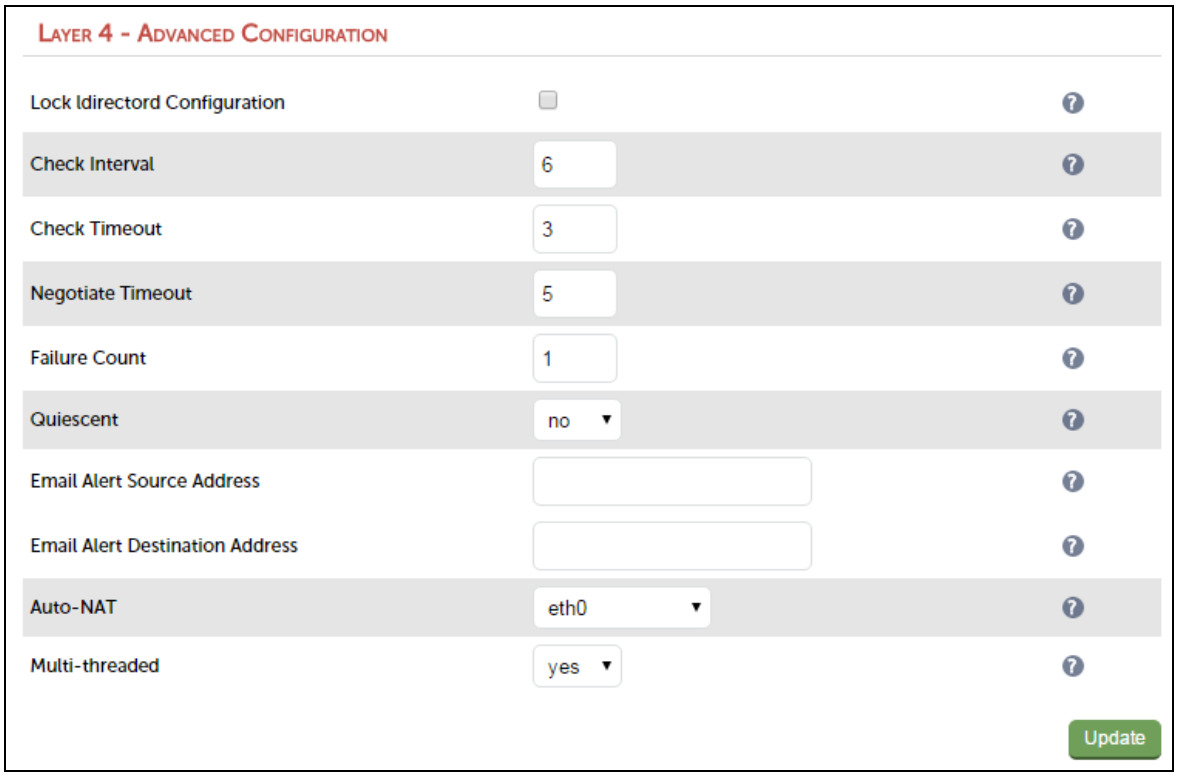
2. Under the Associated Elastic IP's section click **[Associate]** next to the VIPs private IP address (10.0.0.22 in this case), if no Elastic IP's are available, use the **Allocate New Elastic IP** button to add one

> Note:
>
> If EIP(s) are associated with secondary IP's, you **MUST** also associate an EIP with the primary IP on the instance. Please refer to page 10 for more details.

e) Enable Internet Connectivity via the Load Balancer for the Real Servers (if required)

If the Real Servers need to access the Internet, 'Autonat' must be enabled on the load balancer to enable this functionality.

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 4 – Advanced Configuration*



2. Change the *Auto-NAT* setting to **eth0**
3. Click **Update**

3 - WEB SERVERS – 2 SUBNETS, 2 LOAD BALANCER NETWORK INTERFACES, LAYER 7

This example uses 2 subnets – the load balancer is configured with 2 interfaces - 1 interface in subnet 1 and the other in subnet 2. The real severs are connected to subnet 2.

a) Setting up AWS

1. Deploy the load balancer instance as described starting on page 11
2. Add a second subnet to your VPC, skip this step if you already have one
3. Add a second Network Interface, associate it with the second subnet and attach it to the load balancer instance
4. Deploy your required web server instances into the second subnet

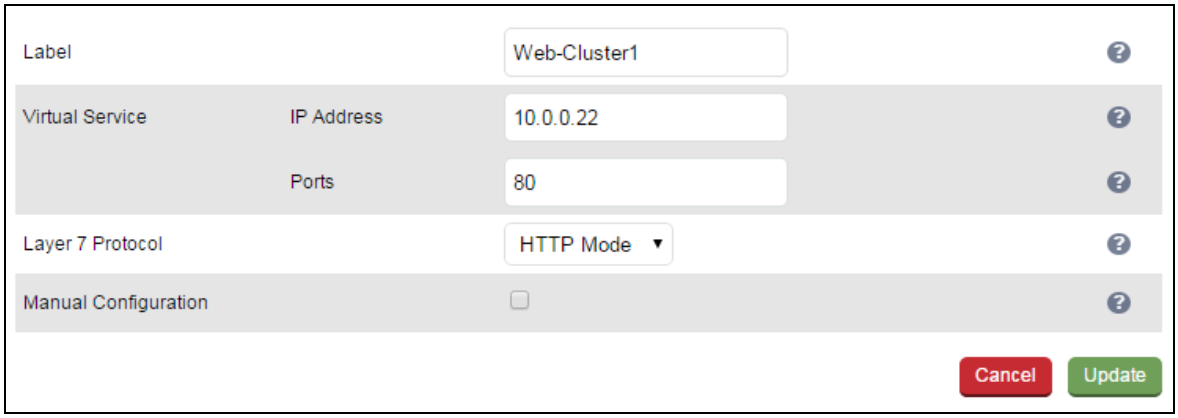
b) Configuring the second Network Interface

1. Using the WebUI, navigate to: *Local Configuration > Network Interface Configuration* , assign an IP address for the second interface (eth1) , e.g. **10.0.2.220/24**

2. Click **Configure Interfaces**

c) Setting up the Virtual Service

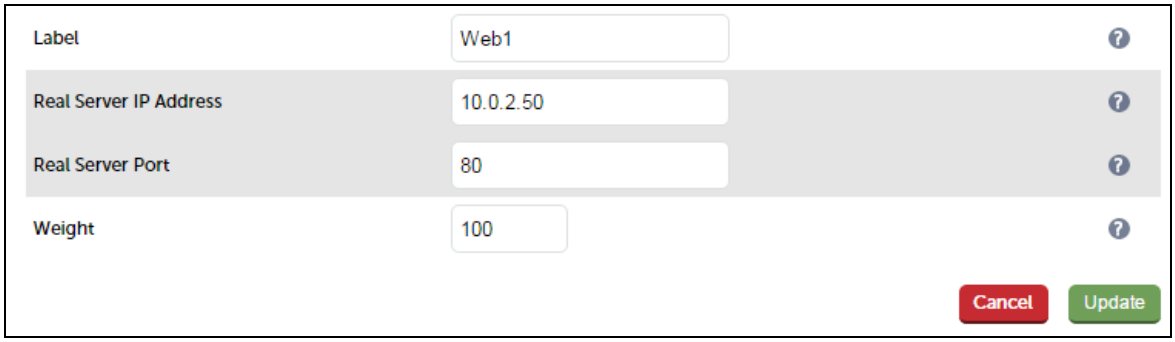
1. Using the WebUI, navigate to: *Cluster Configuration > Layer 7 – Virtual Service* and click **Add a New Virtual Service**
2. Enter the following details:



3. Enter an appropriate label for the VIP, e.g. **Web-Cluster1**
4. Set the *Virtual Service IP address* field to the required IP address, e.g. **10.0.0.22**
5. Set the *Virtual Service Ports* field to the required port, e.g. **80**
6. Leave *Layer 7 Protocol* set to **HTTP Mode**
7. Click **Update**

d) 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:



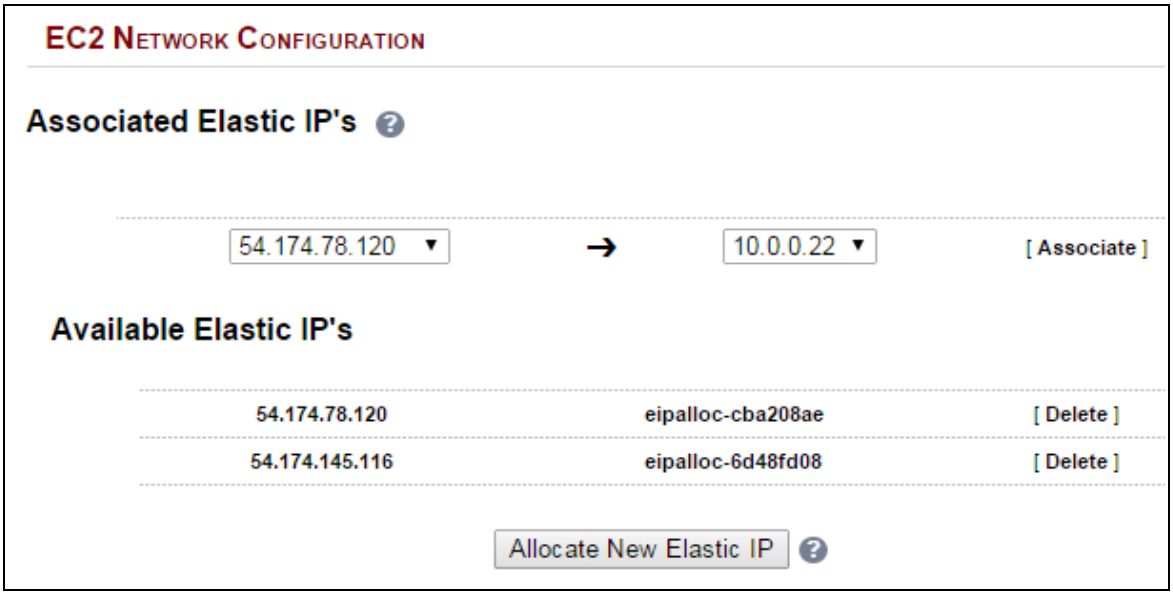
3. Enter an appropriate label for the RIP, e.g. **Web1**
4. Change the *Real Server IP Address* field to the required IP address, e.g. **10.0.2.50**
5. Set the *Real Server Port* field to the required port, e.g. **80**
6. Click **Update**
7. Repeat the above steps to add your other web server(s)

e) Applying the new Layer 7 Settings

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

f) Associating the VIP with an Elastic IP Address (If access from the Internet is required)

1. Using the WebUI, navigate to: *EC2 Configuration > EC2 Network Configuration*



2. Under the Associated Elastic IP's section click **[Associate]** next to the VIPs private IP address (10.0.0.22 in this case), if no Elastic IP's are available, use the **Allocate New Elastic IP** button to add one

> Note:
>
> If EIP(s) are associated with secondary IP's, you **MUST** also associate an EIP with the primary IP on the instance. Please refer to page 10 for more details.

Note:

Dual interface layer 7 SNAT mode with TProxy enabled (for transparency) where each interface of the load balancer is connected to a different subnet and the default gateway of the real servers is configured to be the load balancer is **not** supported. Please refer to example 4 on page 37 instead if you require layer 7 with transparency.

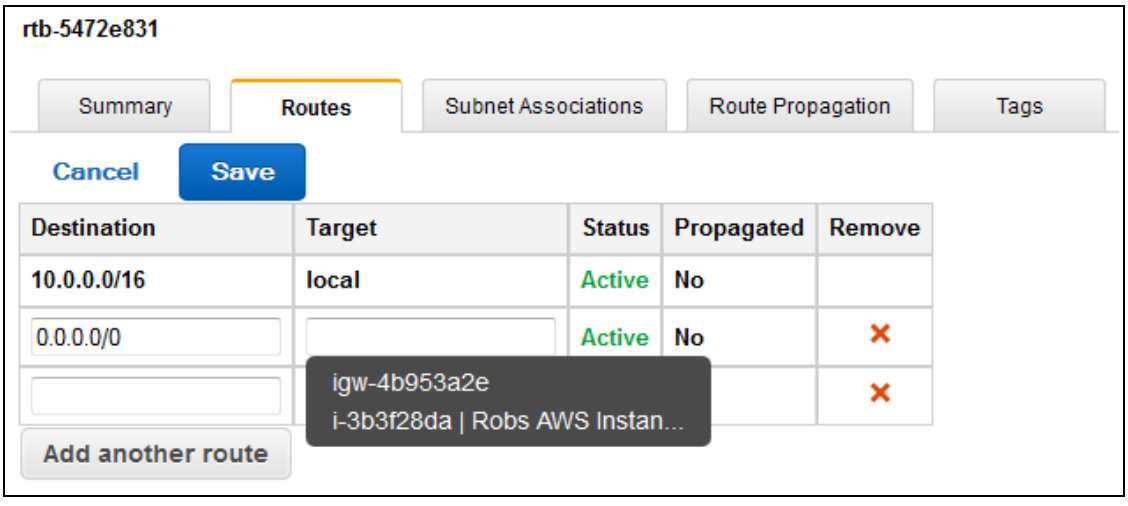
4 - WEB SERVERS - 2 SUBNETS, 1 LOAD BALANCER NETWORK INTERFACE, LAYER 7, TRANSPARENT

This example uses 2 subnets - one subnet for the load balancer and one subnet for the web servers. The load balancer has a single network interface located in the first subnet. Layer 7 transparency is enabled to ensure that the source IP address of packets reaching the web servers is the source IP of the clients and not the IP address of the load balancer. Routing rules for the second subnet must also be changed.

a) Setting up AWS

1. Deploy the load balancer instance as described starting on page 11
2. Add a second subnet to your VPC, skip this step if you already have one
3. Deploy your required web server instances into the second subnet
4. Add a default route to the second subnets routing table (the subnet where the web servers are located), set the target to be the interface on the load balance

   - Under the VPC dashboard, select *Route Tables*
   - Select the route table that relates to the second subnet
   - Select the *Routes* tab, and click **Edit**
   - Click **Add another route**
   - In the blank row at the bottom set the destination to 0.0.0.0/0 and set the target to be the ENI on the load balancer – in this example "i-3b3f28da | Robs AWS Instance" as shown below

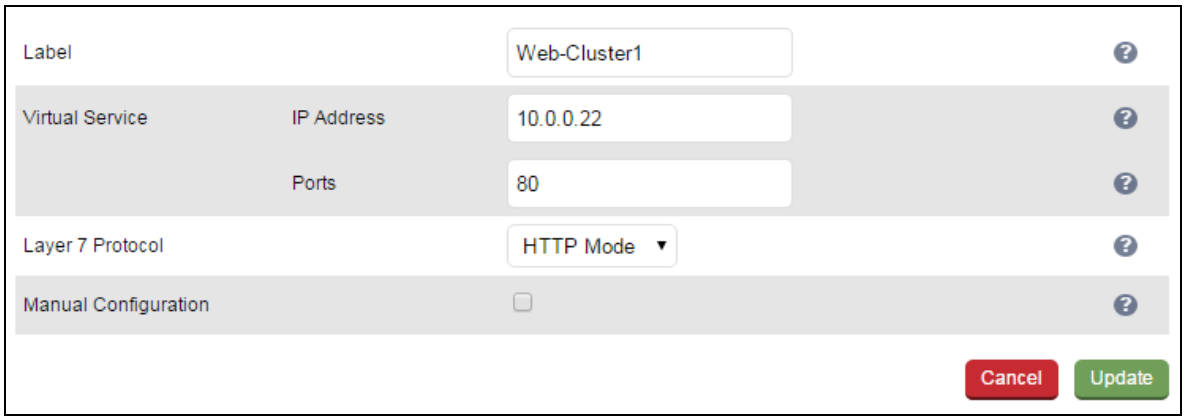


IMPORTANT:

Make sure you have disabled the Source/Destination Check for the Load Balancer instance, otherwise the load balancer will **NOT** be displayed as an option.

b) Setting up the Virtual Service

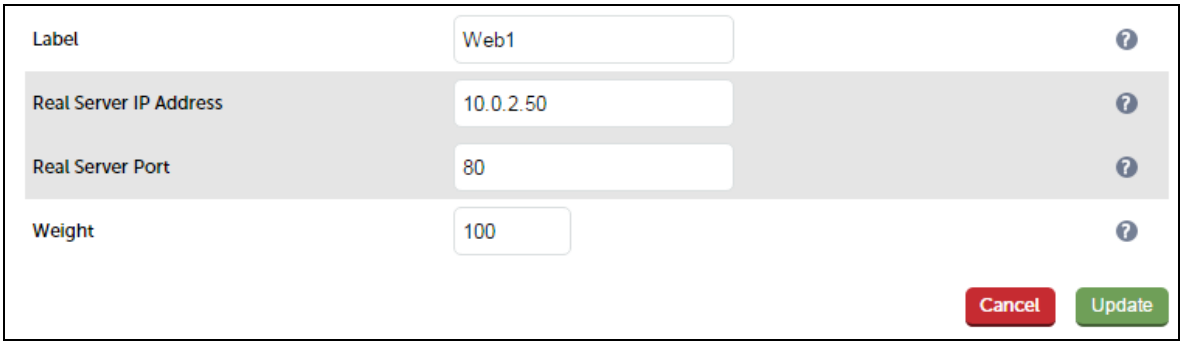
1. Using the WebUI, navigate to: *Cluster Configuration > Layer 7 – Virtual Service* and click **Add a New Virtual Service**
2. Enter the following details:



3. Enter an appropriate label for the VIP, e.g. **Web-Cluster1**
4. Set the *Virtual Service IP address* field to the required IP address, e.g. **10.0.0.22**
5. Set the *Virtual Service Ports* field to the required IP address, e.g. **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. **10.0.2.50**
5. Set the *Real Server Port* field to the required port, e.g. **80**
6. Click **Update**
7. Repeat the above steps to add your other web server(s)

d) Configuring Layer 7 – Advanced Settings

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 7 – Advanced Configuration*

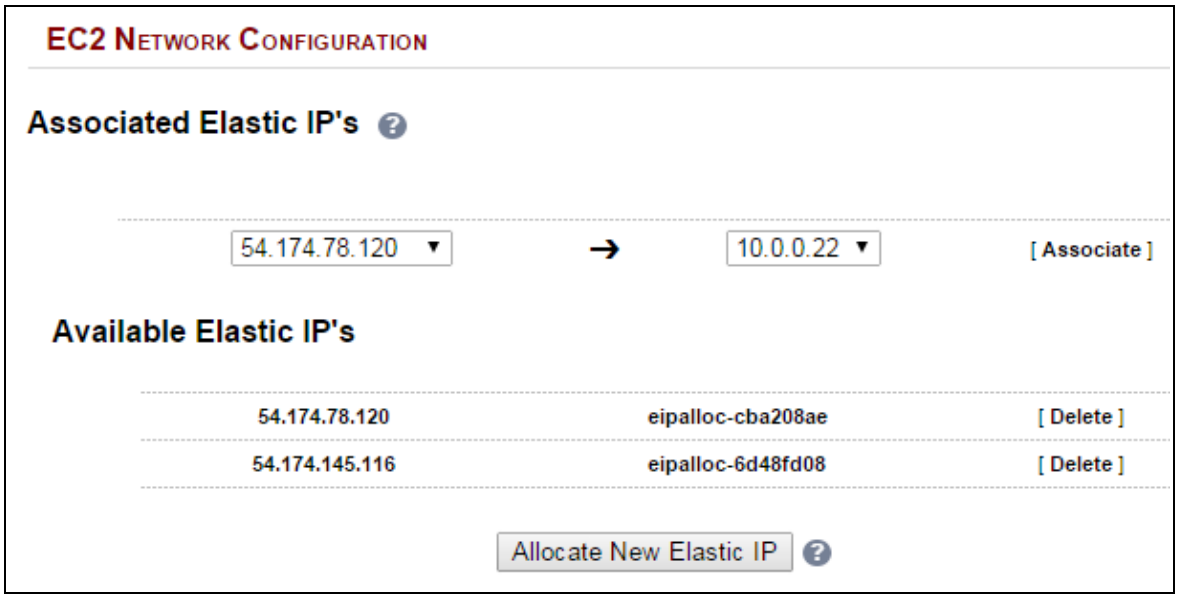
2. Enable (check) *Transparent Proxy*
3. Click **Update**

e) Applying the new Layer 7 Settings

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

f) Associating the VIP with an Elastic IP Address (If access from the Internet is required)

1. Using the WebUI, navigate to: *EC2 Configuration > EC2 Network Configuration*



2. Under the Associated Elastic IP's section click **[Associate]** next to the VIPs private IP address (10.0.0.22 in this case), if no Elastic IP's are available, use the **Allocate New Elastic IP** button to add one

> Note:
>
> If EIP(s) are associated with secondary IP's, you **MUST** also associate an EIP with the primary IP on the instance. Please refer to page 10 for more details.

5 - WEB SERVERS - 1 SUBNET, 1 LOAD BALANCER NETWORK INTERFACE, LAYER 7, SSL TERMINATION

This is the same as example 1 with the addition of SSL termination on the load balancer.

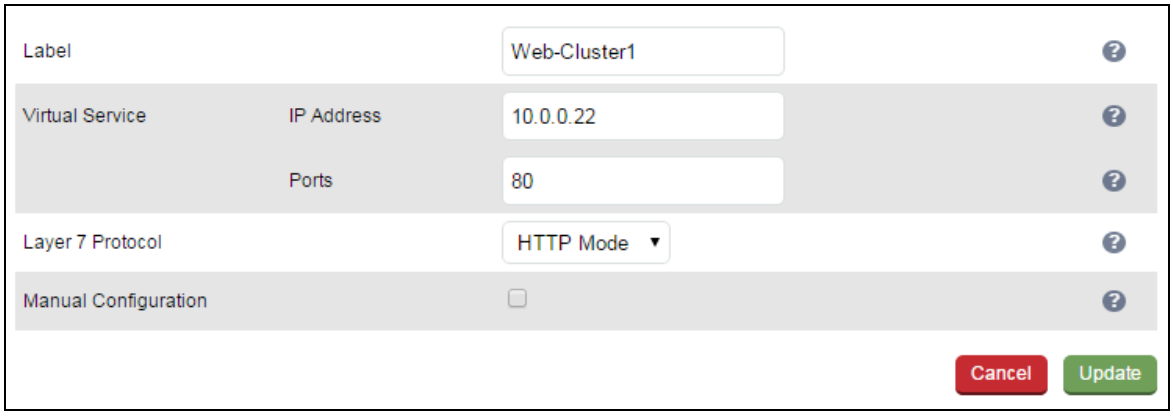
> Note:
>
> We generally recommend that SSL should be terminated on the backend servers rather than the load balancer for scalability reasons.

a) Setting up AWS

1. Deploy the load balancer instance as described starting on page 11
2. Deploy your required web server instances into the same VPC & subnet as the load balancer

b) Setting up the Virtual Service

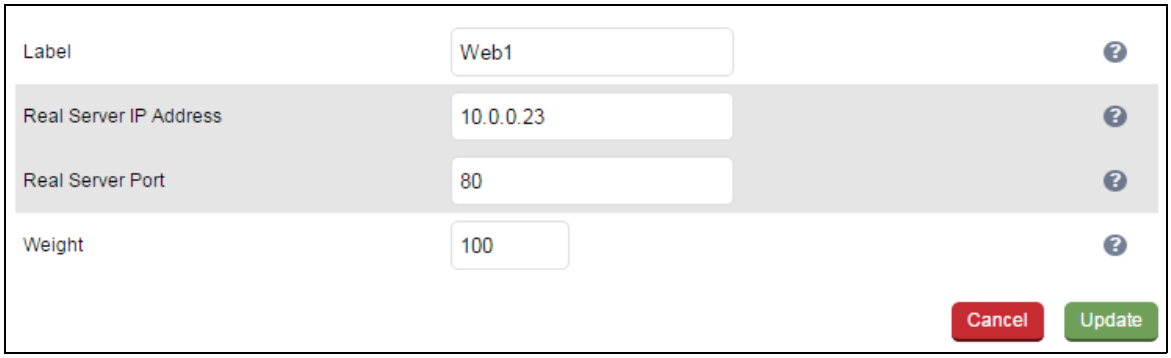
1. Using the WebUI, navigate to: *Cluster Configuration > Layer 7 – Virtual Service* and click **Add a New Virtual Service**
2. Enter the following details:



3. Enter an appropriate label for the VIP, e.g. **Web-Cluster1**
4. Set the *Virtual Service IP address* field to the required IP address, e.g. **10.0.0.22**
5. Set the *Virtual Service Ports* field to the required port, e.g. **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:

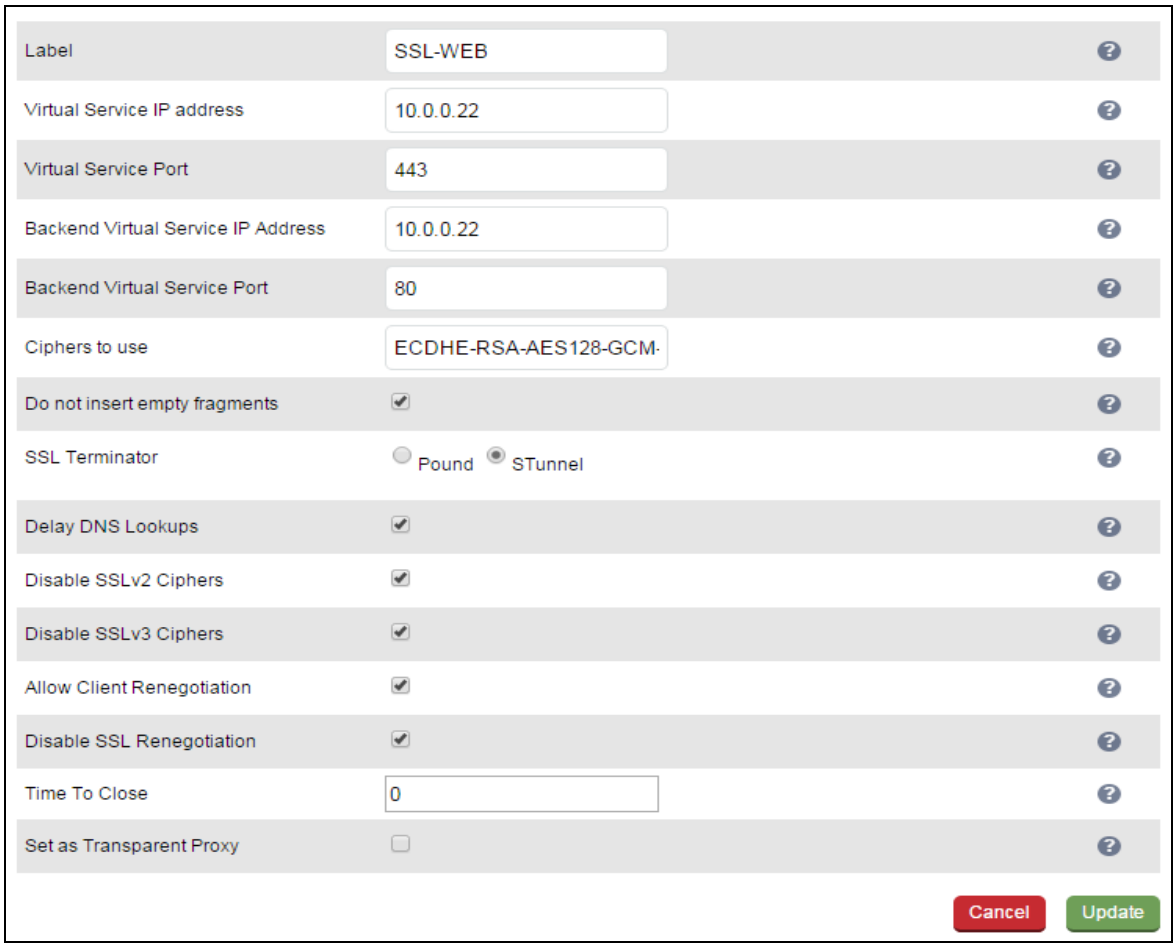


3. Enter an appropriate label for the RIP, e.g. **Web1**
4. Change the *Real Server IP Address* field to the required IP address, e.g. **10.0.0.23**
5. Set the *Real Server Port* field to the required port, e.g. **80**
6. Click **Update**

7. 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. **10.0.0.22**
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. **10.0.0.22**
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:

1. A default self-signed certificate is used when SSL virtual services are first defined
2. To change this, Using the WebUI, navigate to: *Cluster Configuration > SSL Termination*
3. Click **[Certificate]** next to the Virtual Service
4. If you already have a certificate, use the **Upload prepared PEM/PFX file** option at the bottom of

the screen to upload it

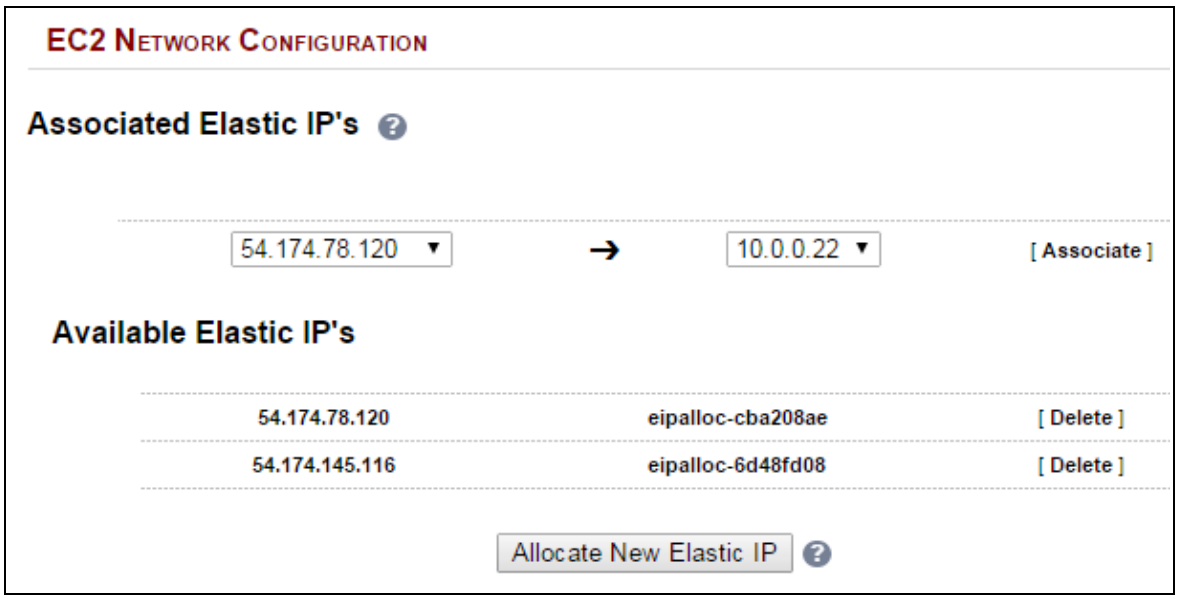
5. If you don't have a certificate, you can create a CSR using the **Generate SSL Certificate Request** section. This will create the CSR in the upper pane of the **Upload Signed Certificate** section based on the settings you enter. This should be copied and sent to your CA
6. Once the signed certificate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

e) Applying the new Settings

1. Once the configuration is complete, use the **Reload HAProxy** button at the top of the screen to apply the changes
2. Once the configuration is complete, use the **Restart Stunnel** button at the top of the screen to apply the changes

f) Associating the VIP with an Elastic IP Address (If access from the Internet is required)

1. Using the WebUI, navigate to: *EC2 Configuration > EC2 Network Configuration*



2. Under the Associated Elastic IP's section click **[Associate]** next to the VIPs private IP address (10.0.0.22 in this case), if no Elastic IP's are available, use the **Allocate New Elastic IP** button to add one

> Note:
>
> If EIP(s) are associated with secondary IP's, you **MUST** also associate an EIP with the primary IP on the instance. Please refer to page 10 for more details.

6 - RD SESSION HOSTS - 2 SUBNETS, 1 LOAD BALANCER NETWORK INTERFACE, LAYER 7

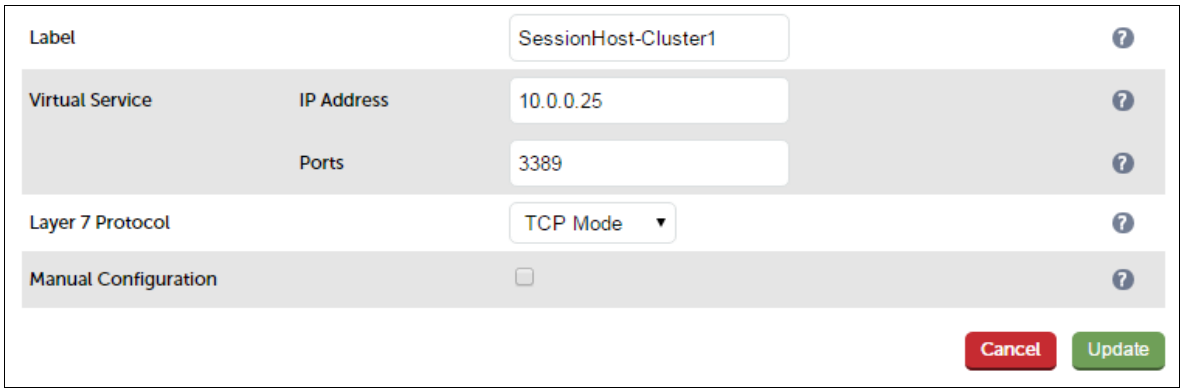
This example uses 2 subnets - one subnet for the load balancer and one subnet for the session hosts. The load balancer has a single network interface located in the first subnet.

a) Setting up AWS

1. Deploy the load balancer instance as described starting on page 11
2. Add a second subnet to your VPC, skip this step if you already have one
3. Deploy your required session host server instances into the second subnet

b) Setting up the Virtual Service

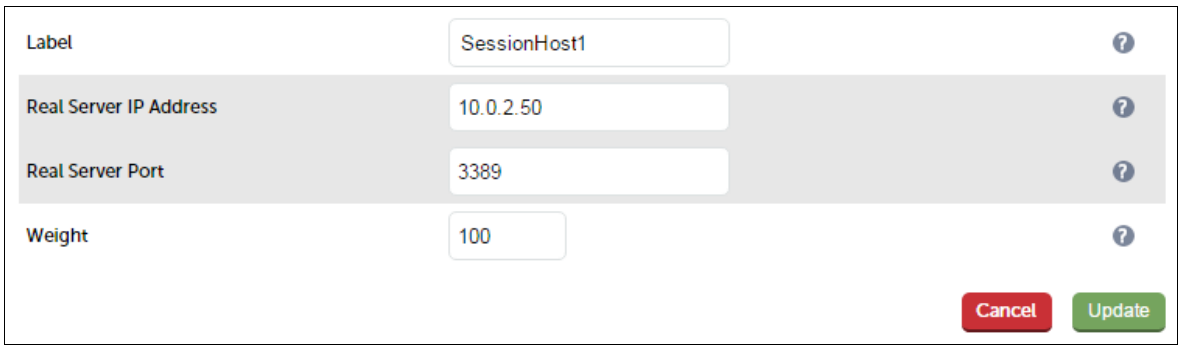
1. Using the WebUI, navigate to: *Cluster Configuration > Layer 7 – Virtual Service* and click **Add a New Virtual Service**
2. Enter the following details:



3. Enter an appropriate label for the VIP, e.g. **SessionHost-Cluster1**
4. Set the *Virtual Service IP address* field to the required IP address, e.g. **10.0.0.25**
5. Set the *Virtual Service Ports* field to the required IP address, e.g **3389**
6. Leave *Layer 7 Protocol* set to **TCP Mode**
7. Click **Update**
8. Now click **Modify** next to the newly created Virtual Service
9. Set *Persistence Mode* to **Source IP**
10. 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. **SessionHost1**

4. Change the *Real Server IP Address* field to the required IP address, e.g. **10.0.2.50**
5. Set the *Real Server Port* field to the required port, e.g. **3389**
6. Click **Update**
7. Repeat the above steps to add your other session host 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 apply the changes

e) Associating the VIP with an Elastic IP Address (If access from the Internet is required)

1. Using the WebUI, navigate to: *EC2 Configuration > EC2 Network Configuration*

2. Under the Associated Elastic IP's section click **[Associate]** next to the VIPs private IP address (10.0.0.25 in this case), if no Elastic IP's are available, use the **Allocate New Elastic IP** button to add one

> Note:
>
> If EIP(s) are associated with secondary IP's, you **MUST** also associate an EIP with the primary IP on the instance. Please refer to page 10 for more details.

7 - WEB SERVERS - 2 SUBNETS, 1 LOAD BALANCER NETWORK INTERFACE, LAYER-4

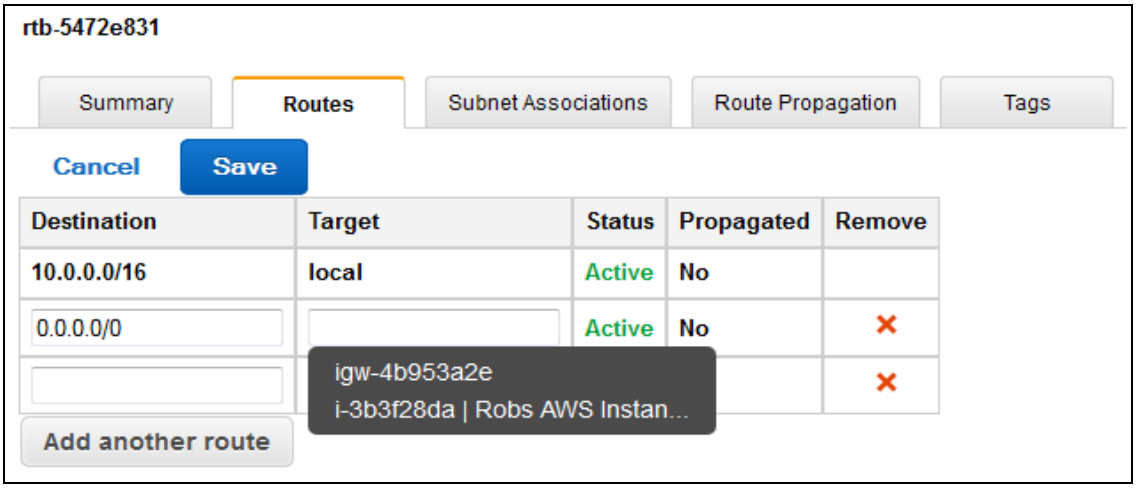
This example uses 2 subnets - one subnet for the load balancer and one subnet for the web servers. The load balancer has a single network interface located in the first subnet. Routing rules for the second subnet must be changed so that return traffic passes back via the load balancer.

a) Setting up AWS

1. Deploy the load balancer instance as described starting on page 11

2. Add a second subnet to your VPC, skip this step if you already have one
3. Deploy your required web server instances into the second subnet
4. Add a default route to the second subnets routing table (the subnet where the web servers are located), set the target to be the interface on the load balancer

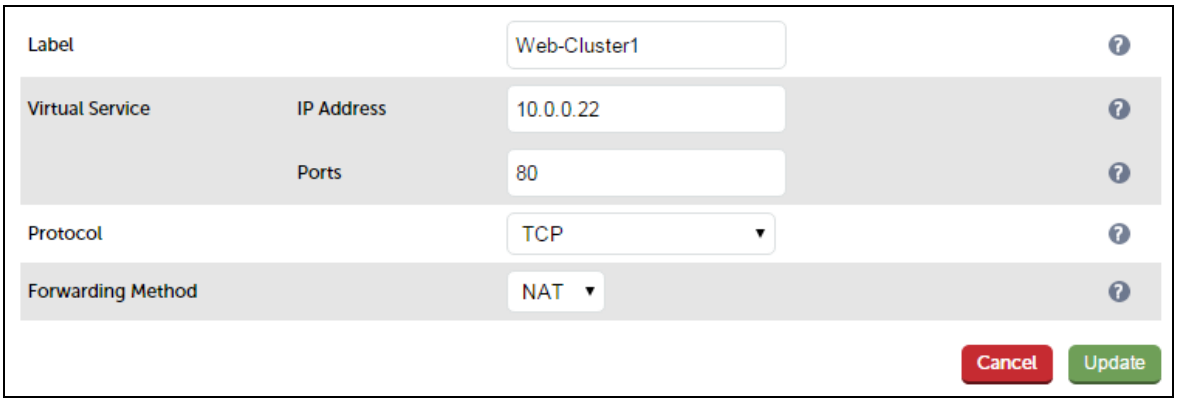
   - Under the VPC dashboard, select *Route Tables*
   - Select the route table that relates to the second subnet
   - Select the *Routes* tab, and click **Edit**
   - In the blank row at the bottom set the destination to 0.0.0.0/0 and set the target to be the ENI on the load balancer – in this example "**i-3b3f28da | Robs AWS Instance**" as shown below



> IMPORTANT:
>
> Make sure you have disabled the Source/Destination Check for the Load Balancer instance, otherwise the load balancer will NOT be displayed as an option.

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:

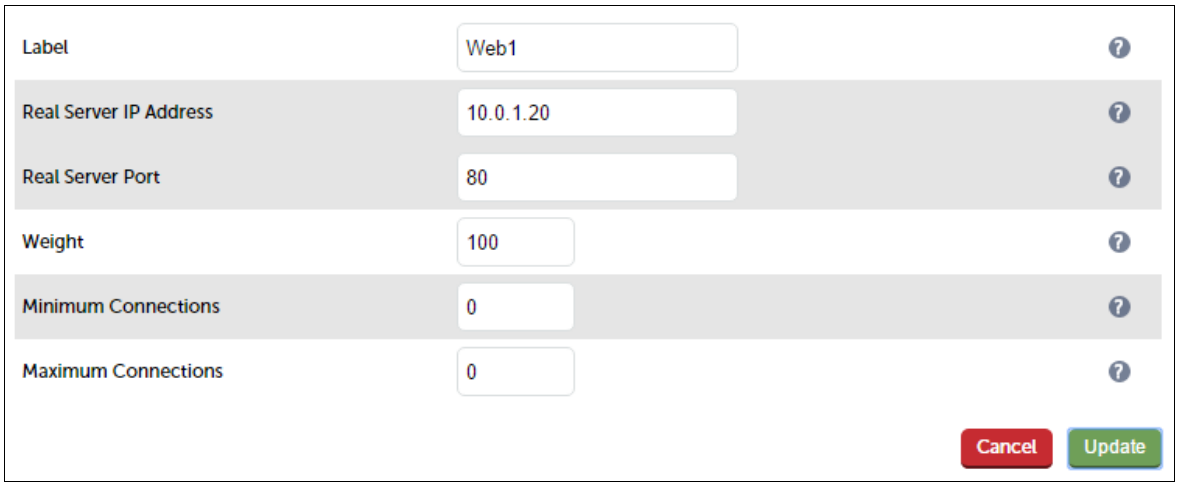


3. Enter an appropriate label for the VIP, e.g. **Web-Cluster1**

4. Set the *Virtual Service IP address* field to the required IP address, e.g. **10.0.0.22**
5. Set the *Virtual Service Ports* field to the required port, e.g. **80**
6. Leave *Protocol* set to **TCP**
7. Click **Update**

c) Setting up the Real Servers

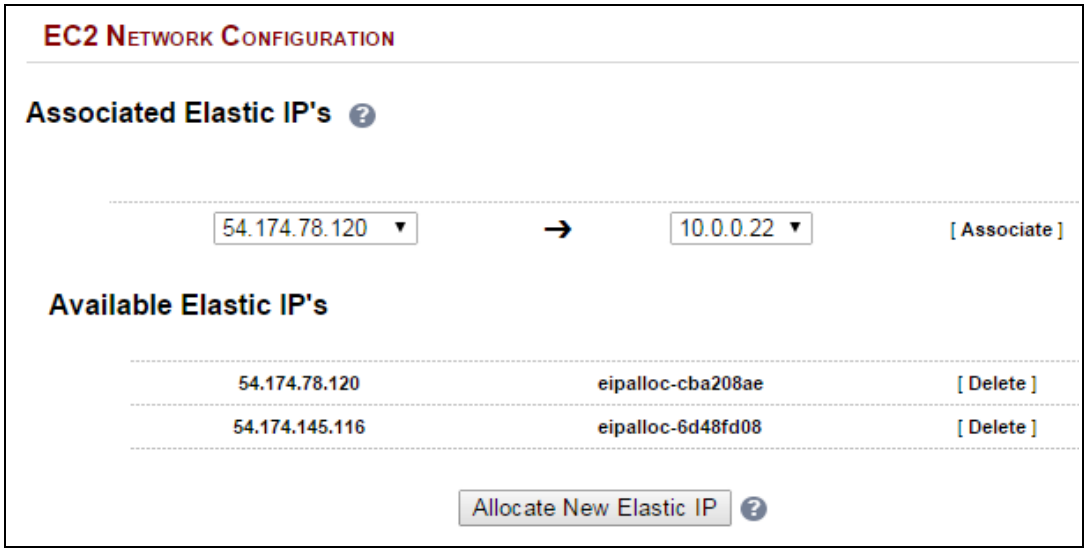
1. Using the WebUI, navigate to: *Cluster Configuration > Layer 4 – Real Servers* and click **Add a new Real Server** next to the newly created VIP
2. Enter the following details:



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. **10.0.1.20**
5. Set *Real Server Port* to **80**
6. Click **Update**
7. Repeat the above steps to add your other web servers(s)

d) Associating the VIP with an Elastic IP Address (If access from the Internet is required)

1. Using the WebUI, navigate to: *EC2 Configuration > EC2 Network Configuration*



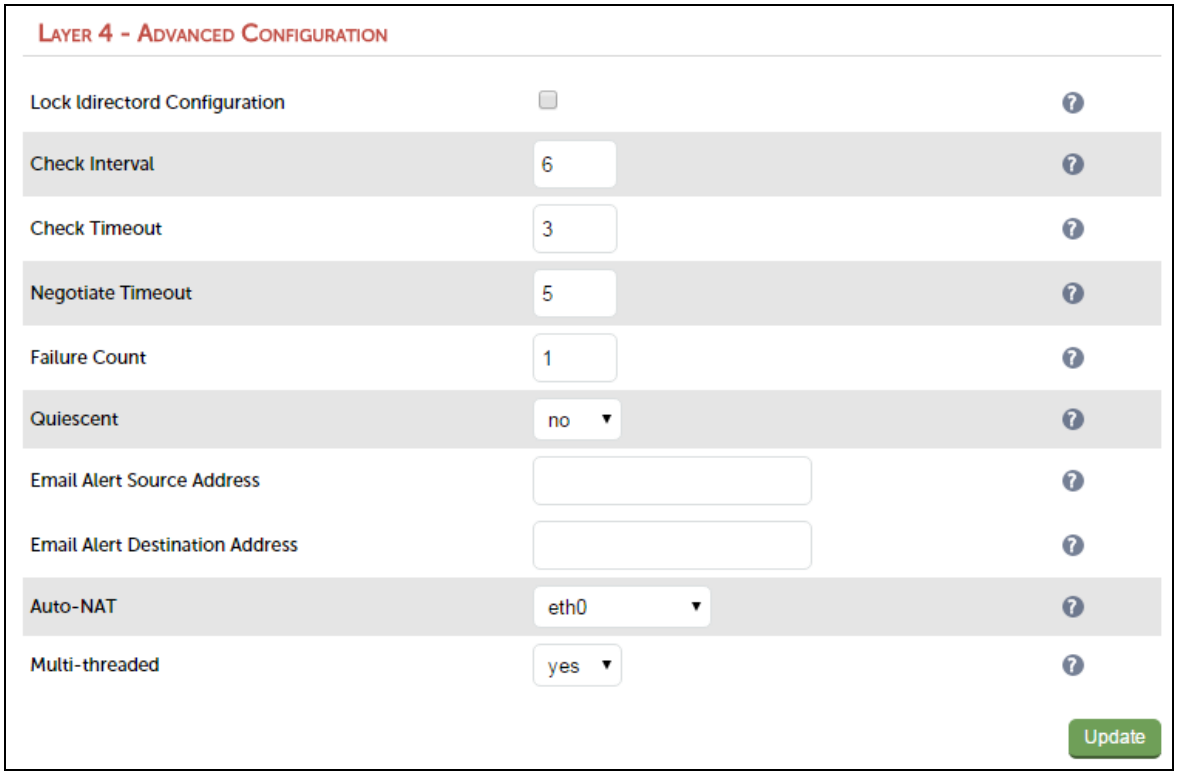
2. Under the Associated Elastic IP's section click **[Associate]** next to the VIPs private IP address (10.0.0.22 in this case), if no Elastic IP's are available, use the **Allocate New Elastic IP** button to add one

> Note:
>
> If EIP(s) are associated with secondary IP's, you **MUST** also associate an EIP with the primary IP on the instance. Please refer to page 10 for more details.

e) Enable Internet Connectivity via the Load Balancer for the Real Servers (If Required)

If the Real Servers need to access the Internet, 'Autonat' must be enabled on the load balancer to enable this functionality.

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 4 – Advanced Configuration*



2. Change the *Auto-NAT* setting to **eth0**
3. Click **Update**

9. Configuring High Availability using two Instances (Master & Slave)

Enterprise AWS supports HA mode using two instances 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). If the active device fails for any reason, the passive device will take over.

> Note:
>
> This procedure assumes the first appliance is already up and running, and that this appliance will be the master unit of the clustered pair.

> Note:
>
> For an internet facing HA pair , you'll need 3 EIPS's - 1 for the primary interface on each instance and 1 for the VIP. See page 10 for more details.

Step 1 – Deploy a second Instance & Configure the Source/Dest. Check

1. Please refer to the steps starting on page 11
2. Right-click the instance and select: *Networking > Change Source/Dest. Check* and ensure this is disabled

Step 2 – Prepare both instances for pairing

1. Using the WebUI, navigate to: *Local configuration > Execute Shell Command* run the following command on both appliances:

   lb_enable_root enable

Step 3 – Update Security Group Settings

1. Ensure that the security group used by both instances has the following additional rules defined. These are required to ensure that heartbeat (used for HA communication) can communicate between the two instances.

   Rule 1:
   Type: Custom UDP rule
   Protocol: UDP
   Port Range: 6694
   Source: Anywhere (or lockdown further if preferred)

   Rule 2:
   Type: Custom ICMP rule
   Protocol: Echo Request
   Port Range: N/A
   Source: Anywhere (or lockdown further if preferred)

> Note:
>
> Make sure you select ICMP Echo Request rather than ICMP Echo Reply.

Step 4 – Configure Heartbeat Failover Script (Applies to Layer 4 NAT mode and Layer 7 with Tproxy)

For Layer 4 NAT mode, or Layer 7 mode with TProxy enabled, AWS routing rules must be configured so that the load balancer is the default gateway. To enable successful failover to the slave device, these routing rules must then be changed to route via the slave instance. To set this up:

1. On the master instance select the menu option: *Cluster Configuration > Heartbeat Advanced* and add the following line:

   aws ec2 replace-route --route-table-id rtb-15127270 --destination-cidr-block 0.0.0.0/0 --instance-id i-f40efc59 --region eu-west-1

   *(change **rtb-15127270** to the Route Table ID of the table associated with your real servers subnet)*
   *(change **i-f40efc59** to the Instance-Id of your master instance)*
   *(change **eu-west-1** to your region)*

   this sets the default route for the routing table associated with the subnet where your real servers are located to be the master instance. It's run automatically each time the master becomes active

2. On the slave instance select the menu option: *Cluster Configuration > Heartbeat Advanced* and add the following line:

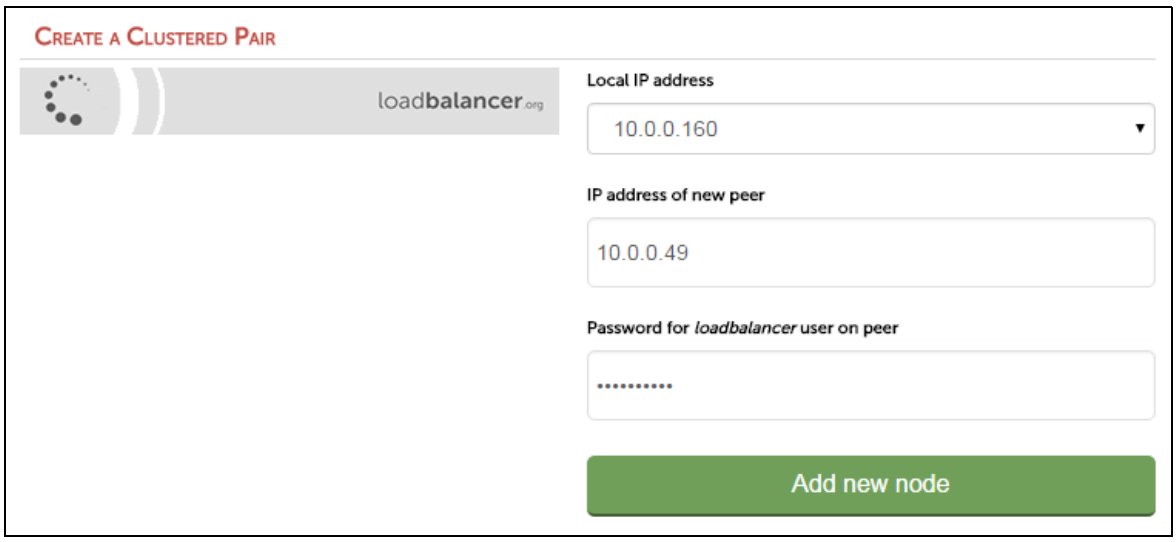
   aws ec2 replace-route --route-table-id rtb-15127270 --destination-cidr-block 0.0.0.0/0 --instance-id i-f45ejc53 --region eu-west-1

   *(change **rtb-15127270** to the Route Table ID of the table associated with your real servers subnet)*
   *(change **i-f45ejc53** to the Instance-Id of your slave instance)*
   *(change **eu-west-1** to your region)*

   this sets the default route for the routing table associated with the subnet where your real servers are located to be the slave instance. It's run automatically each time the slave becomes active

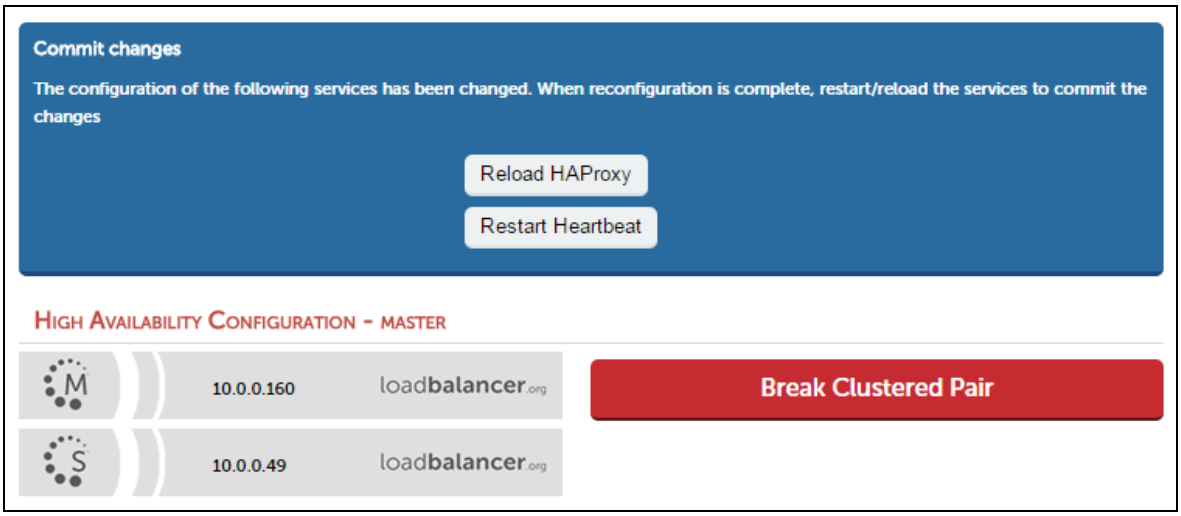
Step 5 – Configure High-Availability

1. Open the WebUI on the master unit
2. Select the menu option: *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 *Instance-ID* of the slave appliance
5. Click **Add new node**
6. Once the pairing configuration has finished, any required 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

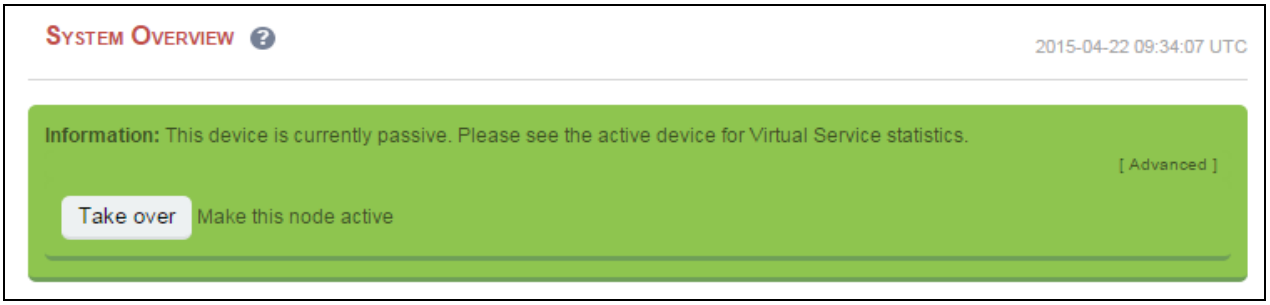
1. Once all services have restarted, the synchronization process will be complete
2. Verify that the status on the master & slave is as follows:

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

10. Configuring High Availability using two Instances across Availability Zones

From v8.2.2 Enterprise AWS also supports HA mode using two instances deployed in different AZ's. In this mode, VIPs are configured on both instances and are always locally active, but only one is made available via the associated EIP. For more information on how this mode works, please refer to page 7.

> Note:
>
> For this configuration, you'll need 3 EIPS's - 1 for the primary interface on each instance and 1 for the VIP. Please refer to page 10 for more details on IP address requirements.

Step 1 - Configure a VPC with 2 Public Subnets, each in a different AZ

1. Create a VPC – the simplest way is to use the VPC wizard and using the option ***VPC with a Single Public Subnet***
2. Add a second subnet and specify a different Availability Zone
3. Now make this second subnet a public subnet, by adding a default route with the Target set as an Internet Gateway, e.g.:

Subnet	AZ	CIDR	Destination	Internet GW
1	AZ-1	10.0.0.0/24	0.0.0.0/0	igw-a72528c2
2	AZ-2	10.0.1.0/24	0.0.0.0/0	igw-a72528c2

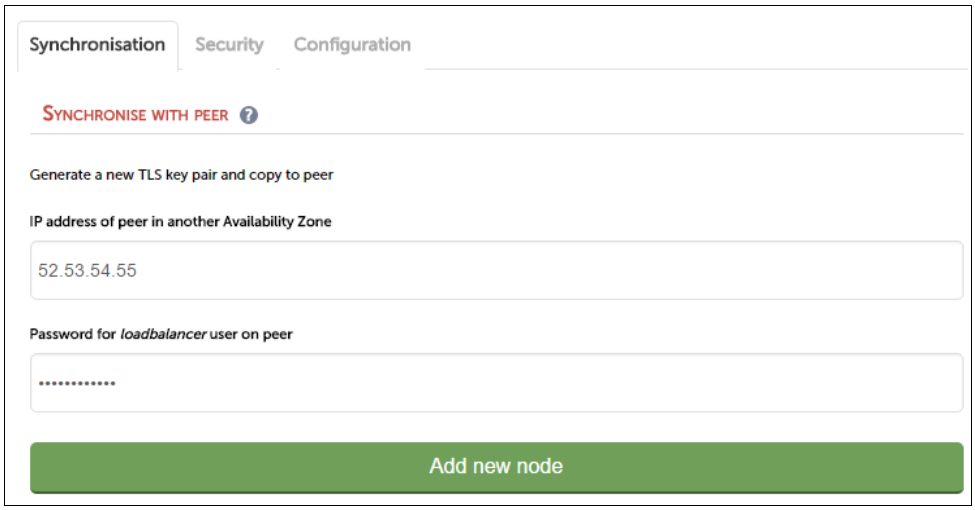
Step 2 – Deploy 2 Instances & Configure the Source/Dest. Check

1. Now deploy 2 instances - one in subnet 1 , the other in subnet 2 , and associate an EIP with each instance. For more information on deploying instances, refer to the steps starting on page 11
2. Right-click each instance and select: *Networking > Change Source/Dest. Check* and ensure this is disabled

Step 3 – Configure Zone HA settings to enable the 2 instances to Communicate

1. On the instance in subnet 1, using the WebUI option: *EC2 Configuration > EC2 Zone HA*

Configuration, select the *Synchronization* Tab

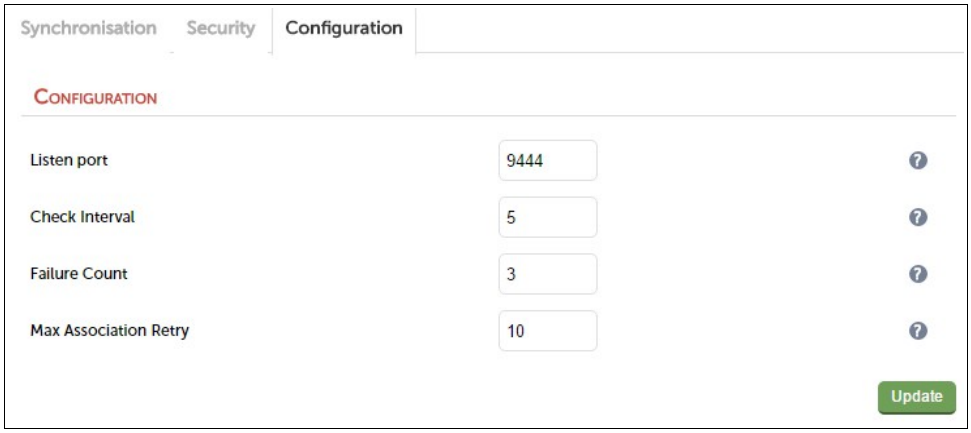


2. Enter the IP address (EIP) and loadbalancer user password for the second instance in subnet 2 (by default this is the *instance ID*)
3. Click **Add new node**
4. A new Keypair & associated certificates will be generated and copied to the second instance. These can be viewed and also deleted if required using the *Security* tab on each appliance

Step 4 - Configure Zone HA Settings

The status of the EIP is constantly checked, and if the EIP is down for longer than the time defined by the check parameters - by default this is 15s (3 x 5) , an EIP association request is generated by the second instance . To view/configure the check parameters:

1. Using the WebUI, navigate to: *EC2 Configuration > EC2 Zone HA Configuration*
2. Select the *Configuration* tab



3. The default values work well in most situation. If these do need to be changed, make the changes on both instances

Step 5 – Update Security Group Settings

1. Ensure that the security group used by each instance has the following additional rule defined, this is required to ensure that the Zone HA check service can contact the peer node

Type: Custom TCP rule
Protocol: TCP
Port Range: 9444
Source: Anywhere (or lockdown further if preferred)

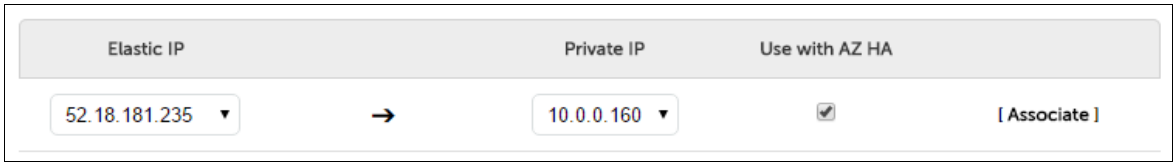
Step 6 – Configure VIPs on both Instances (local private IP addresses)

1. Define VIP1 (e.g **10.0.0.160/24**) with associated RIPs on LB1 in subnet 1/AZ-1
2. Define VIP2 (e.g. **10.0.1.160/24**) with associated RIPs on LB2 in subnet 2/AZ-2

Step 7 – Associate EIPs to Private IPs on the FIRST Instance

1. On the first instance, using the WebUI option: *EC2 Configuration > EC2 Network Configuration*, select the required EIP in the first dropdown and the VIP 1 address in the second dropdown

> Note:
>
> The EIP selected here will be the IP address used by clients to connect to the load balanced services.

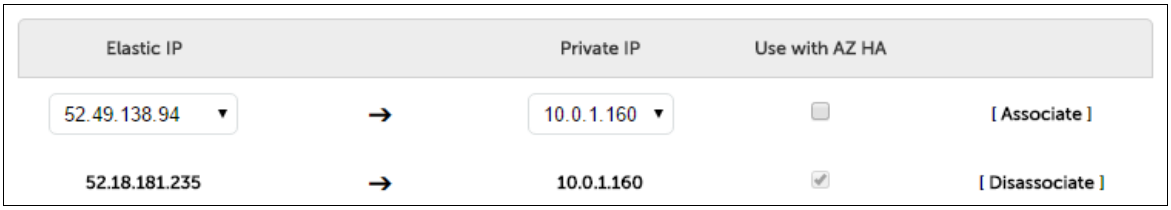


2. Check (tick) the **Use with AZ HA** checkbox
3. Now click the **[Associate]** link to the right of the checkbox, at this point the screen will appear similar to the following:

4. After around 30 seconds, the final status on the first instance configured will be as follows:

Step 8 – Associate EIPs to Private IPs on the SECOND Instance

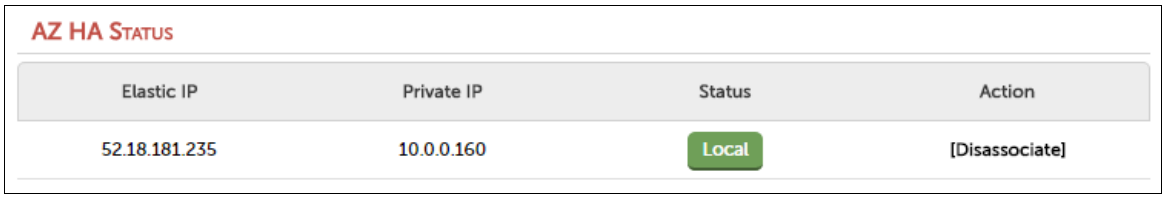
1. Now repeat the procedure listed in step 7 on the second instance, making sure you select the *same EIP address*
2. The final status on the second appliance will be as follows:



> Note:
>
> The Network Configuration screen of the 2 instances will look slightly different as shown in the last 2 screen shots. The instance that is currently associated with the EIP will appear as shown in the first of these 2 screen shots.

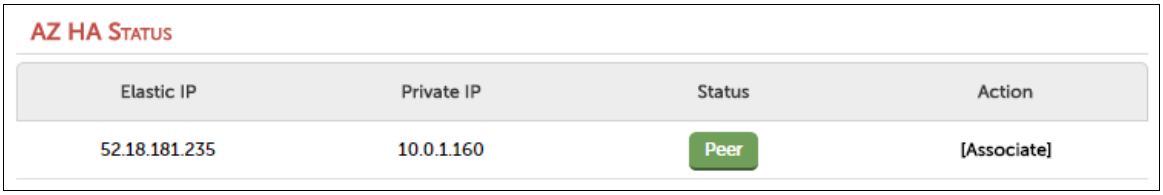
Checking EIP Status

1. On the first instance, using the WebUI, navigate to: *EC2 Configuration > EC2 Zone HA Status* will show the following status:



*- The EIP status is **Local** , i.e. it's active on this instance*

2. On the second instance, Using the WebUI, navigate to: *EC2 Configuration > EC2 Zone HA Status* will show the following status:



*- The EIP status is **Peer** , i.e. it's active on the other instance*

Testing EIP failover

1. Stop the instance where the EIP is currently associated, i.e. where the status is **Local**
2. Verify that the EIP is now associated with the other instance

> Note:
>
> This can take up to 30 seconds to complete.

Manually moving the EIP to the other Instance

To force the EIP to be associated with the other instance:

- Click the **[Associate]** link on the instance where the EIP is not currently active

  Or

- Click the **[Dissociate]** link on the instance where the EIP is currently active

> Note:
>
> This will be the slower of the 2 options because the other device has to first detect that the EIP is down which will cause some initial delay, whereas the first option forces an immediate EIP re-association.

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://192.168.110.10**

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

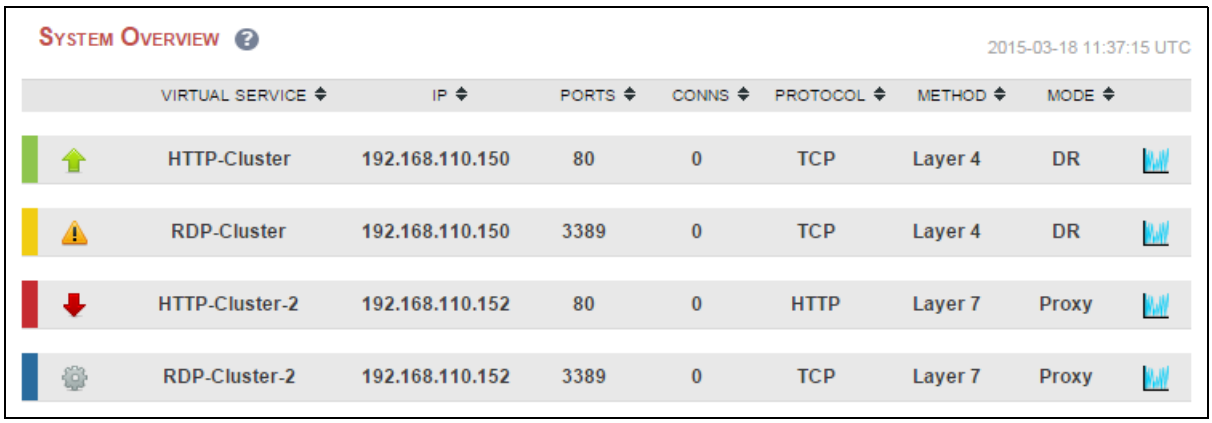
1. ***Make sure that the device is active*** - this can be checked in the WebUI. For a single appliance, the status bar should report **Master** & **Active** as shown below:

2. ***Check that the VIP/floating IP is up*** - Using *View Configuration > Network Configuration* verify that the VIP is active on the load balancer, if not check *Logs > Heartbeat* for errors.

```
eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP qlen 1000
    link/ether 02:bd:88:12:2f:5b brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.220/24 brd 10.0.0.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet 10.0.0.22/24 brd 10.0.0.255 scope global secondary eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::bd:88ff:fe12:2f5b/64 scope link
       valid_lft forever preferred_lft forever
```

The above example shows that the interface (10.0.0.220) and VIP address (10.0.0.22) are both up.

3. ***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).



4. ***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:

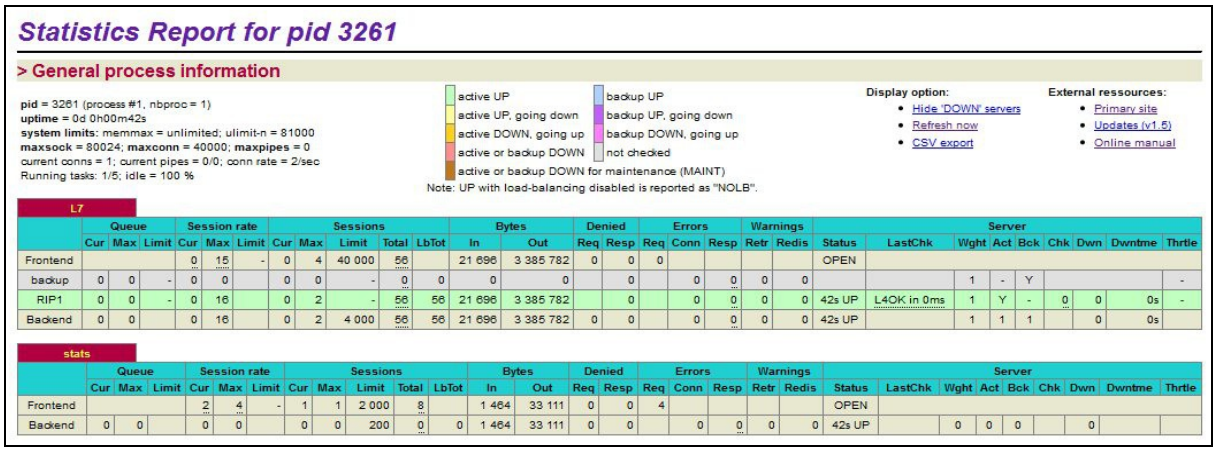
→ for single subnet Layer 4 mode make sure that the default gateway on all real servers is set to be the load balancer

→ for dual subnet Layer 4 mode make sure that routing on the second subnet has 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) :



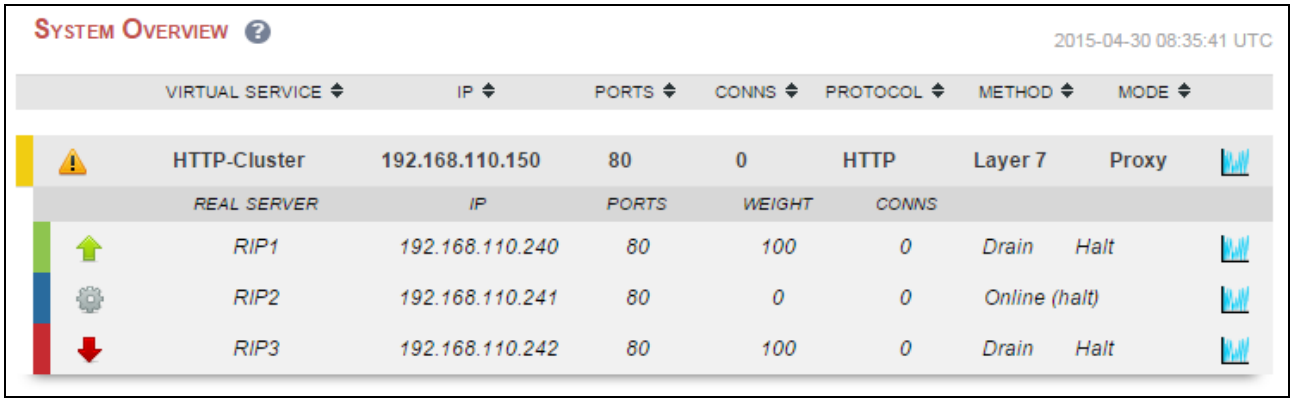
TAKING REAL SERVERS OFFLINE

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. Appendix

1 - IAM ROLE CONFIGURATION

Once configured and associated with the load balancer instance, the IAM role enables the load balancer to securely make EC2 API requests. These requests enable EC2 console functions to be called automatically and minimize the need to configure both the load balancer and EC2. e.g. When EIP's are configured via the load balancer's WebUI, they are also auto-configured in EC2. To configure the required IAM role:

1. In the AWS Console, under the **Security & Identity** section select the **Identity & Access Management** Option
2. Select **Policies** in the Dashboard
3. Click **Create Policy**
4. Click **Select** next to *Create Your Own Policy*
5. Enter a suitable *Policy Name*
6. Copy and paste the policy definition shown on the following page
7. Click **Create Policy**
8. Now select **Roles** in the Dashboard
9. Click **Create New Role**
10. Specify a suitable name and click **Next Step**
11. Click **Select** next to *Amazon EC2*
12. Select the Policy just created
13. Click **Next Step** and then click **Create Role**

IAM Policy Definition – copy & paste this into the new Policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AllocateAddress",
                "ec2:AssignPrivateIpAddresses",
                "ec2:AssociateAddress",
                "ec2:AttachNetworkInterface",
                "ec2:CreateNetworkInterface",
                "ec2:DeleteNetworkInterface",
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAddresses",
                "ec2:DescribeInstanceAttribute",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstances",
                "ec2:DescribeNetworkInterfaceAttribute",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DisassociateAddress",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:ReleaseAddress",
                "ec2:ResetNetworkInterfaceAttribute",
                "ec2:UnassignPrivateIpAddresses"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "autoscaling:*",
            "Resource": "*"
        }
    ]
}
```

2 - CONFIGURING THE LOAD BALANCER TO AUTO ADD/REMOVE AUTO-SCALED REAL SERVERS

If auto-scaling is used, the load balancer must be notified when EC2 instances are either launched or shutdown to ensure that the list of load balanced servers is kept up-to-date. The steps below explain what must be done to achieve this:

Step 1 – Setup the Launch Configuration & Auto-Scaling Group

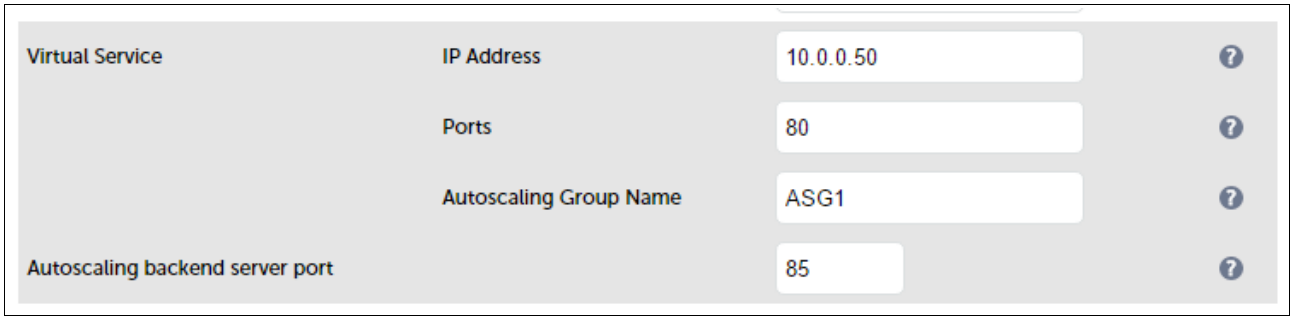
Using the EC2 Dashboard, create your launch configuration and auto-scaling group according to your requirements.

Step 2 – Create the Virtual Service on the Load Balancer

Now create the layer 4 or layer 7 Virtual Service in the normal way. There is no need to manually add the real servers, these will be automatically added once step 3 below is complete.

Step 3 – Associate the Auto-Scaling Group with the Virtual Service

Modify the layer 4 or layer 7 VIP, then in the *Autoscaling Group Name* field specify the Auto-Scaling group created in step 1 as shown in the example below:



> Note:
>
> For Layer 7 VIPs there is an additional field called *Autoscaling backend server port*. This can be used to define the backend server port if it's different from the VIP. This is only used when the autoscaling service adds a new server. If left empty, by default new backend servers will be created using the same port as the VIP.

Now save the updated configuration and restart services as prompted.

> Note:
>
> For more information on configuring Auto-scaling in AWS, please refer to the following URL:
> http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/GettingStartedTutorial.html

3 - CONFIGURING AUTO-SCALING TO AUTO DEPLOY A NEW LB.ORG INSTANCE ON FAILURE

Follow this procedure to configure Auto Scaling for your Loadbalancer.org instance. Once configured, if the load balancer instance is stopped or terminated, auto-scaling will automatically start a new instance with the same settings and configuration. The steps required to set this up are shown below:

Step 1 - Deploy a Load Balancer instance

Launch and configure your Loadbalancer.org instance if not already done so.

Step 2 - Create an image of the instance

This will be the source image when new instances are deployed.

1. Right click the running instance and select: *Image > Create Image*
2. Enter an appropriate name & description for the image – e.g. **AS-LB-Recovery, LB recovery image**
3. Click **Create Image** to start the image creation process

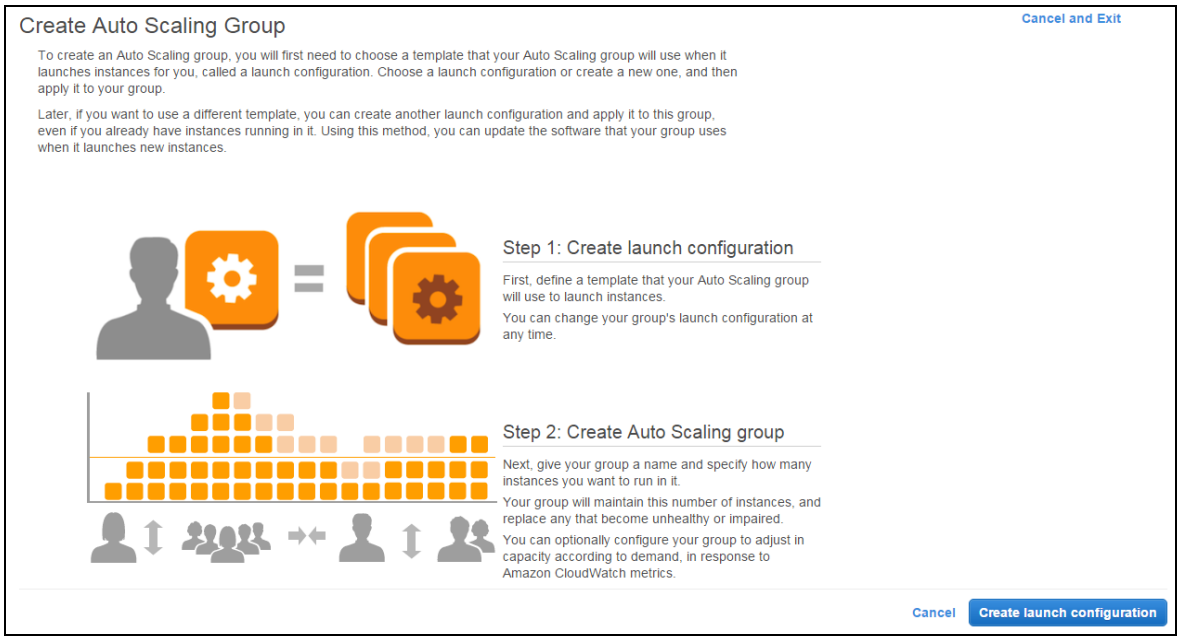
- Image (AMI) creation should be completed in less than 1 minute, creation status can be checked under: *IMAGES > AMIs*

Step 3 - Configure AWS Auto Scaling

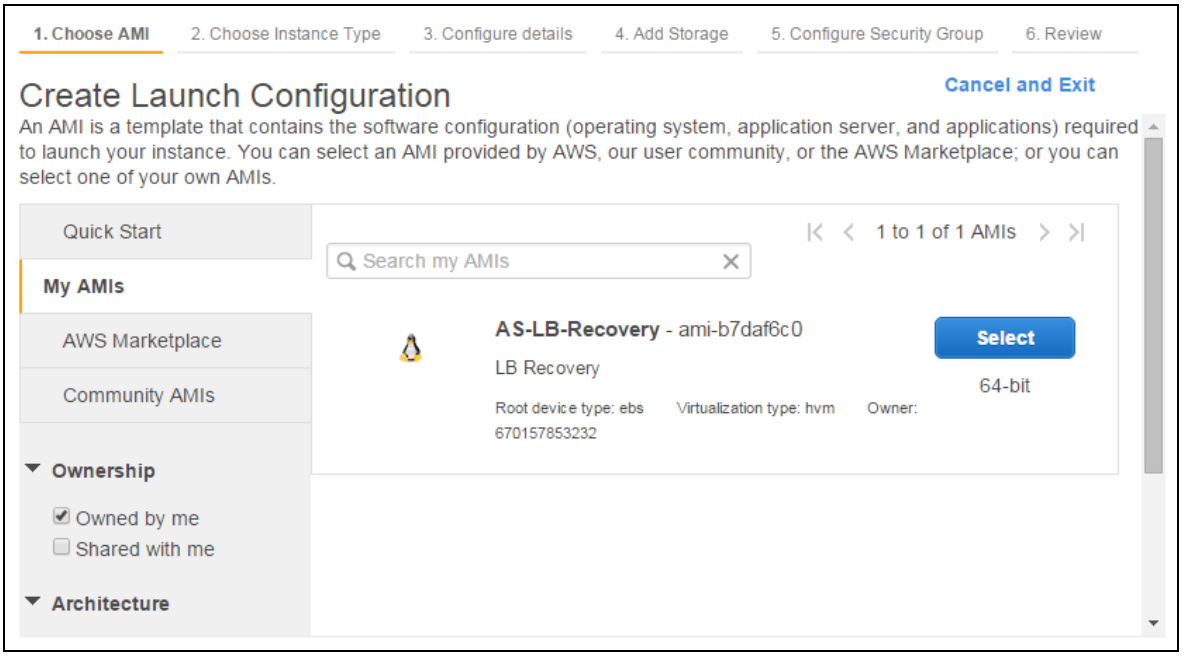
This configuration enables new instances to be automatically started when needed.

1. Under *AUTO SCALING* select **Launch Configurations**
2. Click the **Create Auto Scaling Group** button

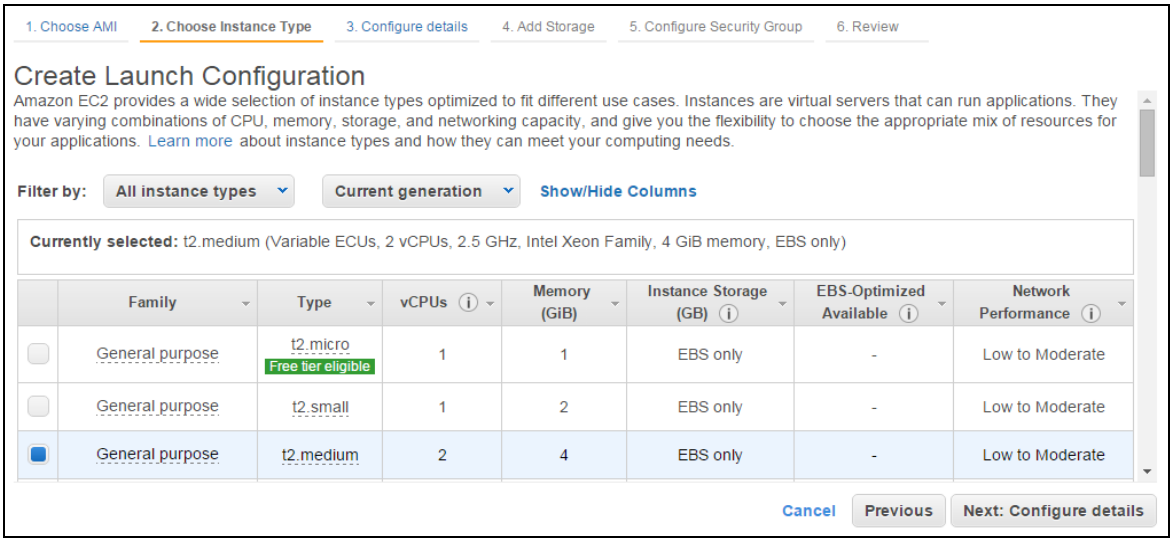
- If no *Launch Configuration* exists, you'll be prompted to create one as shown below (Step 1)



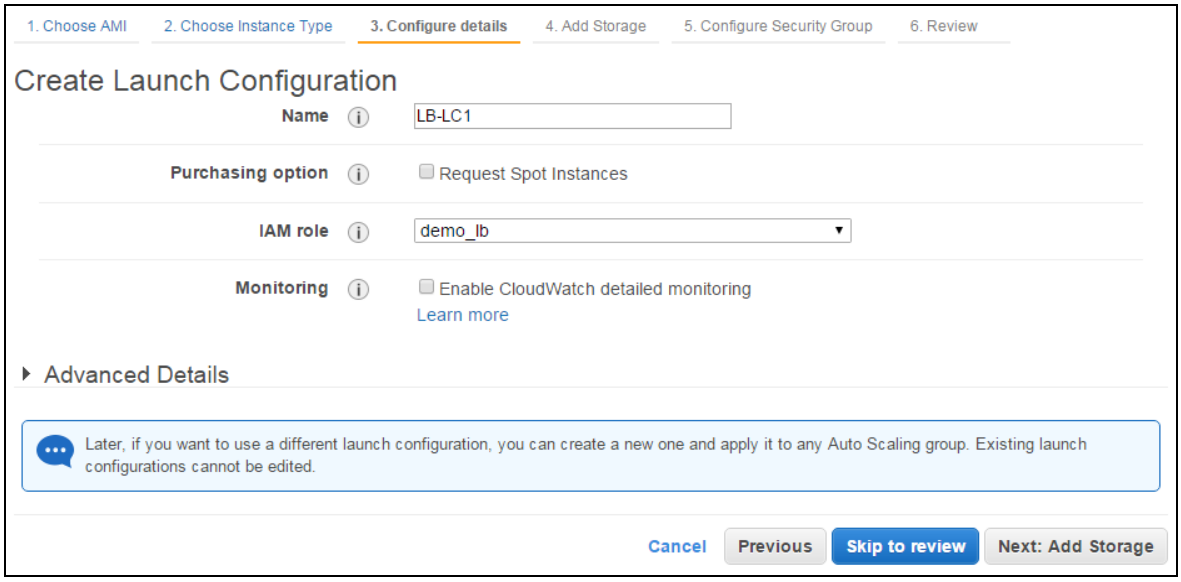
3. Click the **Create Launch Configuration** button
4. Select **My AMIs**



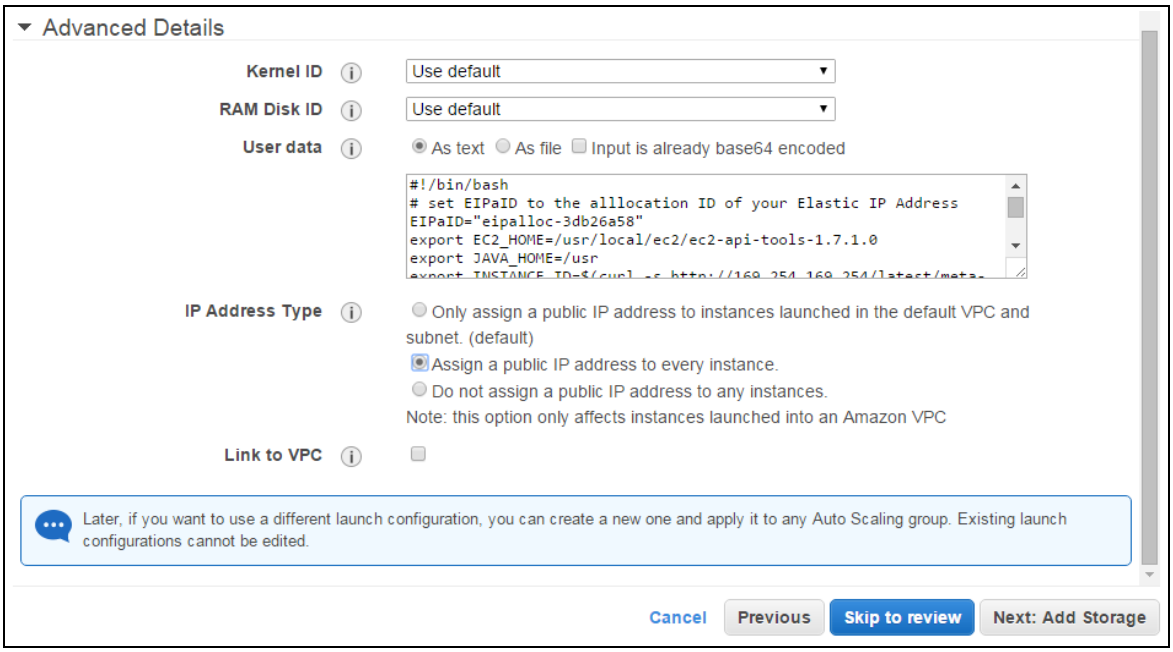
5. Click the **Select** button next to the AMI just created



6. Select the same Instance Type used for the load balancer instance and click **Next: Configure Details**



7. Define a name for the Launch Configuration, e.g. **LB-LC1**
8. Select the same IAM role as was used for the original load balancer
9. To enable the same Elastic IP Address (EIP) to be attached to the new instance, expand the **Advanced Details** section and complete steps a) and b) below:



a) Copy/paste the following script into the *User data* field:

```
#!/bin/bash
        # set EIPid to the allocation ID of your Elastic IP Address
        EIPid="eipalloc-3db26a58"
        # set LBregion to the appropriate region
        Lbregion="eu-west-1"
        export INSTANCE_ID=$(curl -s http://169.254.169.254/latest/meta-data/instance-id)
        export IPV4=$(curl -s http://169.254.169.254/latest/meta-data/local-ipv4)
```

```
/usr/local/bin/aws ec2 ec2-associate-address --instance-id $INSTANCE_ID
--region $LBregion --allow-reassociation --allocation-id $EIPid
--private-ip-address $IPV4 > /var/log/lbas.log 2>&1
```

Note:

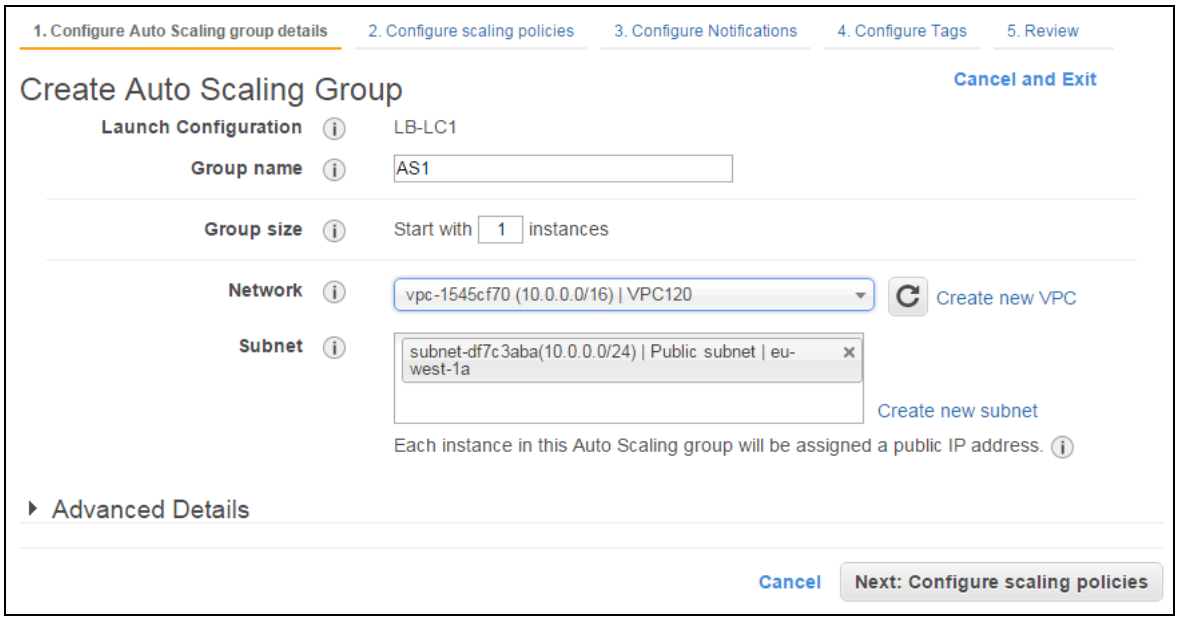
Make the following changes to the above script to suit your environment:

- change **EIPid** in line 3 to the allocation ID of your EIP - this can be found in the lower information pane for the EIP
- change **LBregion** in line 5 to the appropriate region

b) Change *IP Address Type* to **Assign a public IP address to every instance**

Now continue as follows:

10. Click **Next: Add Storage**
11. Click **Next: Configure Security Group**
12. Select the same security group as used for the original load balancer instance
13. Click **Review**
14. Click **Create Launch Configuration**
15. Configure the required *key pair* option
16. Click **Create Launch Configuration**, you'll now be prompted to enter details for the Auto Scaling group:



17. Enter an appropriate *Group name* , e.g. **AS1**
18. Select the correct VPC for your environment
19. Select the correct subnet for your environment
20. If required, expand *Advanced Details* and change the *Health Check grace period* from the 300s default value
21. Click **Next: Configure scaling policies**

22. Leave the option set to *Keep this group at its initial size* and click **Next: Configure Notifications**
23. Configure any required notifications and Click **Next: Configure Tags**
24. Define any required Tags, e.g. **Name = LB - autoscaled** , etc.
25. Click **Review**
26. Click **Create Auto Scaling group**

A new instance will now start automatically. You can now shutdown the original instance

> Note:
>
> The password to access the WebUI will be the instance-id of the source instance, not the new auto scaled instance.

Testing

You can now test the new indestructible instance using the Amazon Web Management Console. Simply stop the instance, the auto-scaling configuration should then start a brand new copy of the instance.

Completely Terminating the Instance

Do not simply terminate the instance using the console, this will cause another replacement instance to automatically start. You'll need to delete the Auto Scaling group. This will also terminate any associated instances.

4 - 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