Enterprise AWS Quick Start Guide

v8.1.1

rev. 1.0.2

Table of Contents

Introduction....4
About Enterprise AWS....4
Main Differences to the Non-Cloud Product....4
Why use Enterprise AWS?....5
Amazon Terminology....5
Getting Started....5
Deployment Concepts....6
Overview....6
VPC Wizard Setup....6
VPC IP Address Types....7
VPC Network Interfaces (ENI)....7
10GB Support....7
Instance Type....8
Deploying Enterprise AWS....8
STEP A - Create a VPC....8
STEP B – Accessing & Deploying the AMI....10
Checking your Subscriptions....15
Accessing the Appliance....15
Accessing the Appliance using the WUI....15
WUI Menu Options....16
Accessing the Appliance using SSH....19
Using Linux....19
Using Windows....19
Accessing the Appliance using SCP....22
Using Linux....22
Using Windows....22
Configuration Examples....23
Deployment Notes....23
1) Load balancing Web Servers – 1 Subnet, 1 Interface, Layer 7....24
a) Setting up AWS....24
b) Setting up the Virtual Service....24
c) Setting up the Real Servers....24
d) Applying the new Layer 7 Settings....25
e) Associating the VIP with an Elastic IP Address (If access from the Internet is required)....25
2) Load balancing Web Servers - 1 Subnet, 1 Interface, Layer 4....26
a) Setting up AWS....26
b) Setting up the Virtual Service....26
c) Setting up the Real Servers....27
d) Associating the VIP with an Elastic IP Address (If access from the Internet is required)....27
e) Enable Internet Connectivity via the Load Balancer for the Real Servers (If Required)....28
3) Load balancing Web Servers – 2 Subnets, 2 Interfaces, Layer 7....29
a) Setting up AWS....29
b) Configuring the second Network Interface....29
c) Setting up the Virtual Service....29
d) Setting up the Real Servers....30
e) Applying the new Layer 7 Settings....30
f) Associating the VIP with an Elastic IP Address (If access from the Internet is required)....31
4) Load balancing Web Servers - 2 Subnets, 1 Interface, Layer 7 with Transparency....32
a) Setting up AWS....32
b) Setting up the Virtual Service....32
c) Setting up the Real Servers....33
d) Configuring Layer 7 – Advanced Settings....34
e) Applying the new Layer 7 Settings....34

f) Associating the VIP with an Elastic IP Address (If access from the Internet is required)....34
5) Load balancing Web Servers - 1 Subnet, 1 Interface, Layer 7 with SSL Termination....35
a) Setting up AWS....35
b) Setting up the Virtual Service....35
c) Setting up the Real Servers....35
d) Configuring SSL Termination....36
e) Applying the new Settings....37
f) Associating the VIP with an Elastic IP Address (If access from the Internet is required)....37
6) Load balancing RD Session Hosts - 2 Subnets, 1 Interface, Layer 7....39
a) Setting up AWS....39
b) Setting up the Virtual Service....39
c) Setting up the Real Servers....40
d) Applying the new Layer 7 Settings....41
e) Associating the VIP with an Elastic IP Address (If access from the Internet is required)....41
7) Load balancing Web Servers - 2 Subnets, 1 Interface, Layer 4....42
a) Setting up AWS....42
b) Setting up the Virtual Service....42
c) Setting up the Real Servers....43
d) Associating the VIP with an Elastic IP Address (If access from the Internet is required)....43
e) Enable Internet Connectivity via the Load Balancer for the Real Servers (If Required)....44
Testing & Validation....45
Testing Load Balanced Services....45
Diagnosing VIP Connection Problems....45
Taking Real Servers Offline....46
Using Reports & Log Files....47
Configuring High Availability using two Instances (Master & Slave)....48
Loadbalancer.org Technical Support....51
Appendix....52
1 - IAM Role Configuration....52
2 - Configuring the load balancer to auto add/remove auto-scaled Real Servers....53
3 - Configuring Auto-Scaling to auto deploy a new LB.org Instance on Failure....54
4 - Company Contact Information....59

Introduction

Amazon Web Services (AWS) provides a cloud based platform to deploy web services. 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.

About Enterprise AWS

The core software is based on customized versions of Centos 6 / RHEL 6, Linux 3.10, 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 48.

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 32 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 WUI option: *Cluster Configuration > Heartbeat Advanced,* and the AWS command **ec2-replace-route** as detailed on page 48
- 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 42 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 WUI option: *Cluster Configuration > Heartbeat Advanced,* and the AWS command **ec2-replace-route** as detailed on page 48
- The WUI 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

Amazon Terminology

Acronym	Terminology
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

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/

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.

VPC Wizard Setup

When using the wizard to configure a VPC 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 public subnet using Network Address Translation (NAT).	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 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.)

N.B. 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 no longer usable upon instance termination.

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 : From v8.1.1 it's possible to create Virtual Services (VIPs) 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.

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 48.

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 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/

Deploying Enterprise AWS

STEP A - Create a VPC

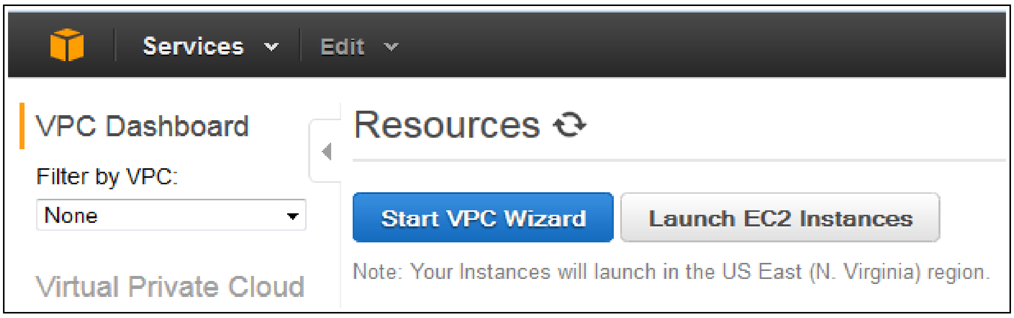
For a manually created VPC, the key steps are:

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:

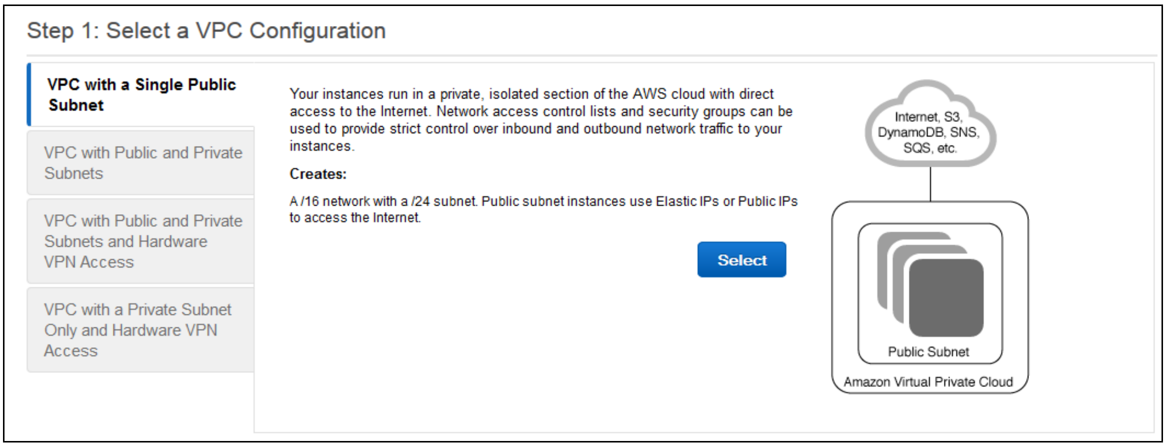
- In the VPC dashboard, click **Start VPC Wizard**



Note: Your Instances will launch in the US East (N. Virginia) region.

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

  N.B. 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.



Step 1: Select a VPC Configuration

VPC with a Single Public Subnet

VPC with Public and Private Subnets

VPC with Public and Private Subnets and Hardware VPN Access

VPC with a Private Subnet Only and Hardware VPN Access

Your 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.

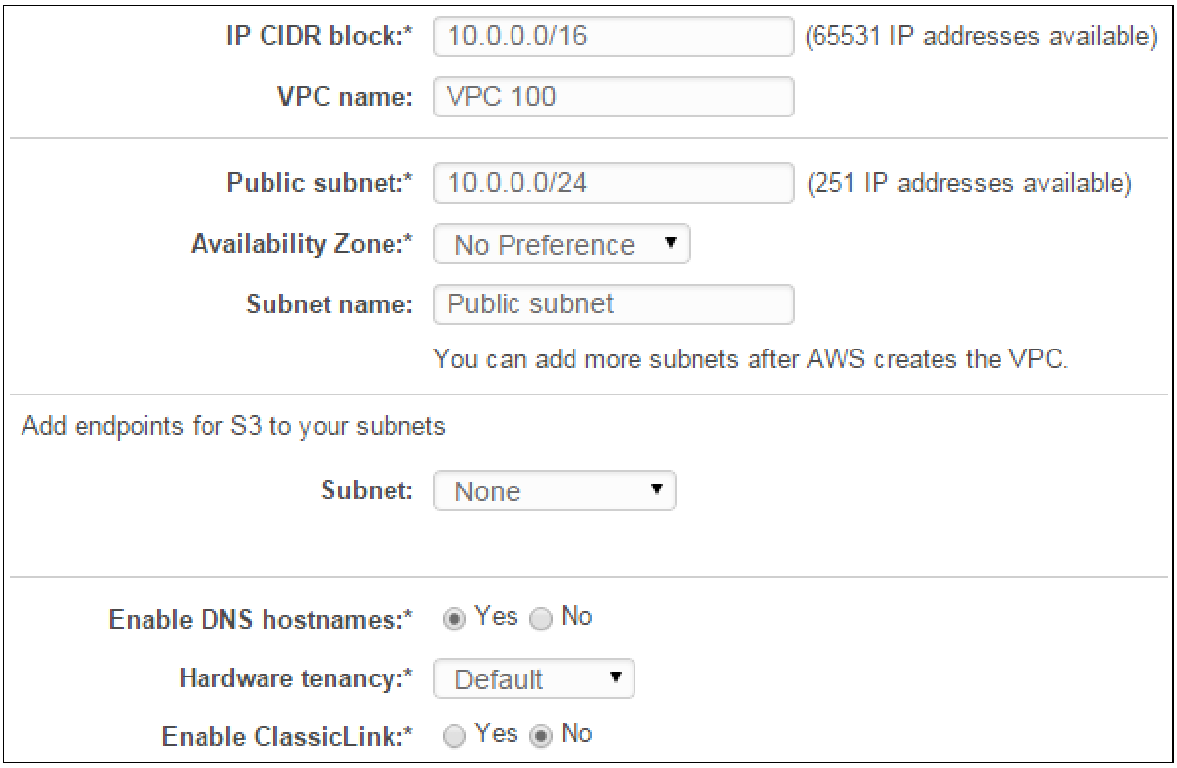
Creates:

A /16 network with a /24 subnet. Public subnet instances use Elastic IPs or Public IPs to access the Internet.

Select

Internet, S3, DynamoDB, SNS, SQS, etc.
Public Subnet
Amazon Virtual Private Cloud

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



IP CIDR block:* 10.0.0.0/16 (65531 IP addresses available)

VPC name: VPC 100

Public subnet:* 10.0.0.0/24 (251 IP addresses available)

Availability Zone:* No Preference ▾

Subnet name: Public subnet

You can add more subnets after AWS creates the VPC.

Add endpoints for S3 to your subnets

Subnet: None ▾

Enable DNS hostnames:* ● Yes ○ No

Hardware tenancy:* Default ▾

Enable ClassicLink:* ○ Yes ● No

- Click **Create VPC**

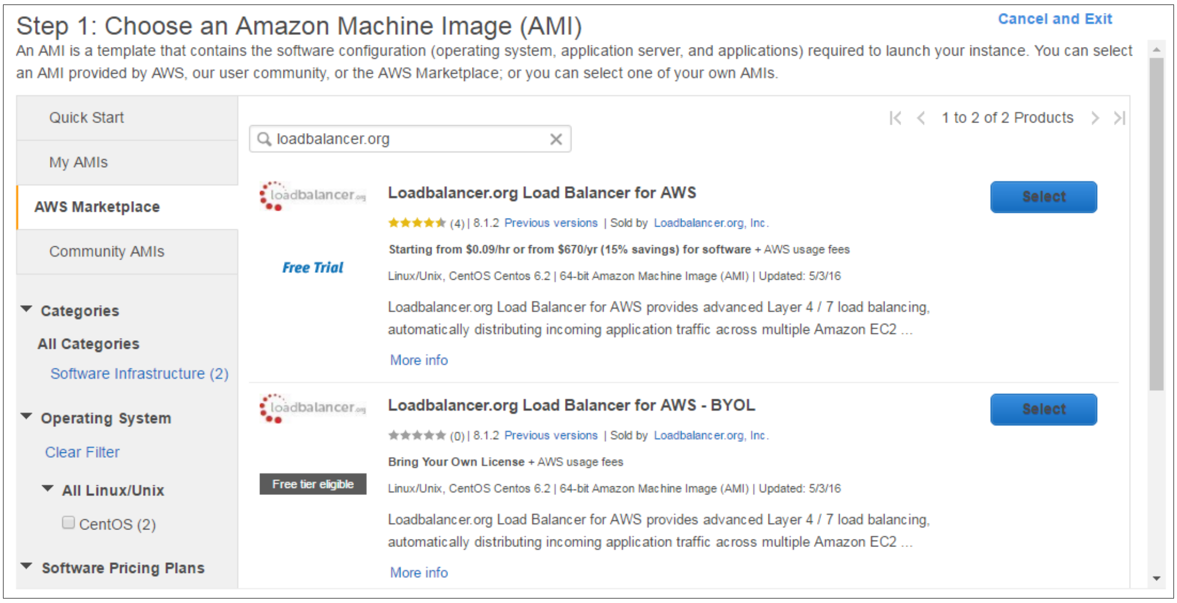
N.B. For more details on Amazon's VPC, please refer to their comprehensive user guide available at the following URL : http://awsdocs.s3.amazonaws.com/VPC/latest/vpc-ug.pdf

STEP B – Accessing & Deploying the AMI

NOTE : From v8.1.1 configuring an **IAM role** for the instance is optional. However, we still strongly 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.

To access and deploy the AMI:

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



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

loadbalancer.org

1 to 2 of 2 Products

Loadbalancer.org Load Balancer for AWS

★★★★★ (4) | 8.1.2 Previous versions | Sold by Loadbalancer.org, Inc.

Starting from $0.09/hr or from $670/yr (15% savings) for software + AWS usage fees

Linux/Unix, CentOS Centos 6.2 | 64-bit Amazon Machine Image (AMI) | Updated: 5/3/16

Loadbalancer.org Load Balancer for AWS provides advanced Layer 4 / 7 load balancing, automatically distributing incoming application traffic across multiple Amazon EC2 ...

More info

Loadbalancer.org Load Balancer for AWS - BYOL

★★★★★ (0) | 8.1.2 Previous versions | Sold by Loadbalancer.org, Inc.

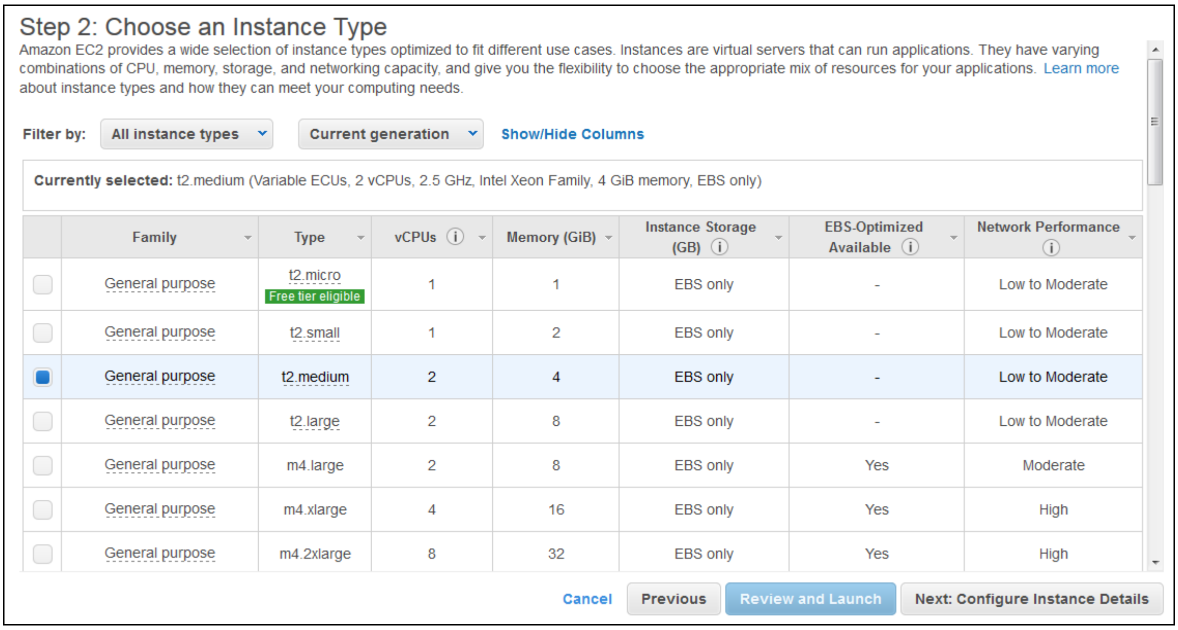
Bring Your Own License + AWS usage fees

Linux/Unix, CentOS Centos 6.2 | 64-bit Amazon Machine Image (AMI) | Updated: 5/3/16

Loadbalancer.org Load Balancer for AWS provides advanced Layer 4 / 7 load balancing, automatically distributing incoming application traffic across multiple Amazon EC2 ...

More info

- Click **Select** next to the required instance (either hourly billing or BYOL)
- Select the required pricing options (hourly or annual)
- Click the **Launch with EC2 Console** button next to the required Region
- Select the required instance type – **t2.medium** is recommended, but depends on your requirements



Step 2: Choose an Instance Type

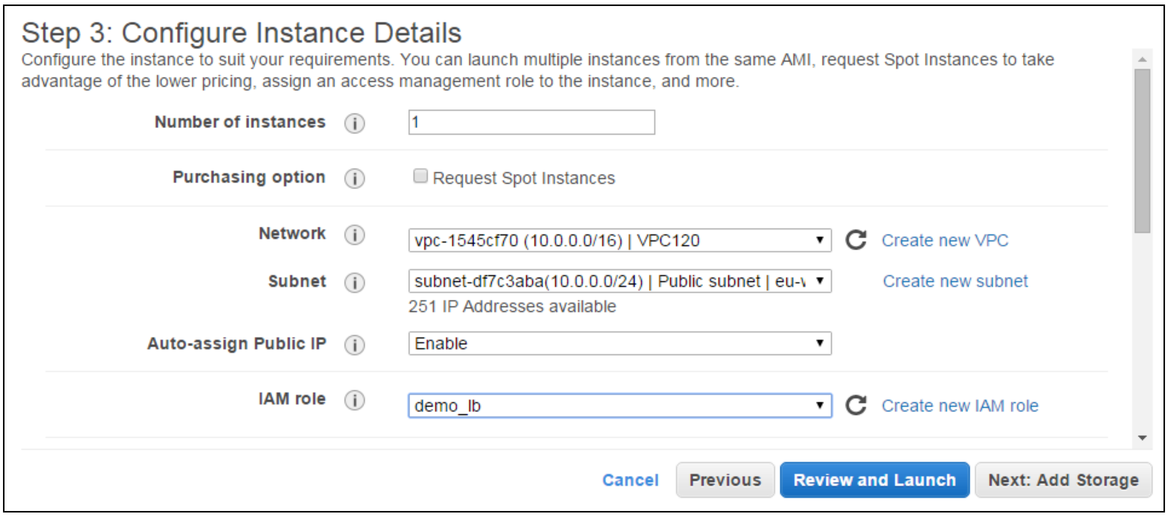
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance types | Current generation | Show/Hide Columns

Currently selected: t2.medium (Variable ECUs, 2 vCPUs, 2.5 GHz, Intel Xeon Family, 4 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
□	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
☑	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
□	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
□	General purpose	m4.large	2	8	EBS only	Yes	Moderate
□	General purpose	m4.xlarge	4	16	EBS only	Yes	High
□	General purpose	m4.2xlarge	8	32	EBS only	Yes	High

- Click **Next: Configure Instance Details**



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1

Purchasing option: □ Request Spot Instances

Network: vpc-1545cf70 (10.0.0.0/16) | VPC120

Subnet: subnet-df7c3aba(10.0.0.0/24) | Public subnet | eu-\
251 IP Addresses available

Auto-assign Public IP: Enable

IAM role: demo_lb

- 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”

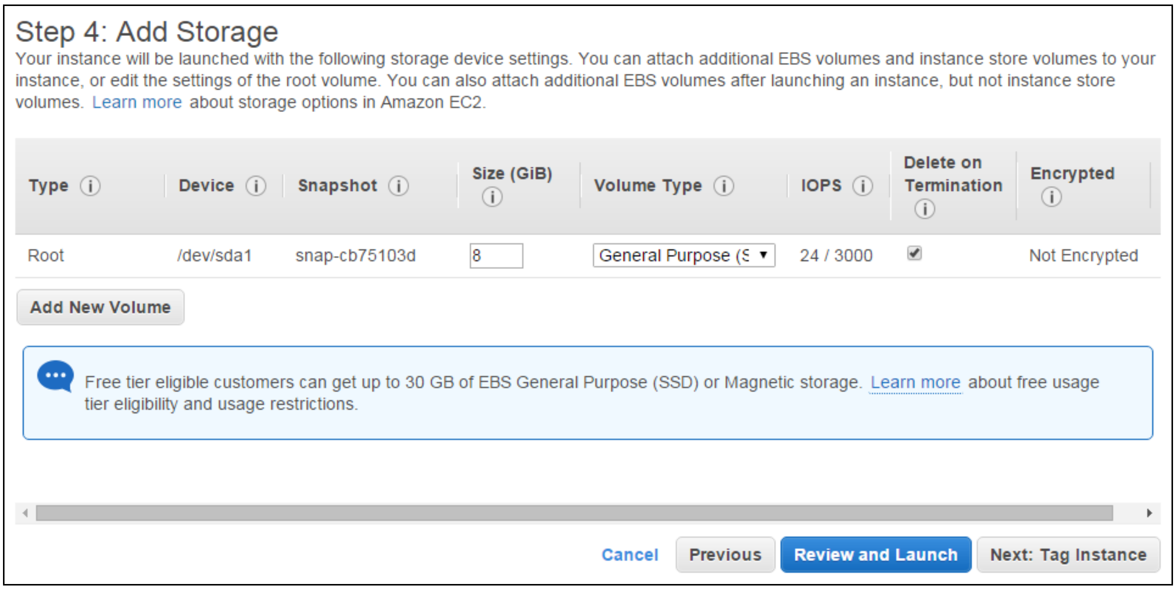
NOTE : If you're configuring an HA pair (2 instances) you'll need to assign EIP's to the primary interface of each appliance.

- 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.

NOTE : From v8.1.1 configuring an **IAM role** for the instance is optional. However, we still strongly 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.

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 & 6 on pages 32 and 39 respectively for more details.*

- Click **Next: Add Storage**



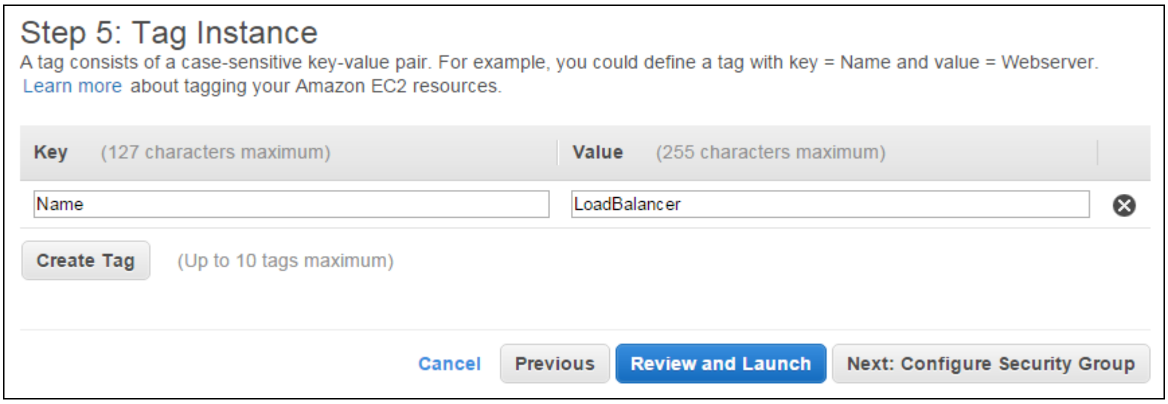
Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Type ⓘ	Device ⓘ	Snapshot ⓘ	Size (GiB) ⓘ	Volume Type ⓘ	IOPS ⓘ	Delete on Termination ⓘ	Encrypted ⓘ
Root	/dev/sda1	snap-cb75103d	8	General Purpose (S ▾	24 / 3000	☑	Not Encrypted

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

- Leave the defaults and click **Next: Tag Instance**



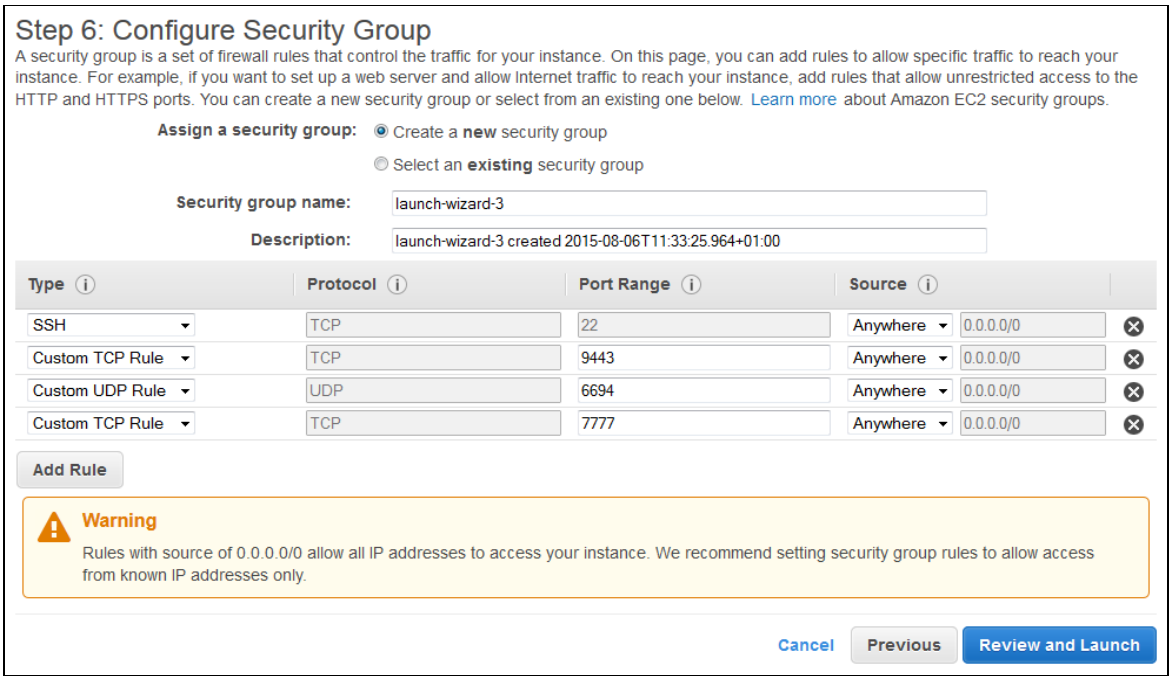
Step 5: Tag Instance

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)
Name	LoadBalancer

(Up to 10 tags maximum)

- Enter a suitable name for the instance and click **Next: Configure Security Group**



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ● Create a **new** security group

○ Select an **existing** security group

Security group name: launch-wizard-3

Description: launch-wizard-3 created 2015-08-06T11:33:25.964+01:00

Type	Protocol	Port Range	Source	
SSH	TCP	22	Anywhere	0.0.0.0/0
Custom TCP Rule	TCP	9443	Anywhere	0.0.0.0/0
Custom UDP Rule	UDP	6694	Anywhere	0.0.0.0/0
Custom TCP Rule	TCP	7777	Anywhere	0.0.0.0/0

Warning

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

- We recommend that at least the rules shown above and listed below are configured. These are required to enable management & monitoring access to the load balancer.

  SSH – TCP port 22

  WUI – TCP port 9443

  Heartbeat – UDP port 6694

  HAProxy Statistics - TCP port 7777

- Additional rules can be added as needed to provide access to the application(s) being load balanced:

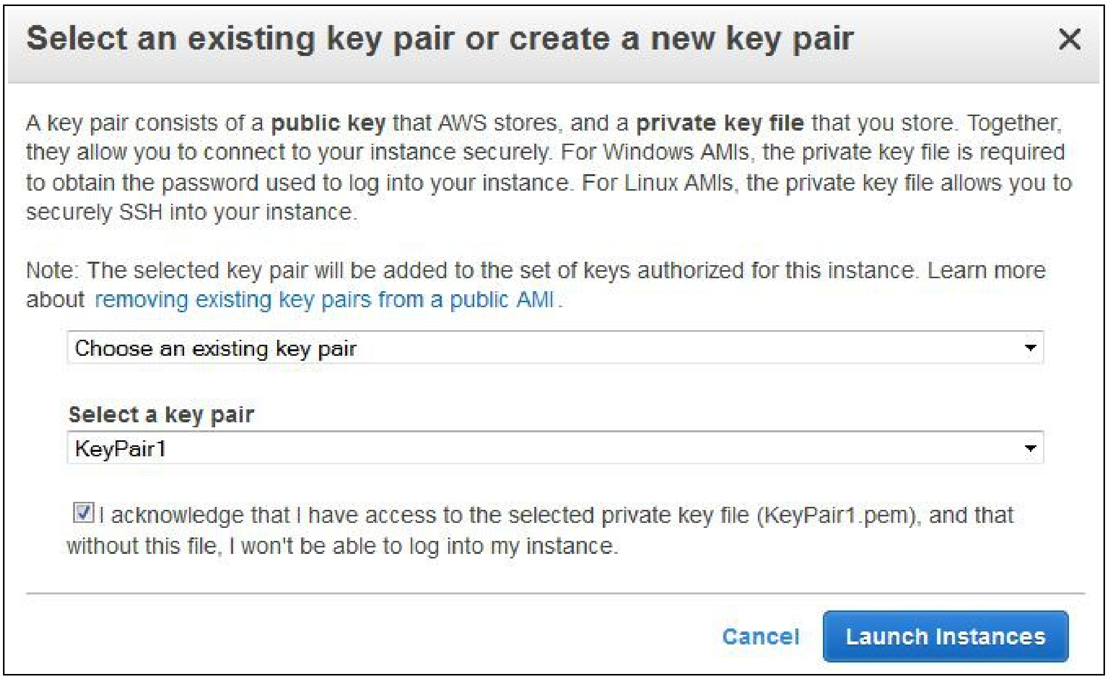
  e.g. If you're load balancing HTTP & HTTPS traffic, add TCP ports 80 & 443

  e.g. If you're load balancing RDP traffic, add TCP port 3389

  etc.

- Click **Review and Launch**

- Check all settings and click **Launch**



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Choose an existing key pair

Select a key pair: KeyPair1

☑ I acknowledge that I have access to the selected private key file (KeyPair1.pem), and that without this file, I won't be able to log into my instance.

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

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

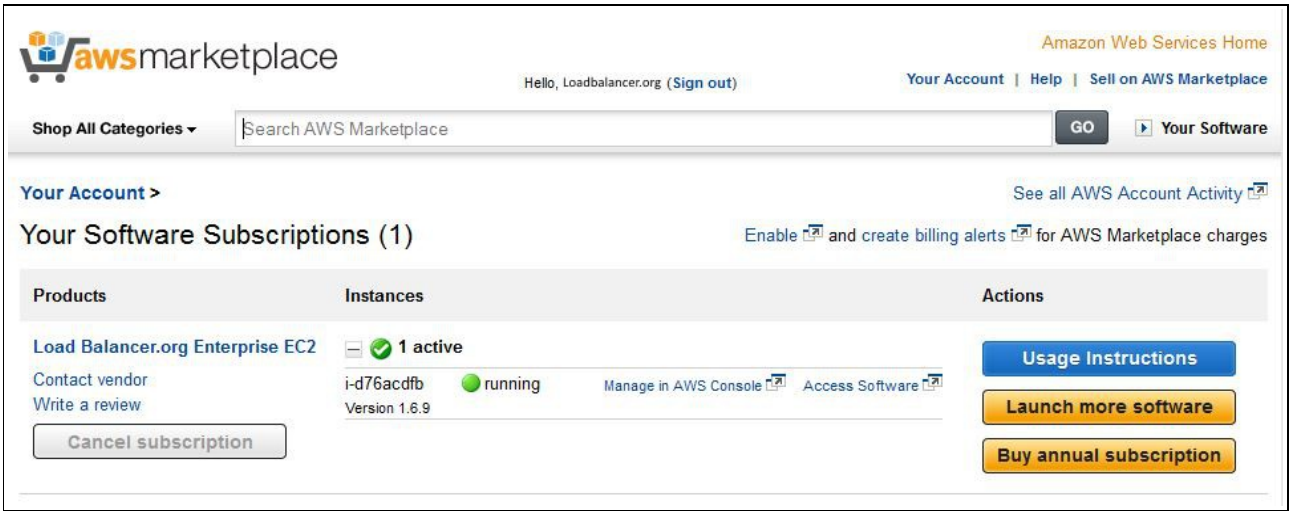
- If using an existing key pair, check (tick) the acknowledgment check-box
- Click the **Launch Instances** button

IMPORTANT!! **-** once the instance is running, right-click the instance and select: ***Networking > Change Source/Dest. Check*** and ensure this is disabled.

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 Account > Your Software > Manage your Software Subscriptions* option in the awsmarketplace console as shown below:



Your Software Subscriptions (1)

Products	Instances				Actions
Load Balancer.org Enterprise EC2 Contact vendor Write a review	1 active				Usage Instructions Launch more software Buy annual subscription
Cancel subscription	i-d76acdfb Version 1.6.9	running	Manage in AWS Console	Access Software	

Accessing the Appliance

Accessing the Appliance using the WUI

In a browser, navigate to the Public DNS name or Public IP address port 9443

i.e.

https://<Public DNS name>:9443

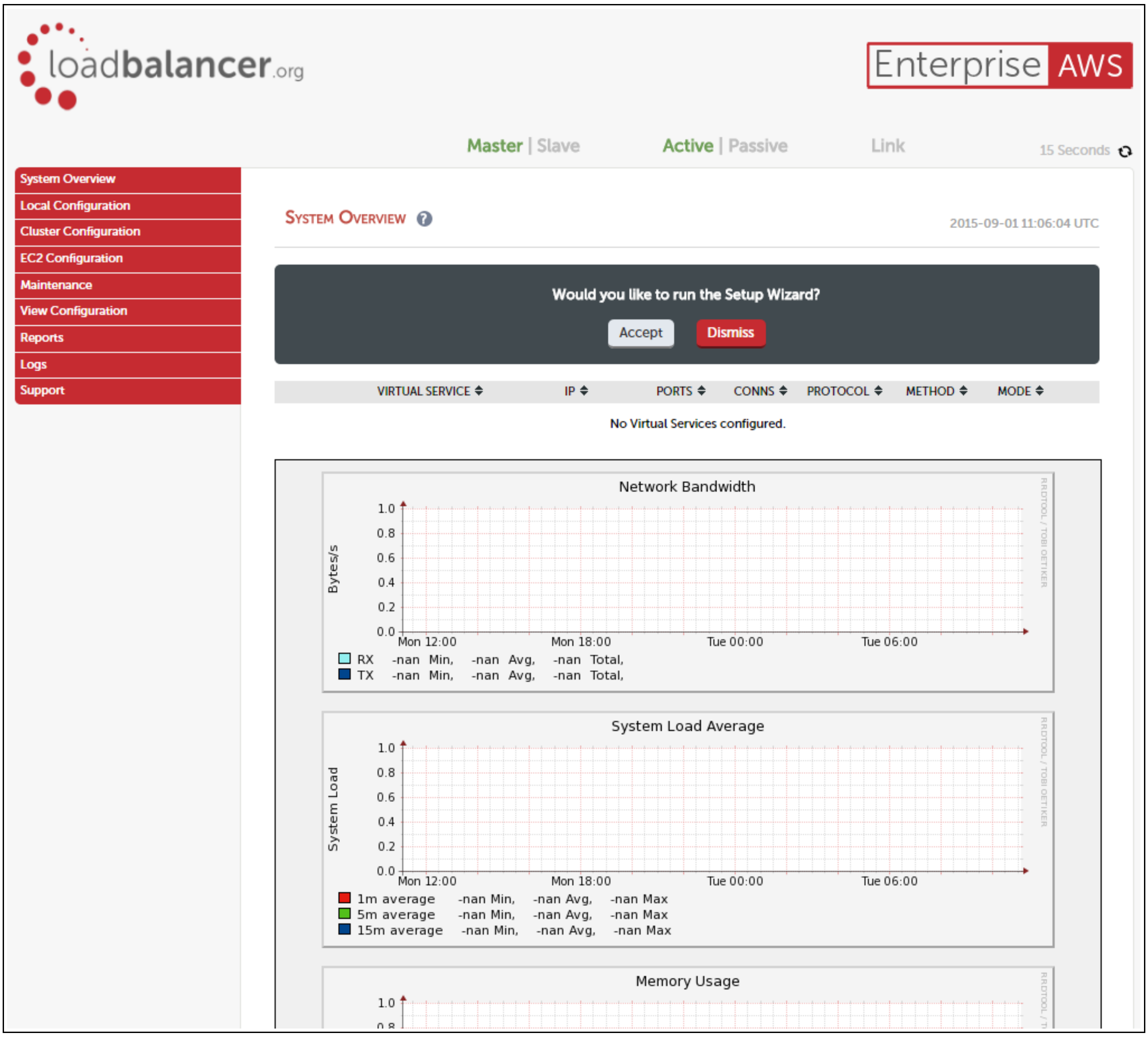
or

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>*

Once logged in, the following screen is displayed:



SYSTEM OVERVIEW

2015-09-01 11:06:04 UTC

Would you like to run the Setup Wizard?

VIRTUAL SERVICE	IP	PORTS	CONNS	PROTOCOL	METHOD	MODE

No Virtual Services configured.

```
RX  -nan Min,  -nan Avg,  -nan Total,
TX  -nan Min,  -nan Avg,  -nan Total,
```

```
1m average   -nan Min,  -nan Avg,  -nan Max
5m average   -nan Min,  -nan Avg,  -nan Max
15m average  -nan Min,  -nan Avg,  -nan Max
```

WUI 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

Maintenance – Perform maintenance tasks such as service restarts and taking backups

View Configuration – Display the saved appliance configuration settings

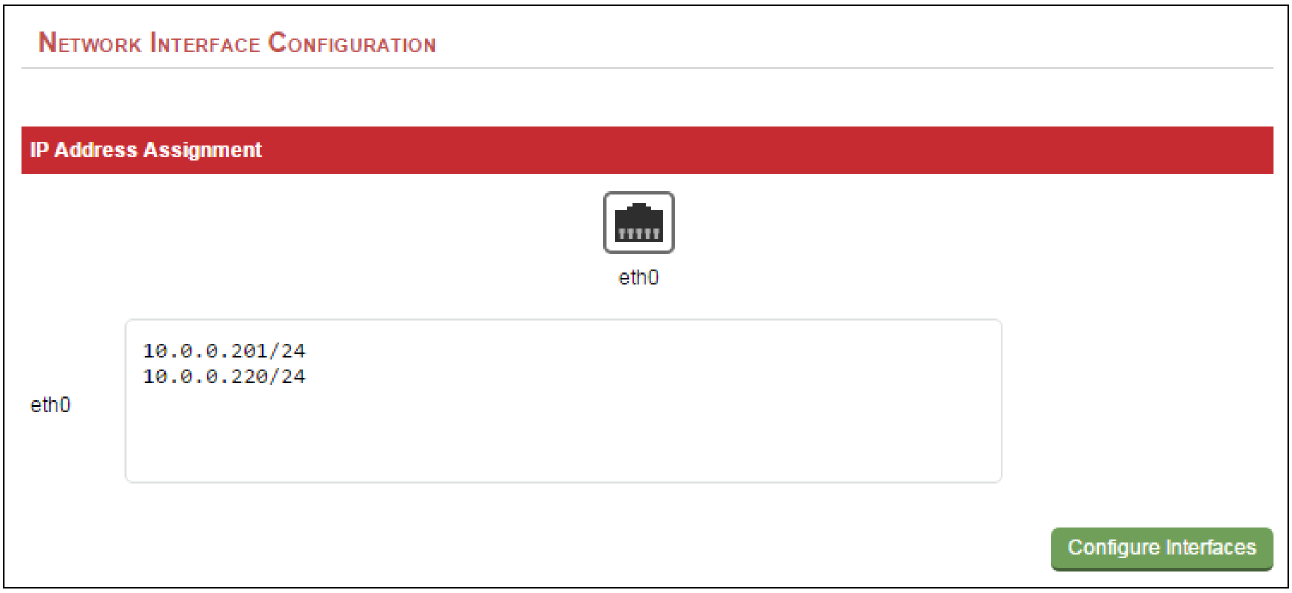
Reports – View various appliance reports & graphs

Logs – View various appliance logs

Support – Create a support download & contact the support team

The following sections detail the menu options that differ from our main product. For all others please refer to our main administration manual : http://pdfs.loadbalancer.org/loadbalanceradministrationv8.pdf

Local Configuration > Network Interface Configuration



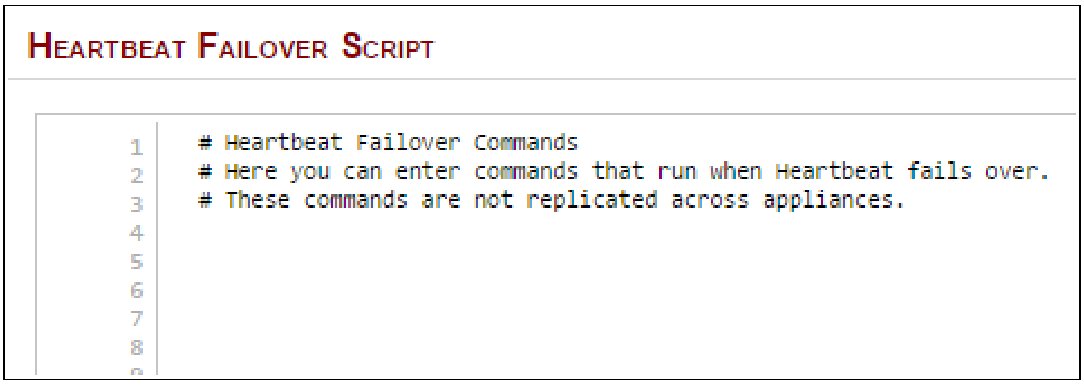
```
10.0.0.201/24
10.0.0.220/24
```

Notes:

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

N.B. Adding additional floating IP's under Cluster Configuration > Floating IP's will be shown as Secondary Private IP's in the EC2 Dashboard

Cluster Configuration > Heartbeat Advanced



HEARTBEAT FAILOVER SCRIPT

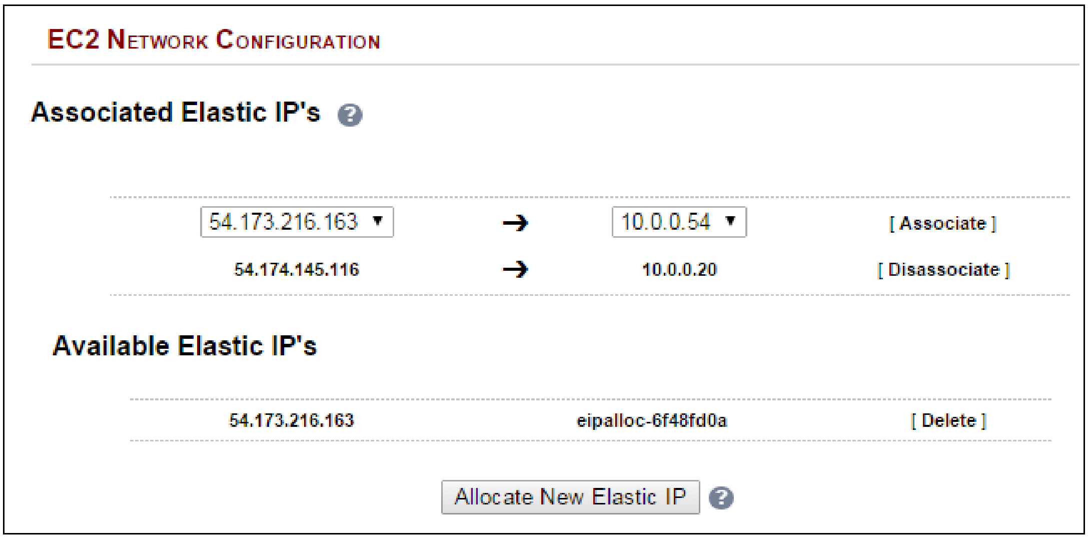
```
# Heartbeat Failover Commands
# Here you can enter commands that run when Heartbeat fails over.
# These commands are not replicated across appliances.
```

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 the following link:

  http://docs.aws.amazon.com/AWSEC2/latest/CommandLineReference/command-reference.html

EC2 Configuration > EC2 Network Configuration



EC2 NETWORK CONFIGURATION

Associated Elastic IP's

54.173.216.163 ▾	→	10.0.0.54 ▾	[Associate]
54.174.145.116	→	10.0.0.20	[Disassociate]

Available Elastic IP's

54.173.216.163	eipalloc-6f48fd0a	[Delete]

Notes:

- This menu option is used to define how Elastic IP's relate to private IP's
- Row-1 above shows EIP 54.173.216.163 and a proposed mapping to private IP 10.0.0.54. If you want to confirm the mapping, click **[Associate]**

  Row-2 above shows that EIP 54.174.145.116 is mapped to private IP 10.0.0.20. If you want to undo the mapping click **[Disassociate]**

  Row-3 above shows that EIP 54.173.216.163 is currently an available Elastic IP. To delete the EIP click **[Delete]**
- New Elastic IP's can be allocated by clicking **Allocate New Elastic IP**. Newly created EIP's will be displayed in the 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.

Accessing the Appliance using SSH

This uses the private key that you downloaded when setting up your instance (please refer to page 13 of this guide). To connect to the load balancer using SSH, this private key 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.

N.B. 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/ec2-key-name.pem
```

# Now connect via SSH specifying the private key file – login as user 'lbuser'

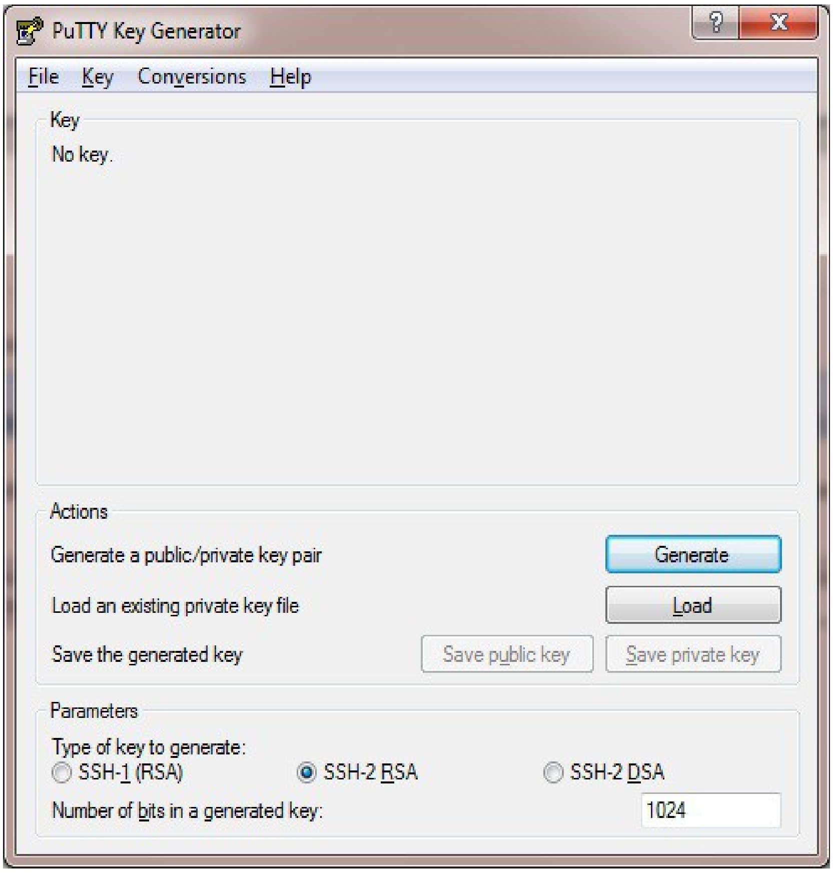
```
ssh -i /path-where-saved/ec2-key-name.pem lbuser@1.2.3.4
```

or

```
ssh -i /path-where-saved/ec2-key-name.pem lbuser@dns-name
```

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:



No key.

Generate a public/private key pair

Load an existing private key file

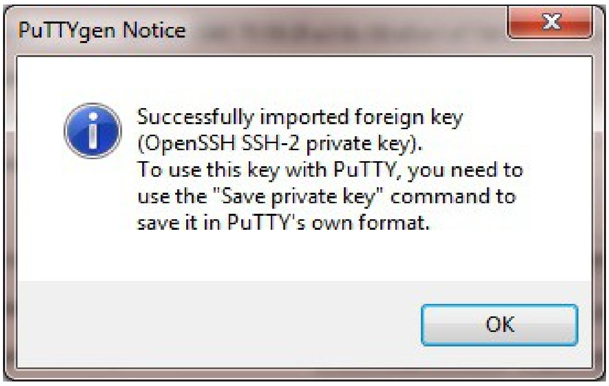
Save the generated key

Type of key to generate:
○ SSH-1 (RSA) ● SSH-2 RSA ○ SSH-2 DSA

Number of bits in a generated key: 1024

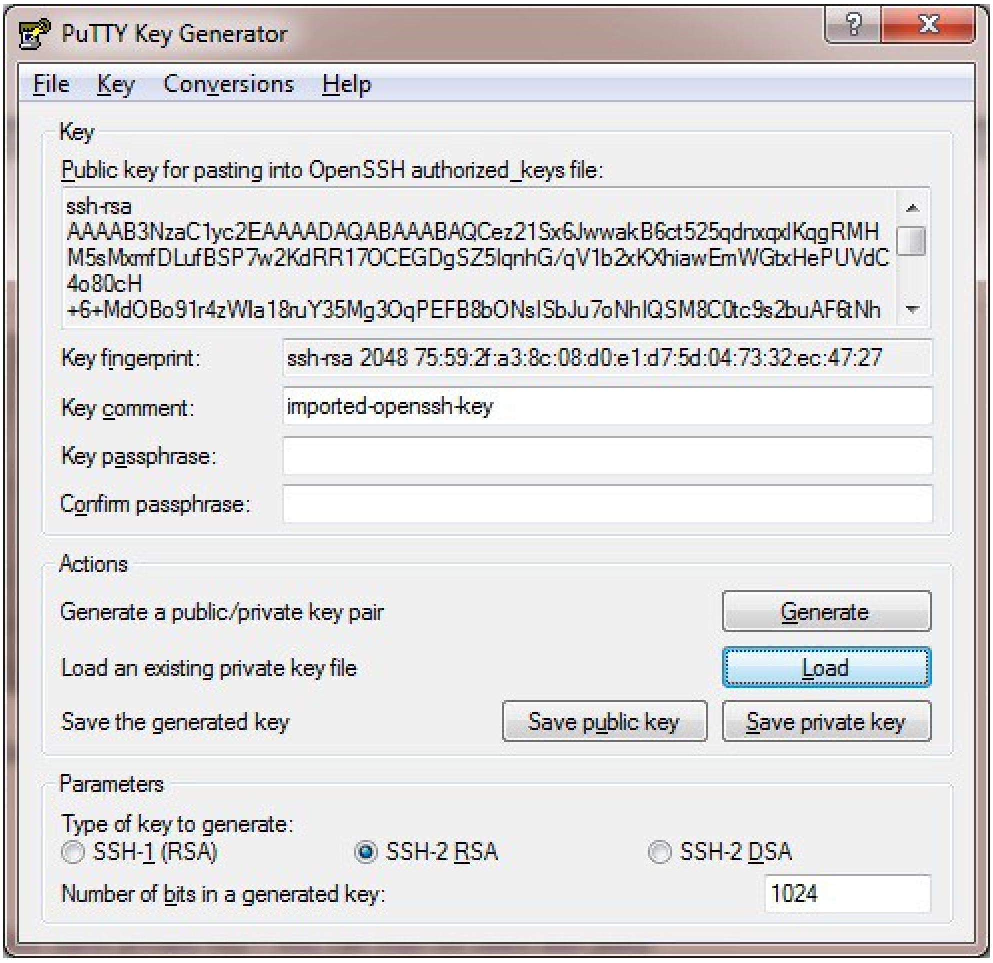
Click **Load**, change the file-type to all files and select the pem file saved earlier when creating your Key Pair.

You should see the following message:



Successfully imported foreign key (OpenSSH SSH-2 private key).
To use this key with PuTTY, you need to use the "Save private key" command to save it in PuTTY's own format.

Click **OK**



Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCez21Sx6JwwakB6ct525qdnxqdKqgRMH
M5sMxmfDLufBSP7w2KdRR17OCEGDgSZ5lqnhG/qV1b2xKXhiawEmWGtxHePUVdC
4o80cH
+6+MdOBo91r4zWla18ruY35Mg3OqPEFB8bONsISbJu7oNhIQSM8C0tc9s2buAF6tNh
```

Key fingerprint: ssh-rsa 2048 75:59:2f:a3:8c:08:d0:e1:d7:5d:04:73:32:ec:47:27

Key comment: imported-openssh-key

Key passphrase:

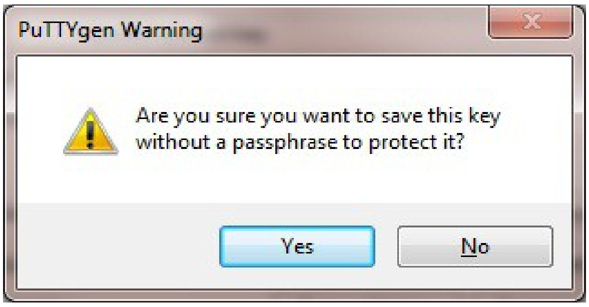
Confirm passphrase:

Type of key to generate:
○ SSH-1 (RSA) ● SSH-2 RSA ○ SSH-2 DSA

Number of bits in a generated key: 1024

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

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

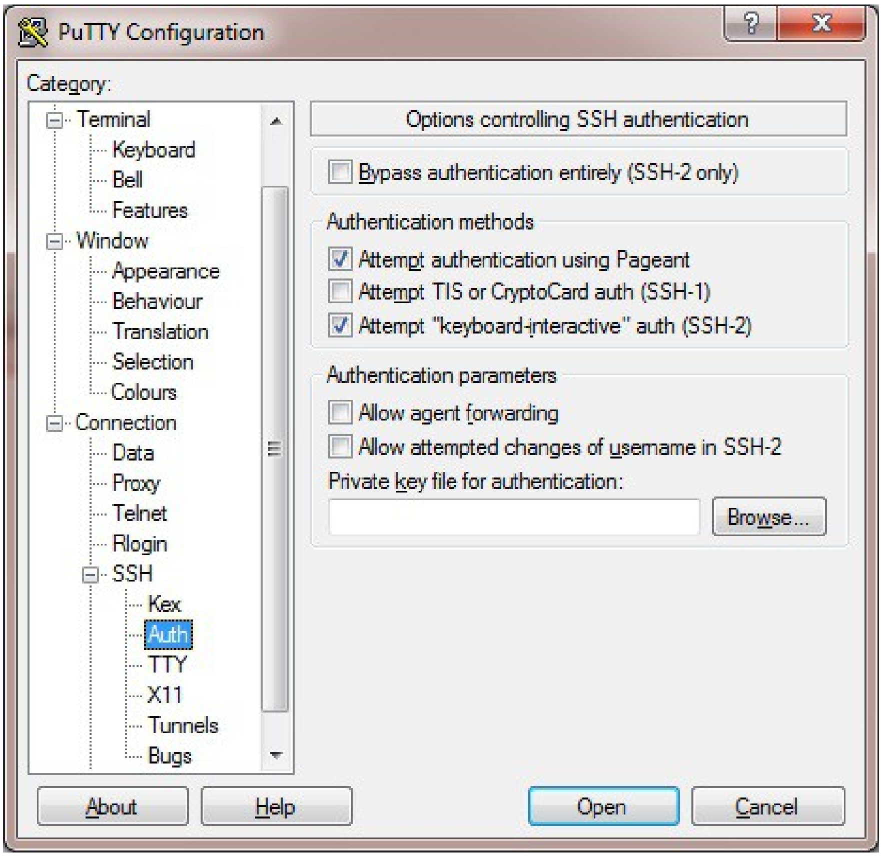

PuTTYgen Warning

Are you sure you want to save this key without a passphrase to protect it?

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

Now close PuTTYgen and start PuTTY

Expand the SSH section as shown below:


PuTTY Configuration

Options controlling SSH authentication

□ Bypass authentication entirely (SSH-2 only)

Authentication methods

☑ Attempt authentication using Pageant

□ Attempt TIS or CryptoCard auth (SSH-1)

☑ Attempt "keyboard-interactive" auth (SSH-2)

Authentication parameters

□ Allow agent forwarding

□ Allow attempted changes of username in SSH-2

Private key file for authentication:

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

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

Accessing the Appliance using SCP

Using Linux

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

```
chmod 400 /path-where-saved/ec2-key-name.pem
```

# Now start SCP specifying the private key file – login as user 'lbuser'

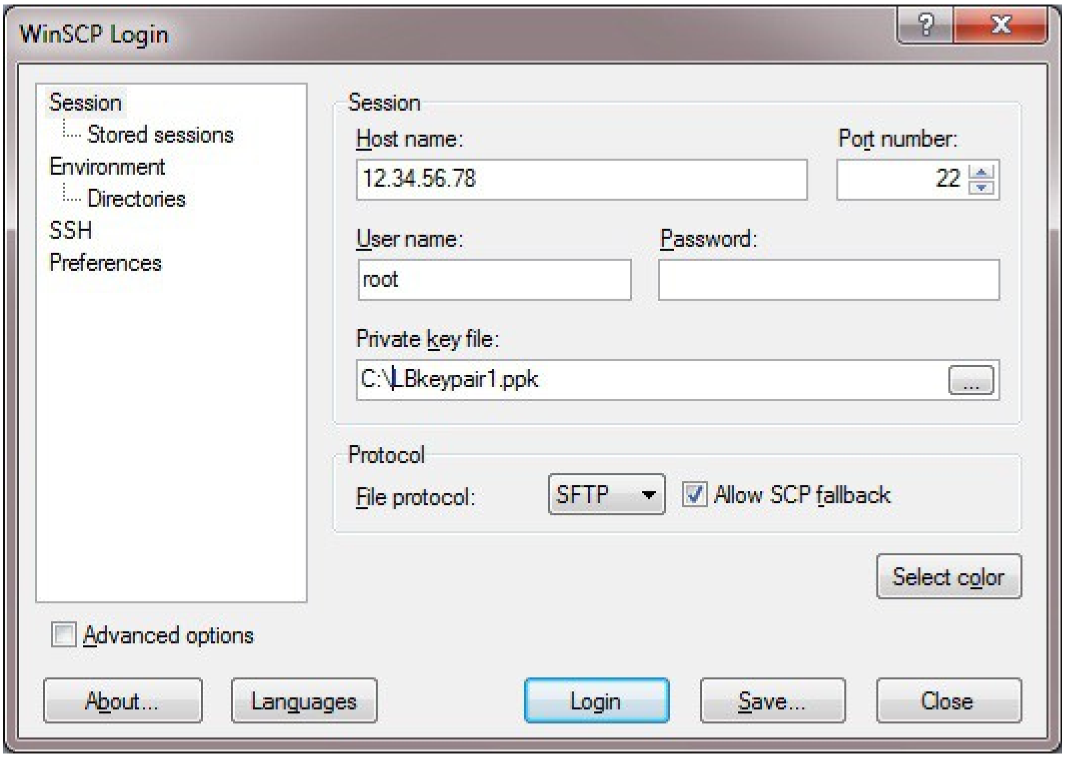
```
scp -i /path-where-saved/ec2-key-name.pem <local-file> lbuser@1.2.3.4:<remote-file>
```

or

```
scp -i /path-where-saved/ec2-key-name.pem <local-file> lbuser@dns-name:<remote-file>
```

Using Windows

With WinSCP, enter the relevant IP address and username root, then browse to the private key file created previously using PuTTYgen.



WinSCP Login

Session

Host name: 12.34.56.78

Port number: 22

User name: root

Password:

Private key file: C:\LBkeypair1.ppk

Protocol

File protocol: SFTP

☑ Allow SCP fallback

□ Advanced options

Click **Login**

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.

NOTE : 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.

Deployment Notes

1) 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 32 and 39 respectively for details on setting this up.

2) Routing Table Target Configuration

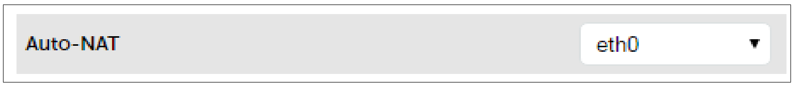
To be able to set the load balancer's ENI as a route target, make sure that the ***Source/Destination Check*** is disabled for the load balancer instance. If this is not disabled, the load balancer's ENI will not be displayed as an option in the target list.

3) 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:

- Using the WUI, go to *Cluster Configuration > Layer 4 – Advanced Configuration > Auto-NAT*
- Set *Auto-NAT* to **eth0**, i.e. the load balancer's ENI



Auto-NAT eth0 ▼

1) Load balancing Web Servers – 1 Subnet, 1 Interface, Layer 7

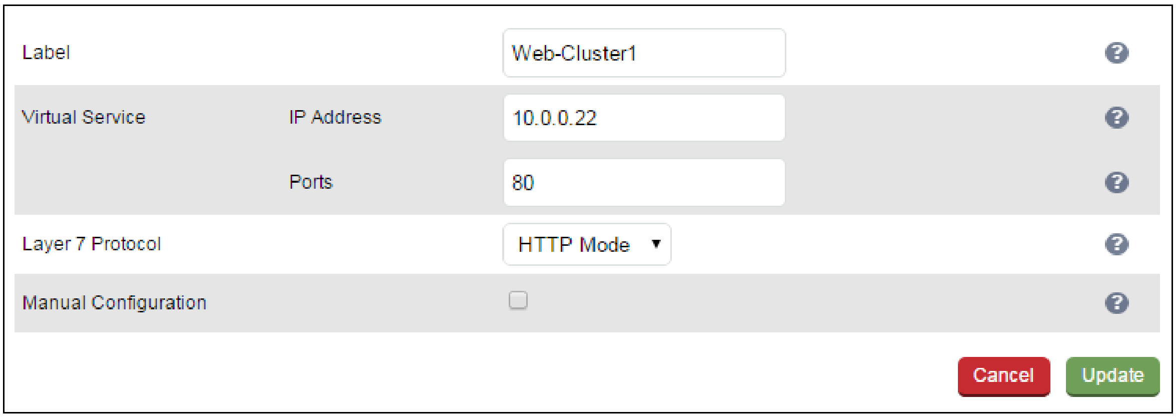
This is a simple layer 7 example using one subnet for both the load balancer and the web servers. The load balancer has a single network interface.

a) Setting up AWS

- Deploy the load balancer instance as described on page 10-13
- Deploy your required web server instances into the same VPC & subnet as the load balancer

b) Setting up the Virtual Service

- Using the WUI, go to *Cluster Configuration > Layer 7 – Virtual Service* and click **[Add a New Virtual Service]**
- Enter the following details:

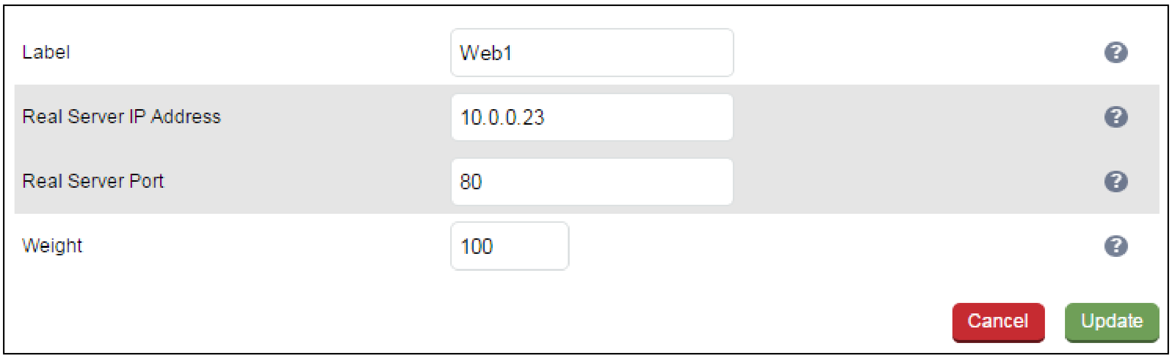


Label		Web-Cluster1	?
Virtual Service	IP Address	10.0.0.22	?
	Ports	80	?
Layer 7 Protocol		HTTP Mode ▾	?
Manual Configuration		□	?

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

c) Setting up the Real Servers

- Using the WUI, go to *Cluster Configuration > Layer 7 – Real Servers* and click **[Add a new Real Server]** next to the newly created VIP
- Enter the following details:



Label	Web1
Real Server IP Address	10.0.0.23
Real Server Port	80
Weight	100

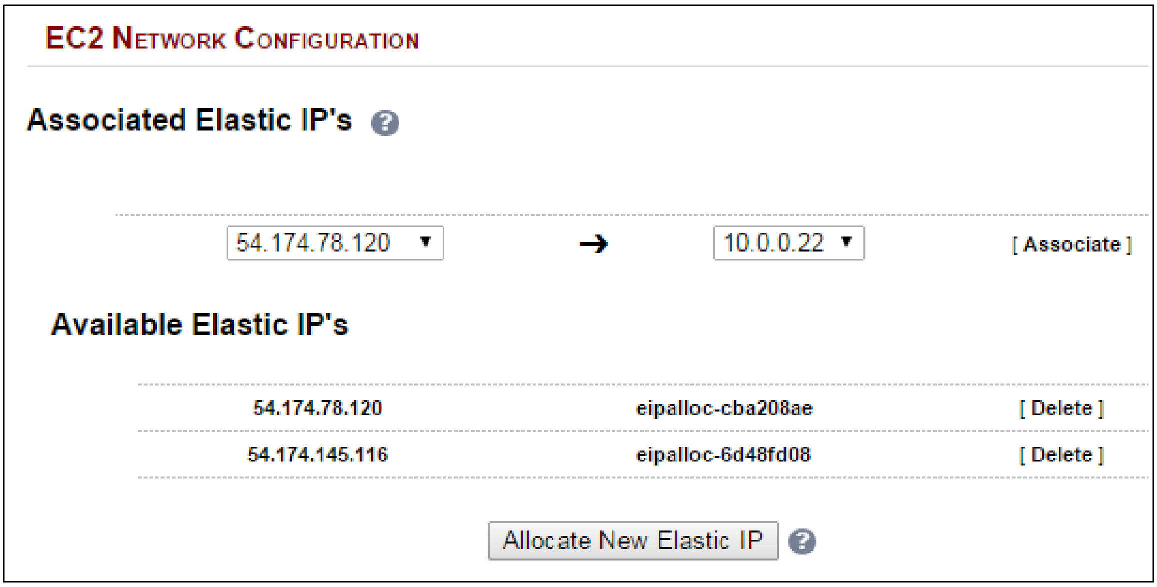
- Enter an appropriate label for the RIP, e.g. **Web1**
- Set the *Real Server IP Address* field to the required IP address, e.g. **10.0.0.23**
- Set the *Real Server Port* field to the required port, e.g. **80**
- Click **Update**
- Repeat the above steps to add your other web server(s)

d) Applying the new Layer 7 Settings

- 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)

- Using the WUI, go to *EC2 Configuration > EC2 Network Configuration*



EC2 Network Configuration

Associated Elastic IP's

54.174.78.120 → 10.0.0.22 [Associate]

Available Elastic IP's

54.174.78.120	eipalloc-cba208ae	[Delete]
54.174.145.116	eipalloc-6d48fd08	[Delete]

- 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

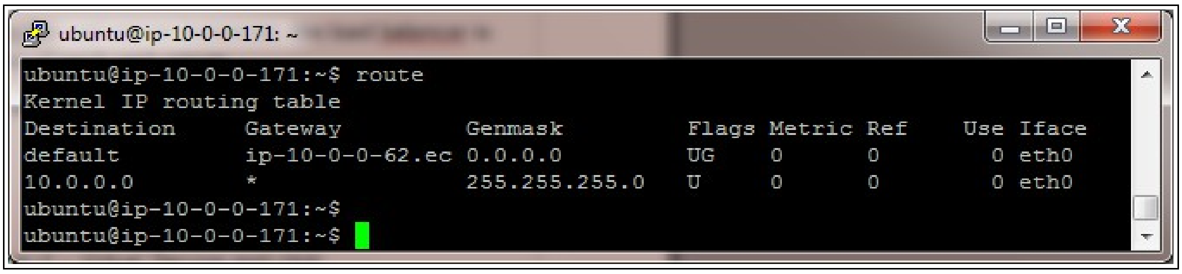
2) Load balancing Web Servers - 1 Subnet, 1 Interface, Layer 4

This is a simple layer 4 example using one subnet for both the load balancer and the web servers. 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

- Deploy the load balancer instance as described on page 10-13
- Deploy your required web server instances into the same VPC & subnet as the load balancer
- 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

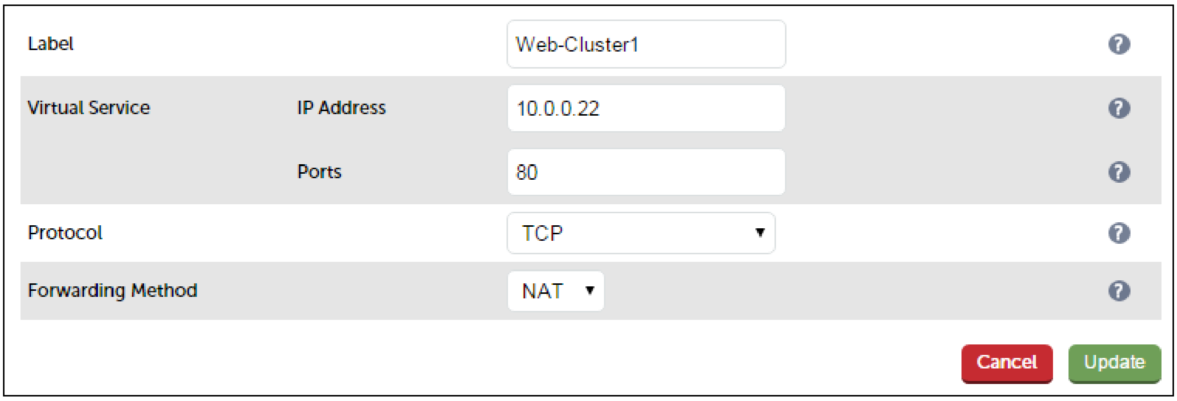
  $ sudo ip route replace default via 10.0.0.62
- The screen shot below shows that the default route is now set as the load balancer



```
ubuntu@ip-10-0-0-171:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         ip-10-0-0-62.ec 0.0.0.0         UG    0      0        0 eth0
10.0.0.0        *               255.255.255.0   U     0      0        0 eth0
ubuntu@ip-10-0-0-171:~$
ubuntu@ip-10-0-0-171:~$
```

b) Setting up the Virtual Service

- Using the WUI, go to *Cluster Configuration > Layer 4 – Virtual Service* and click **[Add a New Virtual Service]**
- Enter the following details:



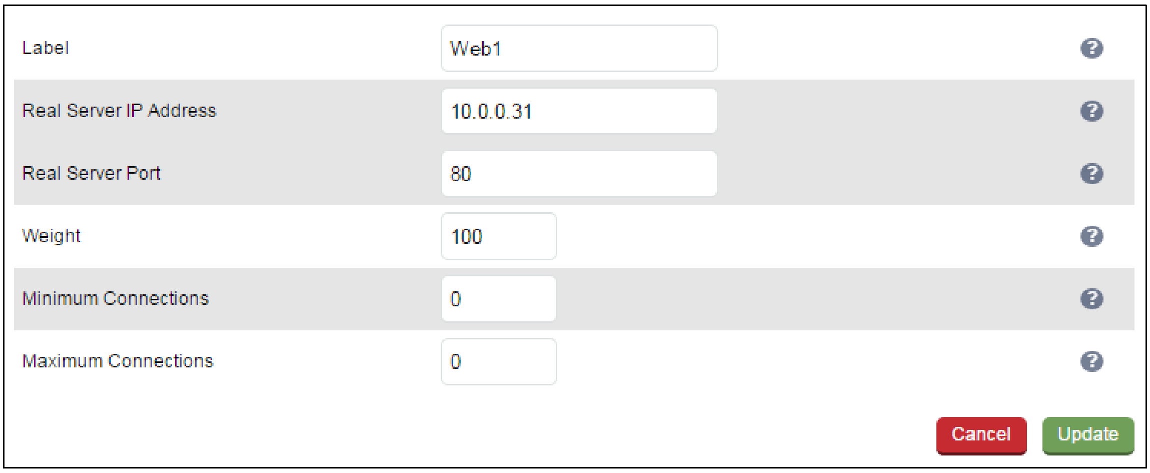
Label		Web-Cluster1	?
Virtual Service	IP Address	10.0.0.22	?
	Ports	80	?
Protocol		TCP ▼	?
Forwarding Method		NAT ▼	?

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

c) Setting up the Real Servers

- Using the WUI, go to *Cluster Configuration > Layer 4 – Real Servers* and click **[Add a new Real Server]** next to the newly created VIP

- Enter the following details:

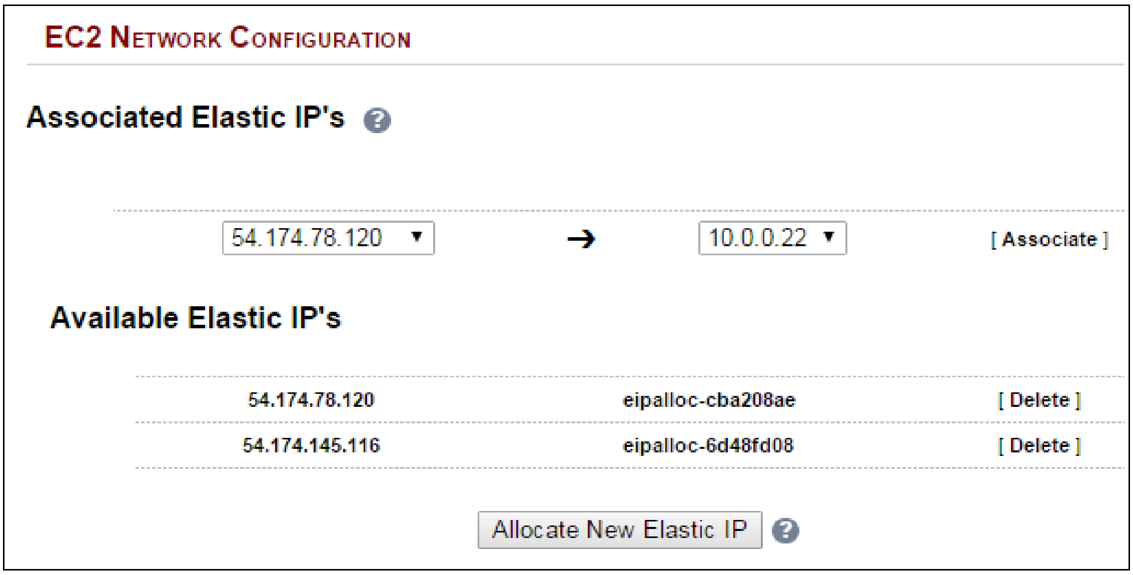


Label	Web1
Real Server IP Address	10.0.0.31
Real Server Port	80
Weight	100
Minimum Connections	0
Maximum Connections	0

- Enter an appropriate label for the RIP, e.g. **Web1**
- Change the *Real Server IP Address* field to the required IP address, e.g. **10.0.0.31**
- Set *Real Server Port* to **80**
- Click **Update**
- 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)

- Using the WUI, go to *EC2 Configuration > EC2 Network Configuration*



EC2 Network Configuration

Associated Elastic IP's

54.174.78.120 ▼ → 10.0.0.22 ▼ [Associate]

Available Elastic IP's

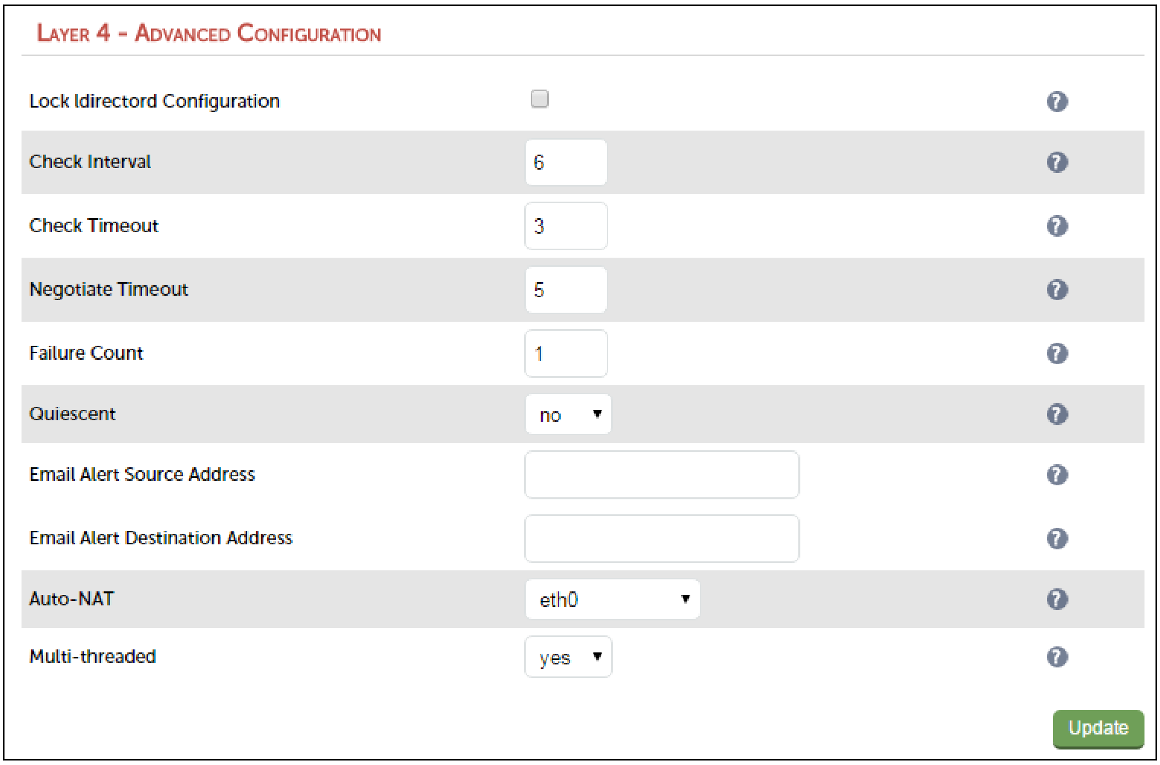
54.174.78.120	eipalloc-cba208ae	[Delete]
54.174.145.116	eipalloc-6d48fd08	[Delete]

- 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

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.

- Using the WUI, go to *Cluster Configuration > Layer 4 – Advanced Configuration*



LAYER 4 - ADVANCED CONFIGURATION

Lock ldirectord Configuration	□
Check Interval	6
Check Timeout	3
Negotiate Timeout	5
Failure Count	1
Quiescent	no
Email Alert Source Address	
Email Alert Destination Address	
Auto-NAT	eth0
Multi-threaded	yes

- Change the *Auto-NAT* setting to **eth0**
- Click **Update**

3) Load balancing Web Servers – 2 Subnets, 2 Interfaces, Layer 7

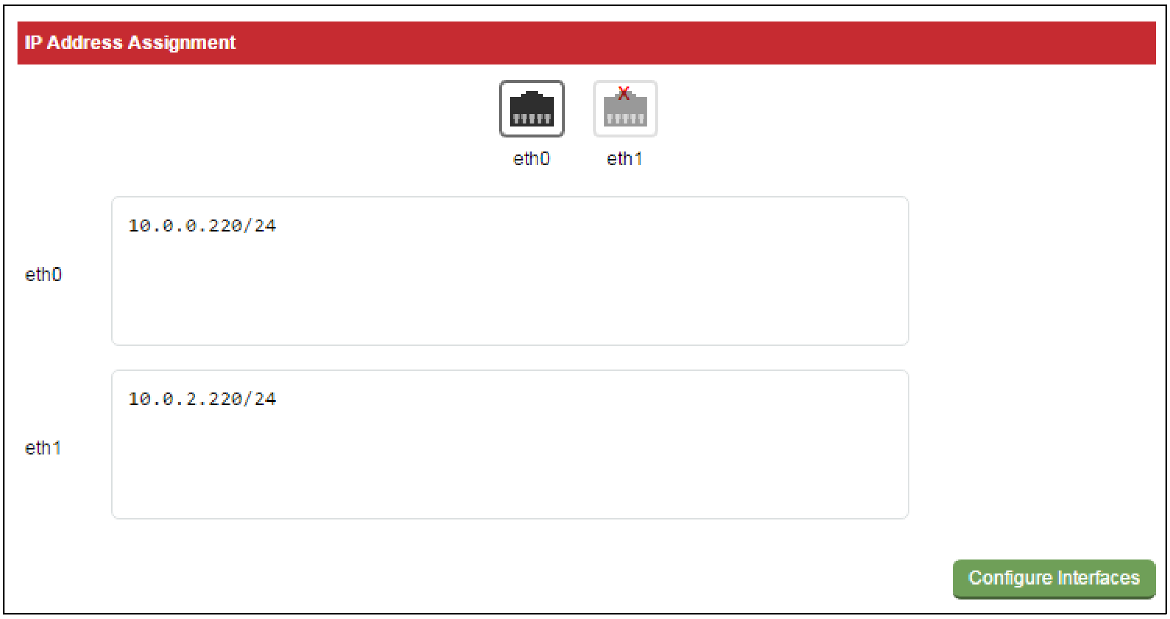
This example uses 2 subnets – the load balancer is configured with 2 interfaces - 1 interface in subnet 1 and the other in subnet 2. The real severs are connected to subnet 2.

a) Setting up AWS

- Deploy the load balancer instance as described on page 10-13
- Add a second subnet to your VPC, skip this step if you already have one
- Add a second Network Interface, associate it with the second subnet and attach it to the load balancer instance
- Deploy your required web server instances into the second subnet

b) Configuring the second Network Interface

- Using the WUI option: *Local Configuration > Network Interface Configuration* , assign an IP address for the second interface (eth1) , e.g. **10.0.2.220/24**



IP Address Assignment

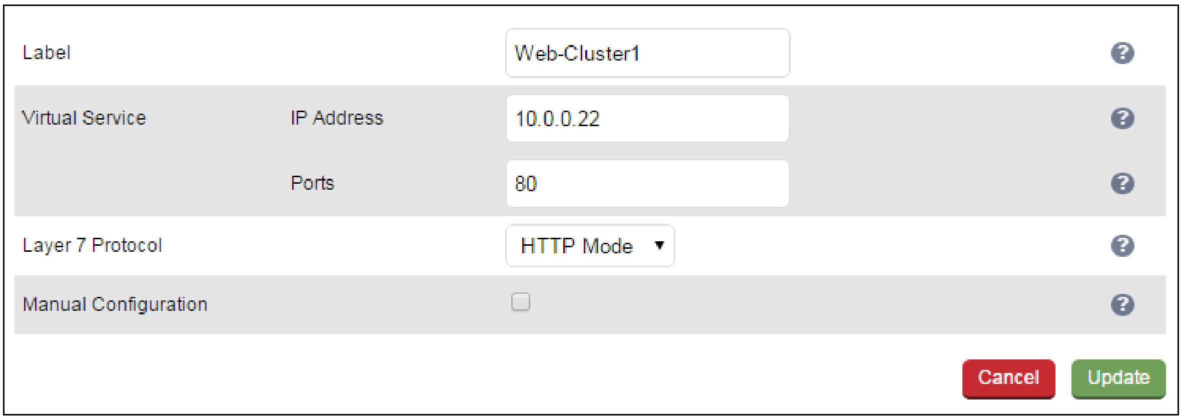
eth0: 10.0.0.220/24

eth1: 10.0.2.220/24

- Click **Configure Interfaces**

c) Setting up the Virtual Service

- Using the WUI, go to *Cluster Configuration > Layer 7 – Virtual Service* and click **[Add a New Virtual Service]**
- Enter the following details:



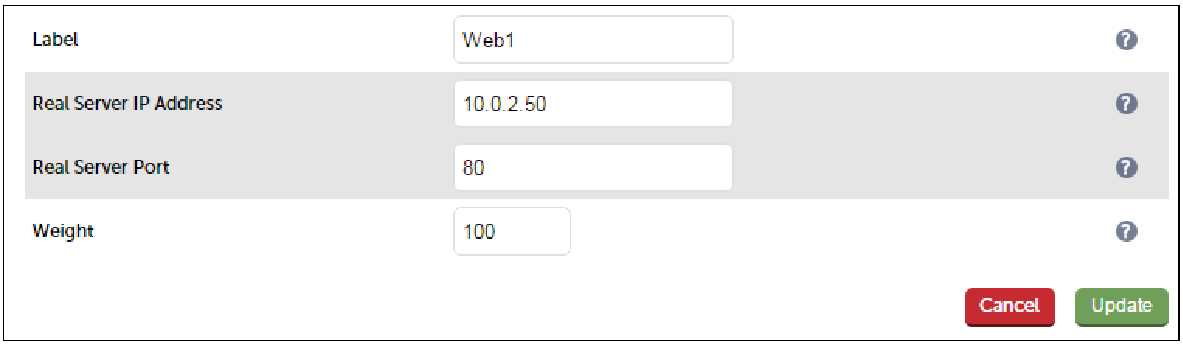
Label		Web-Cluster1
Virtual Service	IP Address	10.0.0.22
	Ports	80
Layer 7 Protocol		HTTP Mode ▾
Manual Configuration		□

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

d) Setting up the Real Servers

- Using the WUI, go to *Cluster Configuration > Layer 7 – Real Servers* and click **[Add a new Real Server]** next to the newly created VIP

- Enter the following details:



Label	Web1
Real Server IP Address	10.0.2.50
Real Server Port	80
Weight	100

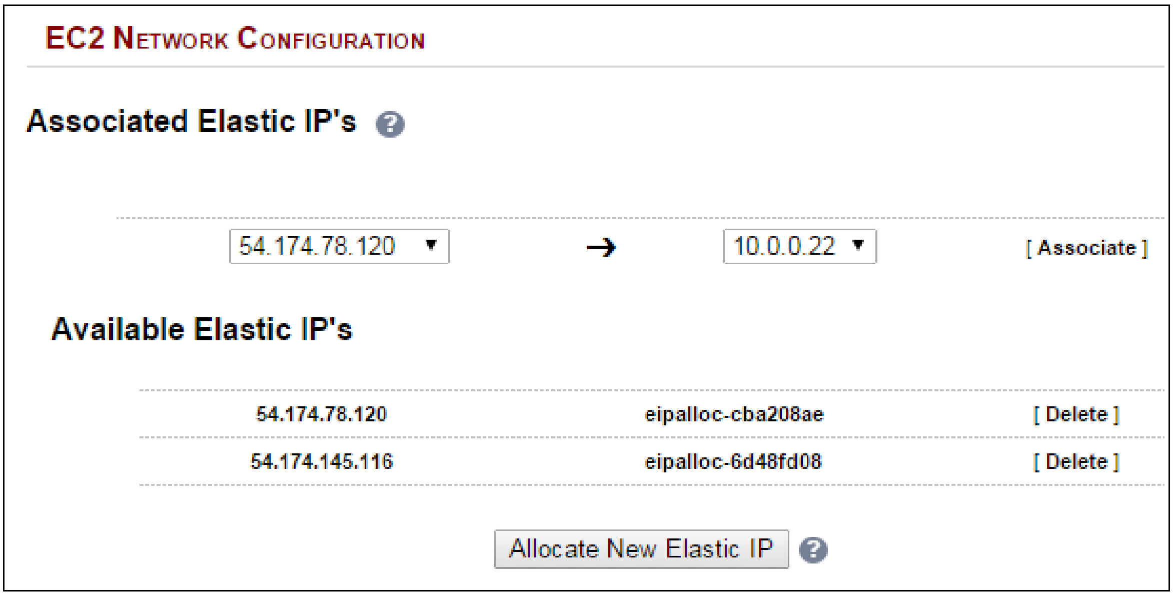
- Enter an appropriate label for the RIP, e.g. **Web1**
- Change the *Real Server IP Address* field to the required IP address, e.g. **10.0.2.50**
- Set the *Real Server Port* field to the required port, e.g. **80**
- Click **Update**
- Repeat the above steps to add your other web server(s)

e) Applying the new Layer 7 Settings

- 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)

- Using the WUI, go to *EC2 Configuration > EC2 Network Configuration*



EC2 Network Configuration

Associated Elastic IP's

54.174.78.120 → 10.0.0.22 [Associate]

Available Elastic IP's

54.174.78.120	eipalloc-cba208ae	[Delete]
54.174.145.116	eipalloc-6d48fd08	[Delete]

- 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

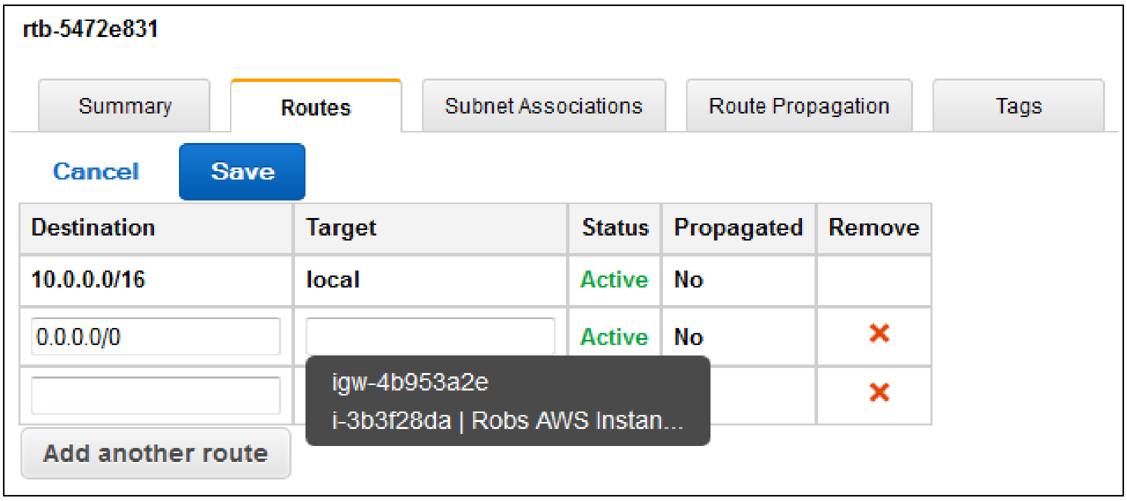
NOTE : Dual interface layer 7 SNAT mode with TProxy enabled (for transparency) where each interface of the load balancer is connected to a different subnet and the default gateway of the real servers is configured to be the load balancer is **not** supported. Please refer to example 4 on the following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

4) Load balancing Web Servers - 2 Subnets, 1 Interface, Layer 7 with Transparency

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

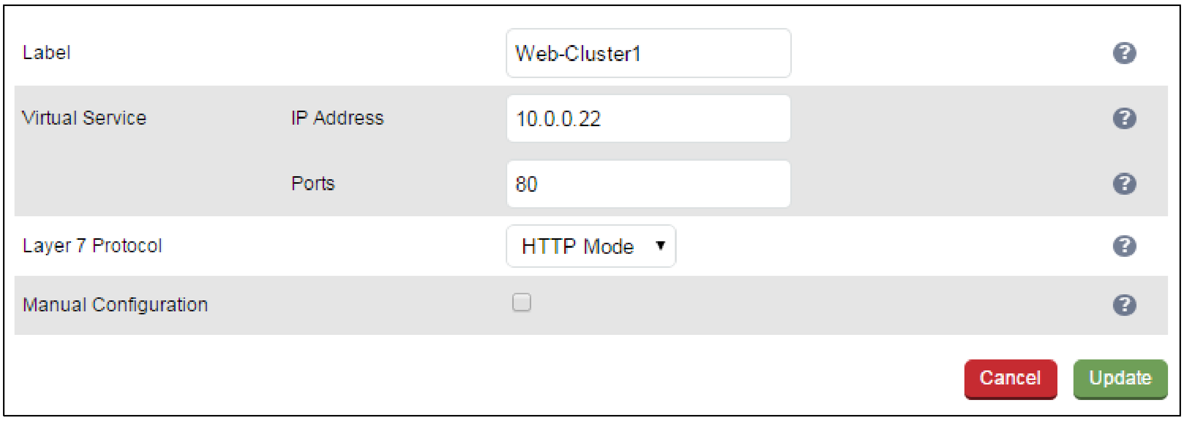
- Deploy the load balancer instance as described on page 10-13
- Add a second subnet to your VPC, skip this step if you already have one
- Deploy your required web server instances into the second subnet
- Add a default route to the second subnets routing table (the subnet where the web servers are located), 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**
    - 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



Destination	Target	Status	Propagated	Remove
10.0.0.0/16	local	Active	No	
0.0.0.0/0		Active	No	✕
				✕

b) Setting up the Virtual Service

- Using the WUI, go to *Cluster Configuration > Layer 7 – Virtual Service* and click **[Add a New Virtual Service]**
- Enter the following details:



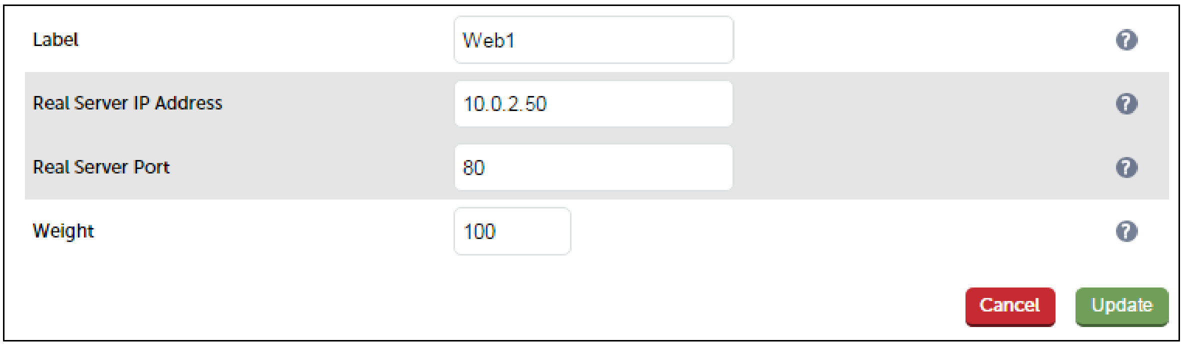
Label		Web-Cluster1	?
Virtual Service	IP Address	10.0.0.22	?
	Ports	80	?
Layer 7 Protocol		HTTP Mode ▾	?
Manual Configuration		□	?

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

c) Setting up the Real Servers

- Using the WUI, go to *Cluster Configuration > Layer 7 – Real Servers* and click **[Add a new Real Server]** next to the newly created VIP

- Enter the following details:



Label	Web1	?
Real Server IP Address	10.0.2.50	?
Real Server Port	80	?
Weight	100	?

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

d) Configuring Layer 7 – Advanced Settings

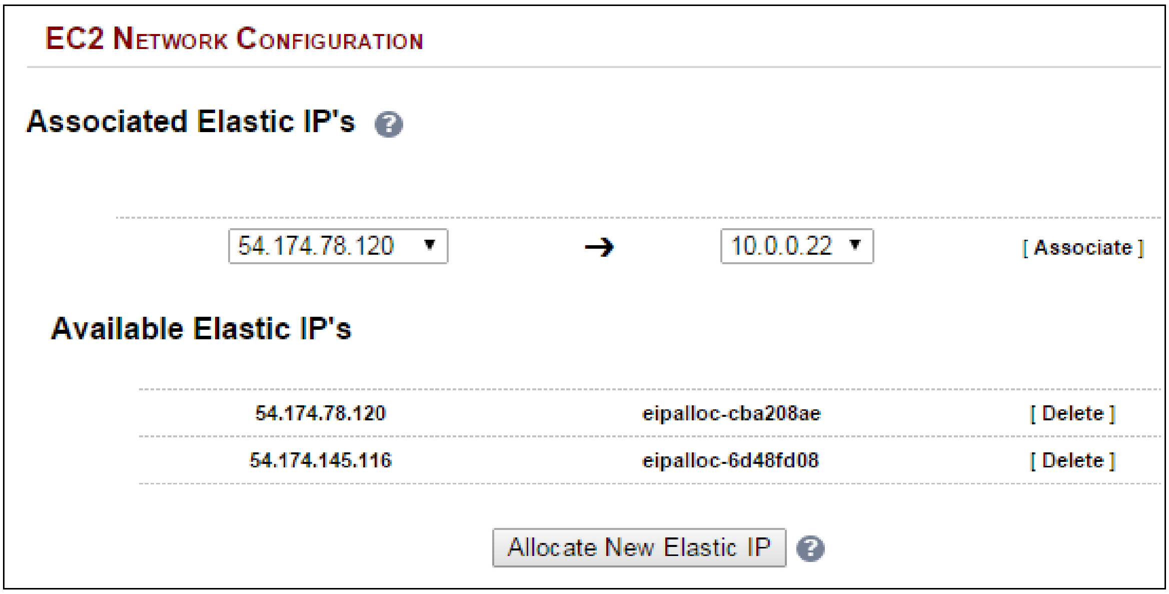
- Using the WUI, go to *Cluster Configuration > Layer 7 – Advanced Configuration*
- Enable (check) *Transparent Proxy*
- Click **Update**

e) Applying the new Layer 7 Settings

- 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)

- Using the WUI, go to *EC2 Configuration > EC2 Network Configuration*



EC2 Network Configuration

Associated Elastic IP's

54.174.78.120 ▼ → 10.0.0.22 ▼ [Associate]

Available Elastic IP's

54.174.78.120	eipalloc-cba208ae	[Delete]
54.174.145.116	eipalloc-6d48fd08	[Delete]

- 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

5) Load balancing Web Servers - 1 Subnet, 1 Interface, Layer 7 with SSL Termination

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

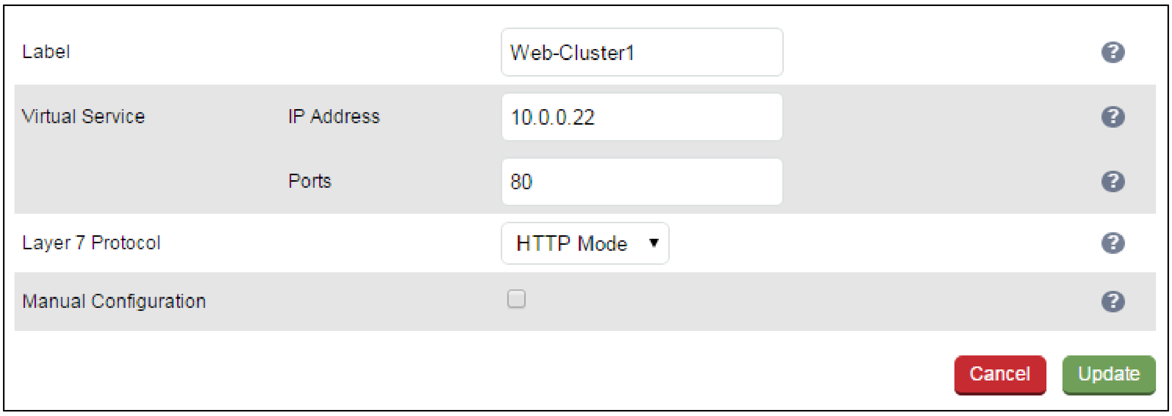
We generally recommend that SSL should be terminated on the backend servers rather than the load balancer for scalability reasons.

a) Setting up AWS

- Deploy the load balancer instance as described on page 10-13
- Deploy your required web server instances into the same VPC & subnet as the load balancer

b) Setting up the Virtual Service

- Using the WUI, go to *Cluster Configuration > Layer 7 – Virtual Service* and click **[Add a New Virtual Service]**
- Enter the following details:

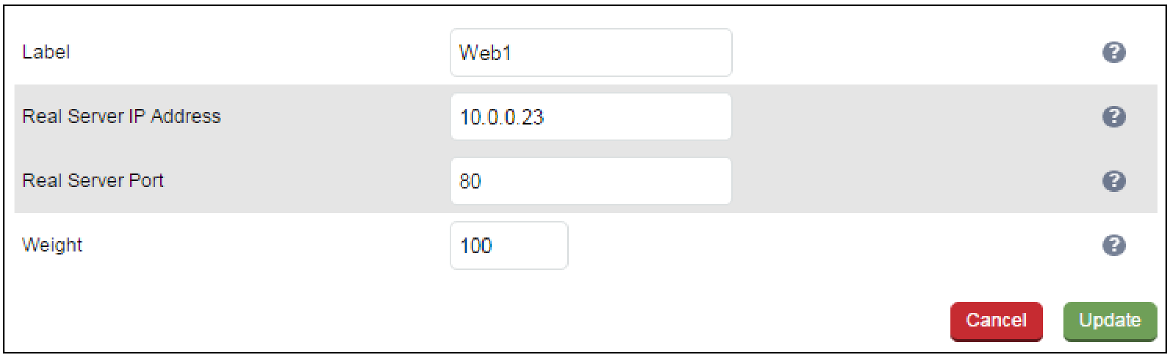


Label		Web-Cluster1
Virtual Service	IP Address	10.0.0.22
	Ports	80
Layer 7 Protocol		HTTP Mode ▾
Manual Configuration		□

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

c) Setting up the Real Servers

- Using the WUI, go to *Cluster Configuration > Layer 7 – Real Servers* and click **[Add a new Real Server]** next to the newly created VIP
- Enter the following details:



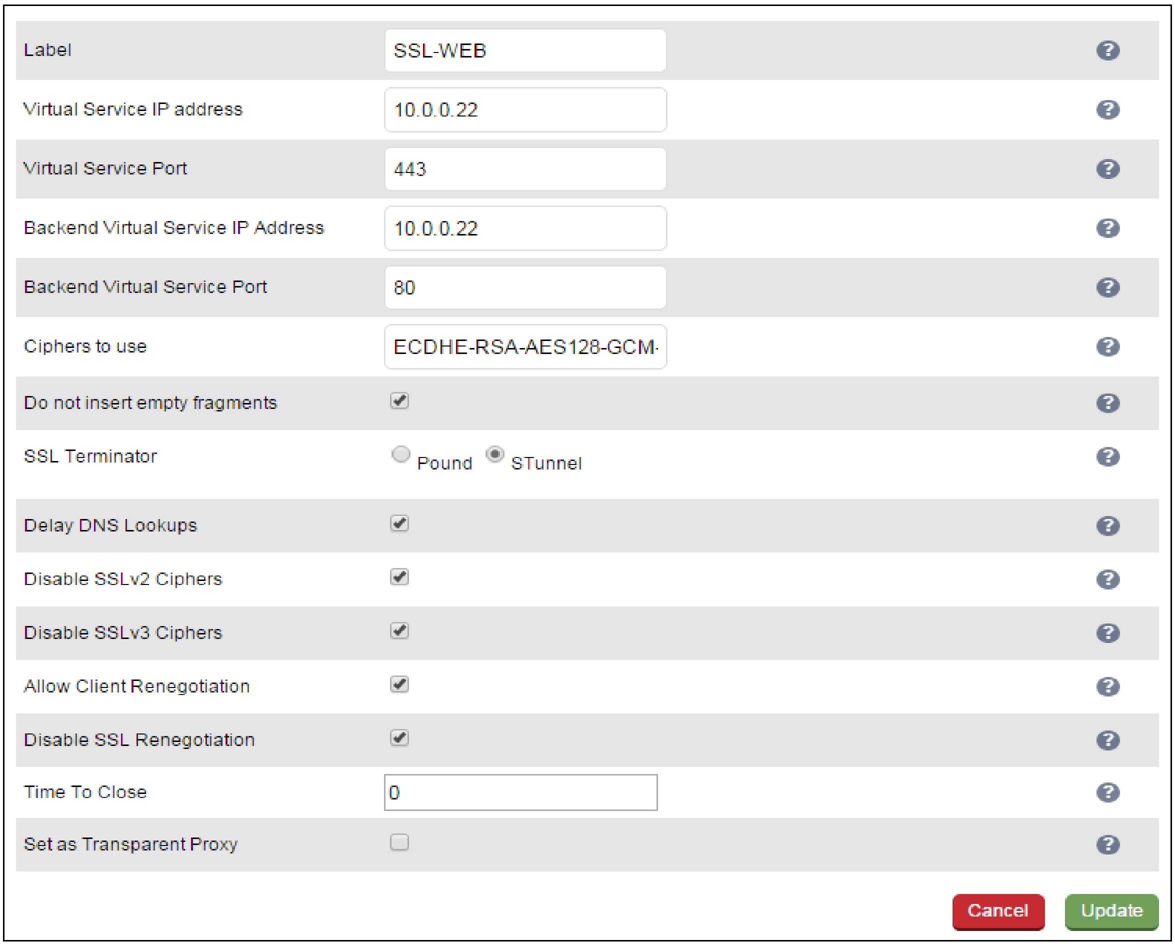
Label	Web1
Real Server IP Address	10.0.0.23
Real Server Port	80
Weight	100

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

d) Configuring SSL Termination

- Using the WUI, go to *Cluster Configuration > SSL Termination* and click **[Add a New Virtual Service]**

- Enter the following details:



Label	SSL-WEB
Virtual Service IP address	10.0.0.22
Virtual Service Port	443
Backend Virtual Service IP Address	10.0.0.22
Backend Virtual Service Port	80
Ciphers to use	ECDHE-RSA-AES128-GCM-
Do not insert empty fragments	☑
SSL Terminator	○ Pound ● STunnel
Delay DNS Lookups	☑
Disable SSLv2 Ciphers	☑
Disable SSLv3 Ciphers	☑
Allow Client Renegotiation	☑
Disable SSL Renegotiation	☑
Time To Close	0
Set as Transparent Proxy	☐

- Enter an appropriate label for the VIP, e.g. **SSL-WEB**
- Set the *Virtual Service IP address* to be the same as the VIP created in step (c) e.g. **10.0.0.22**
- Set the *Virtual Service Ports* field to **443**
- Set the *Backend Virtual Service IP address* to be the same as the VIP created in step (c) e.g. **10.0.0.22**
- Set the *Backend Virtual Service Ports* field to **80**
- Leave all other settings at their default values
- Click **Update**

SSL Certificate Notes:

- A default self-signed certificate is used when SSL virtual services are first defined
- To change this, using the WUI, select: *Cluster Configuration > SSL Termination*
- Click **[Certificate]** next to the Virtual Service
- If you already have a certificate, use the **Upload prepared PEM/PFX file** option at the bottom of the screen to upload it
- If you don't have a certificate, you can create a CSR using the **Generate SSL Certificate Request** section. This will create the CSR in the upper pane of the **Upload Signed Certificate** section based on the settings you enter. This should be copied and sent to your CA
- Once the signed certificate is received, copy/paste it (along with any required intermediate certificates) into the lower pane of the **Upload Signed Certificate** section, and click the **Upload Signed Certificate** button

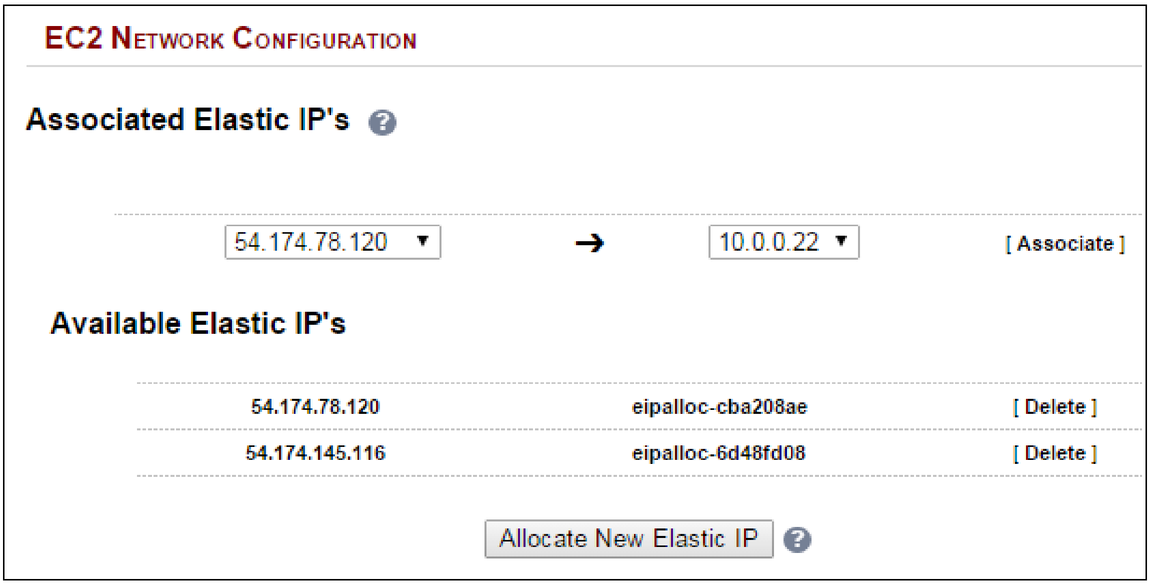
e) Applying the new Settings

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

- 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)

- Using the WUI, go to *EC2 Configuration > EC2 Network Configuration*



EC2 Network Configuration

Associated Elastic IP's

54.174.78.120 ▾ → 10.0.0.22 ▾ [Associate]

Available Elastic IP's

54.174.78.120	eipalloc-cba208ae	[Delete]
54.174.145.116	eipalloc-6d48fd08	[Delete]

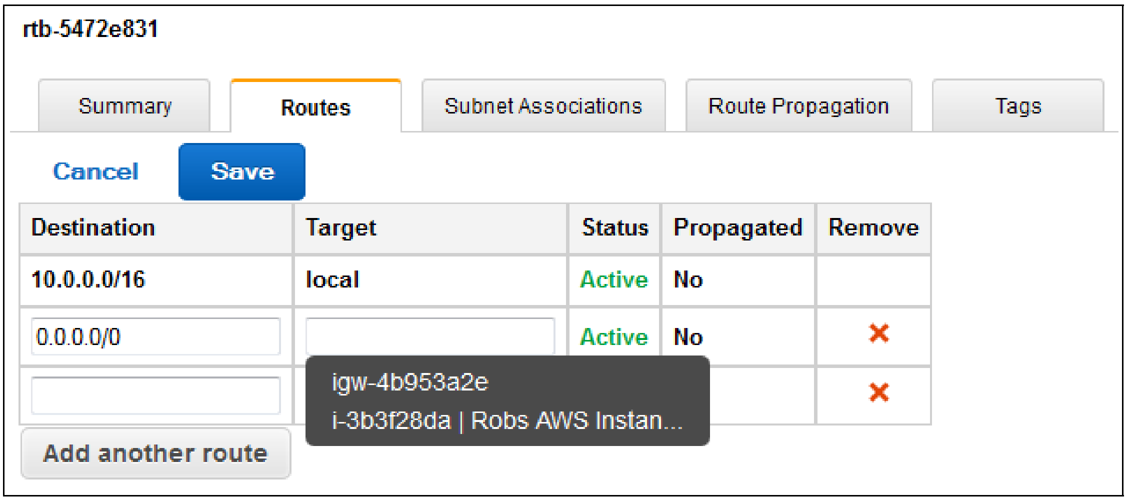
- Under the Associated Elastic IP's section click **[Associate]** next to the VIPs private IP address (10.0.0.22 in this case), if no Elastic IP's are available, use the **Allocate New Elastic IP** button to add one

6) Load balancing RD Session Hosts - 2 Subnets, 1 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. Routing rules for the second subnet where the session hosts are located must also be changed.

a) Setting up AWS

- Deploy the load balancer instance as described on page 10-13
- Add a second subnet to your VPC, skip this step if you already have one
- Deploy your required session host server instances into the second subnet
- Add a default route to the second subnets routing table, 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**
  - 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



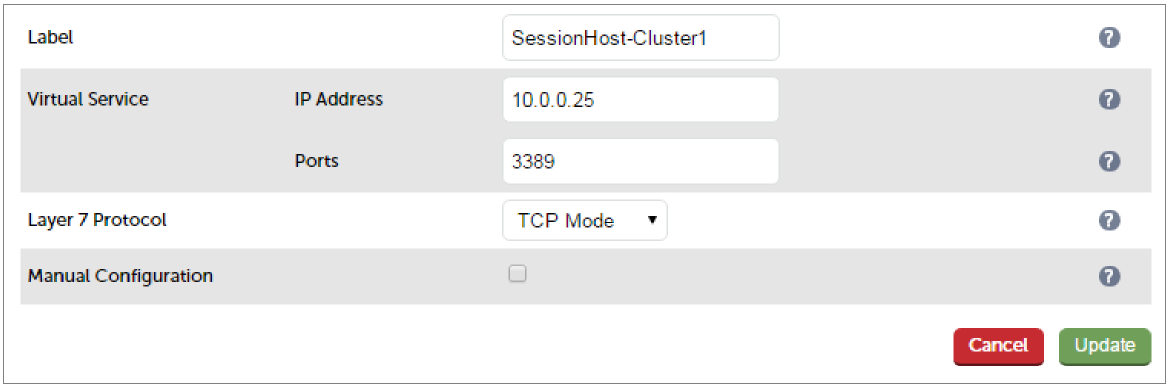
rtb-5472e831

Destination	Target	Status	Propagated	Remove
10.0.0.0/16	local	Active	No	
0.0.0.0/0		Active	No	✕
				✕

igw-4b953a2e
i-3b3f28da | Robs AWS Instan...

b) Setting up the Virtual Service

- Using the WUI, go to *Cluster Configuration > Layer 7 – Virtual Service* and click **[Add a New Virtual Service]**
- Enter the following details:



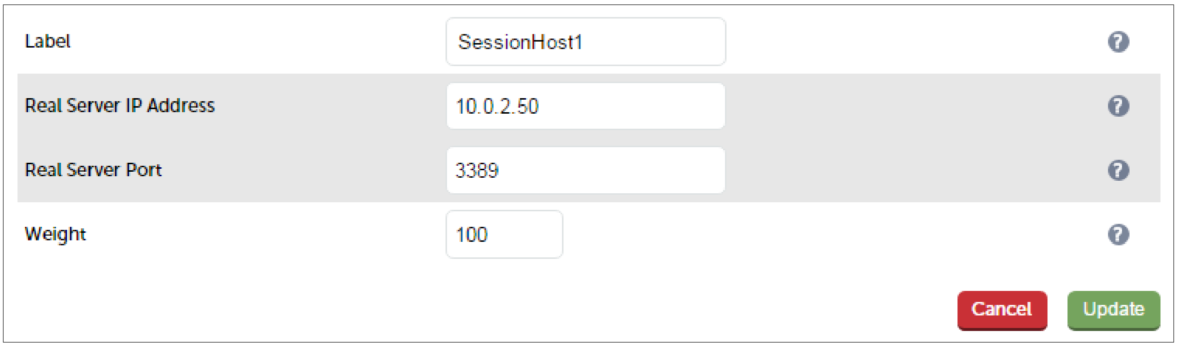
Label		SessionHost-Cluster1
Virtual Service	IP Address	10.0.0.25
	Ports	3389
Layer 7 Protocol		TCP Mode
Manual Configuration		□

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

c) Setting up the Real Servers

- Using the WUI, go to *Cluster Configuration* > *Layer 7 – Real Servers* and click **[Add a new Real Server]** next to the newly created VIP

- Enter the following details:



Label	SessionHost1
Real Server IP Address	10.0.2.50
Real Server Port	3389
Weight	100

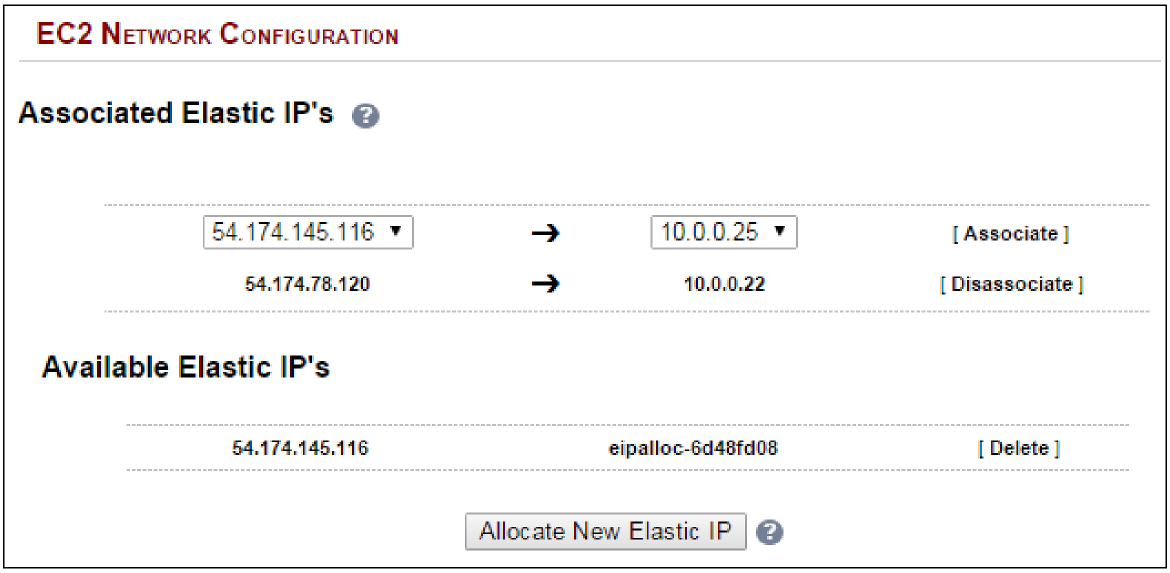
- Enter an appropriate label for the RIP, e.g. **SessionHost1**
- Change the *Real Server IP Address* field to the required IP address, e.g. **10.0.2.50**
- Set the *Real Server Port* field to the required port, e.g. **3389**
- Click **Update**
- Repeat the above steps to add your other session host server(s)

d) Applying the new Layer 7 Settings

- 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)

- Using the WUI, go to *EC2 Configuration > EC2 Network Configuration*



EC2 NETWORK CONFIGURATION

Associated Elastic IP's

54.174.145.116 ▼	→	10.0.0.25 ▼	[Associate]
54.174.78.120	→	10.0.0.22	[Disassociate]

Available Elastic IP's

54.174.145.116	eipalloc-6d48fd08	[Delete]

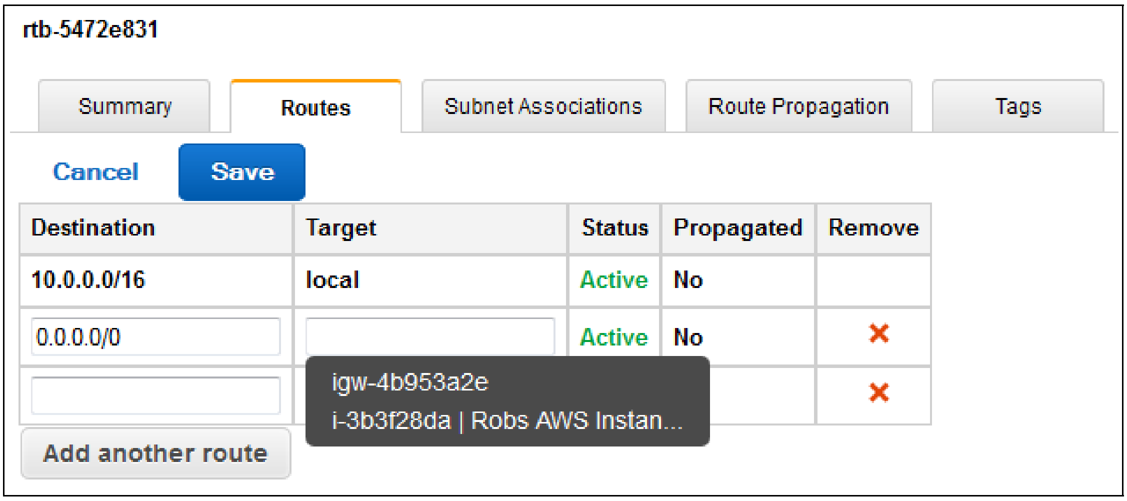
- Under the Associated Elastic IP's section click **[Associate]** next to the VIPs private IP address (10.0.0.25 in this case), if no Elastic IP's are available, use the **Allocate New Elastic IP** button to add one

7) Load balancing Web Servers - 2 Subnets, 1 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

- Deploy the load balancer instance as described on page 10-13
- Add a second subnet to your VPC, skip this step if you already have one
- Deploy your required web server instances into the second subnet
- 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

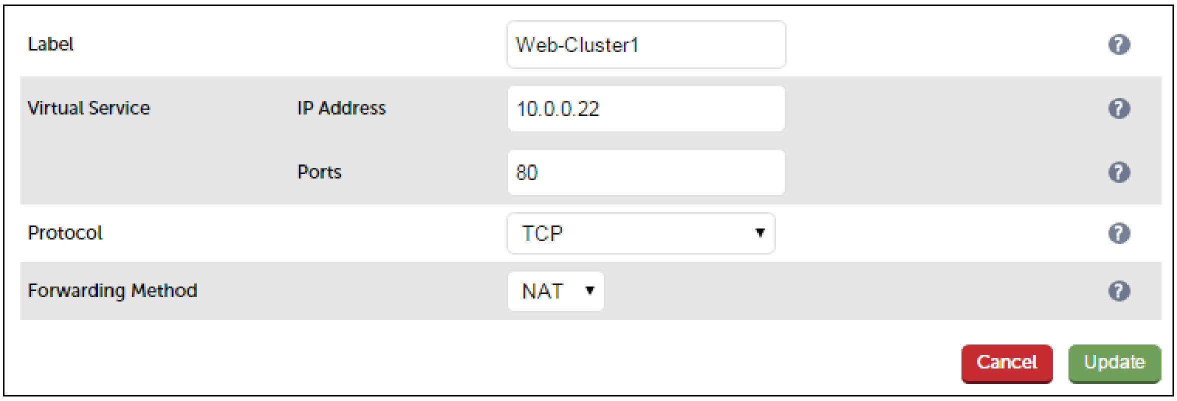


Destination	Target	Status	Propagated	Remove
10.0.0.0/16	local	Active	No	
0.0.0.0/0		Active	No	✕
				✕

igw-4b953a2e
i-3b3f28da | Robs AWS Instan...

b) Setting up the Virtual Service

- Using the WUI, go to *Cluster Configuration > Layer 4 – Virtual Service* and click **[Add a New Virtual Service]**
- Enter the following details:



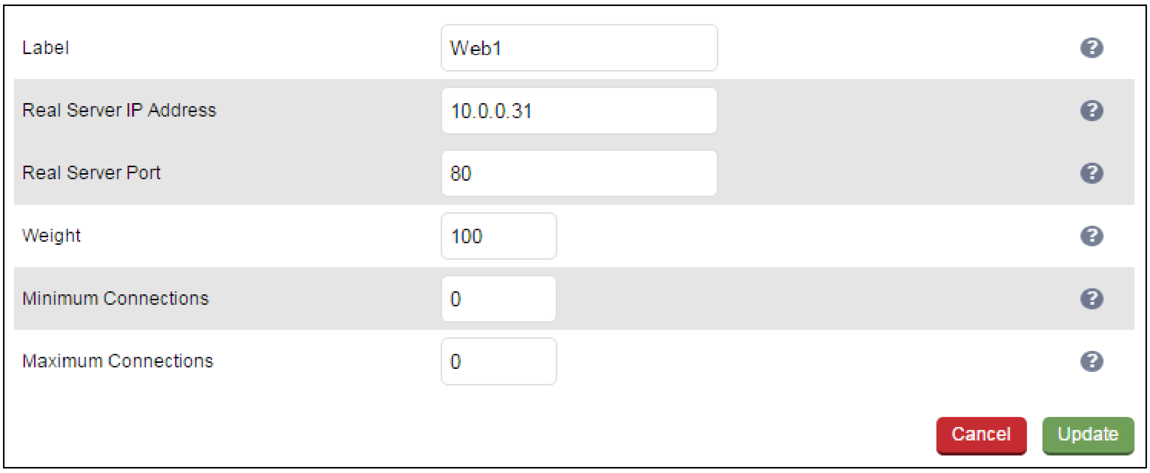
Label		Web-Cluster1
Virtual Service	IP Address	10.0.0.22
	Ports	80
Protocol		TCP
Forwarding Method		NAT

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

c) Setting up the Real Servers

- Using the WUI, go to *Cluster Configuration > Layer 4 – Real Servers* and click **[Add a new Real Server]** next to the newly created VIP

- Enter the following details:

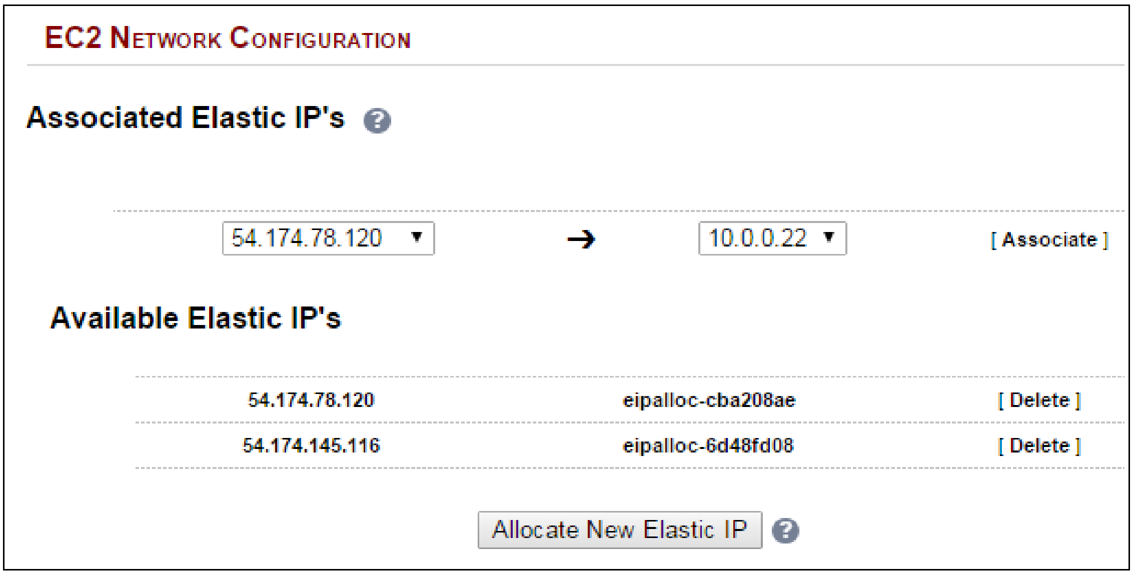


Label	Web1
Real Server IP Address	10.0.0.31
Real Server Port	80
Weight	100
Minimum Connections	0
Maximum Connections	0

- Enter an appropriate label for the RIP, e.g. **Web1**
- Change the *Real Server IP Address* field to the required IP address, e.g. **10.0.0.31**
- Set *Real Server Port* to **80**
- Click **Update**
- 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)

- Using the WUI, go to *EC2 Configuration > EC2 Network Configuration*



EC2 Network Configuration

Associated Elastic IP's

54.174.78.120 ▾	→	10.0.0.22 ▾	[Associate]

Available Elastic IP's

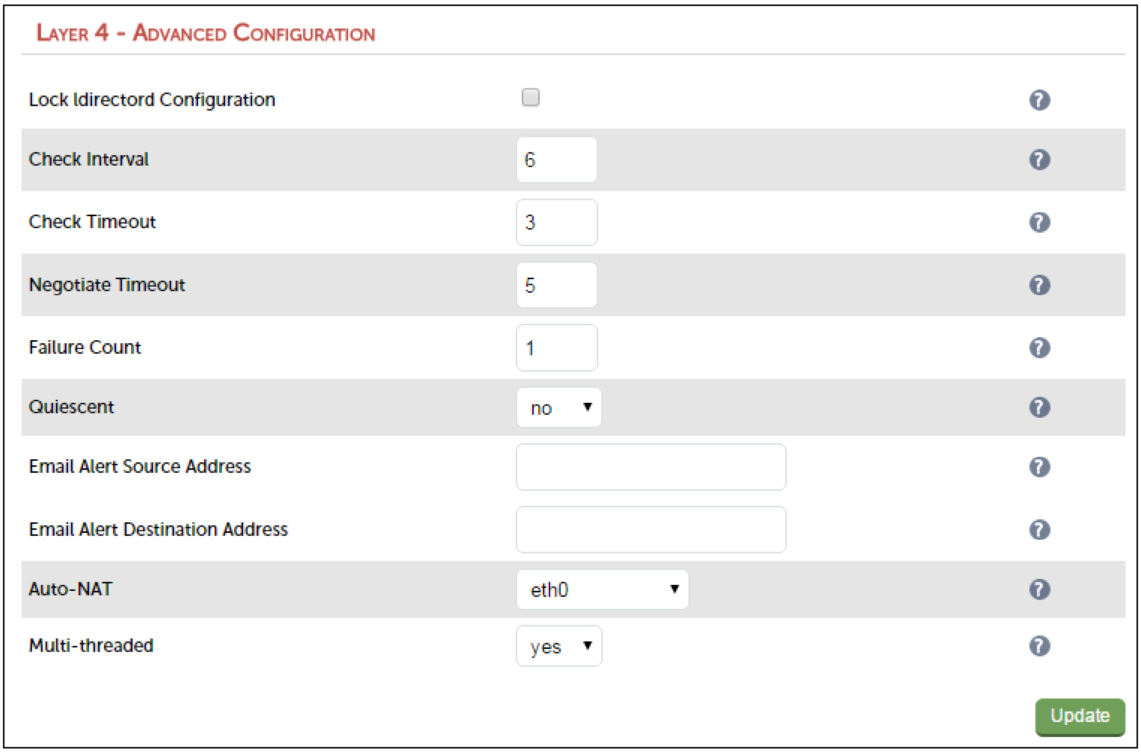
54.174.78.120	eipalloc-cba208ae	[Delete]
54.174.145.116	eipalloc-6d48fd08	[Delete]

- 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

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.

- Using the WUI, go to *Cluster Configuration > Layer 4 – Advanced Configuration*



Layer 4 - Advanced Configuration

Lock Idirectord Configuration	☐
Check Interval	6
Check Timeout	3
Negotiate Timeout	5
Failure Count	1
Quiescent	no ▾
Email Alert Source Address	
Email Alert Destination Address	
Auto-NAT	eth0 ▾
Multi-threaded	yes ▾

- Change the *Auto-NAT* setting to **eth0**
- **Click Update**

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.

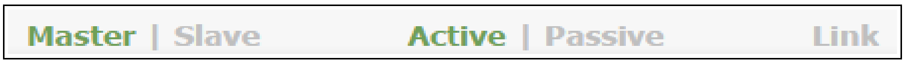
Use two or more clients to do the testing. Open up a web browser on each test clients and enter the URL for the VIP e.g. **http://192.168.110.10**

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 WUI. For a single appliance, the status bar should report **Master** & **Active** as shown below:



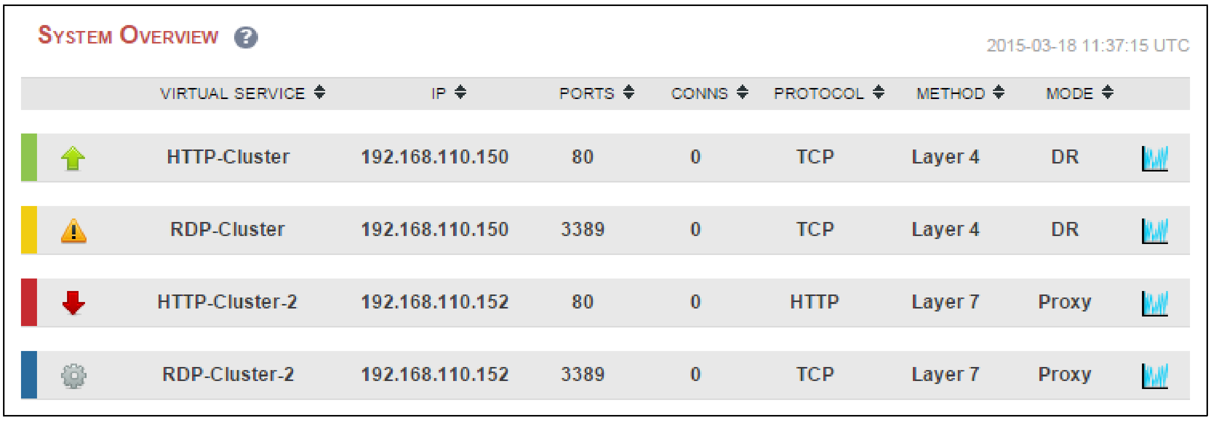
Master | Slave Active | Passive Link

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).



SYSTEM OVERVIEW ? 2015-03-18 11:37:15 UTC

		VIRTUAL SERVICE	IP	PORTS	CONNS	PROTOCOL	METHOD	MODE	
🟩	⬆	HTTP-Cluster	192.168.110.150	80	0	TCP	Layer 4	DR	📈
🟨	⚠	RDP-Cluster	192.168.110.150	3389	0	TCP	Layer 4	DR	📈
🟥	⬇	HTTP-Cluster-2	192.168.110.152	80	0	HTTP	Layer 7	Proxy	📈
🟦	⚙	RDP-Cluster-2	192.168.110.152	3389	0	TCP	Layer 7	Proxy	📈

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

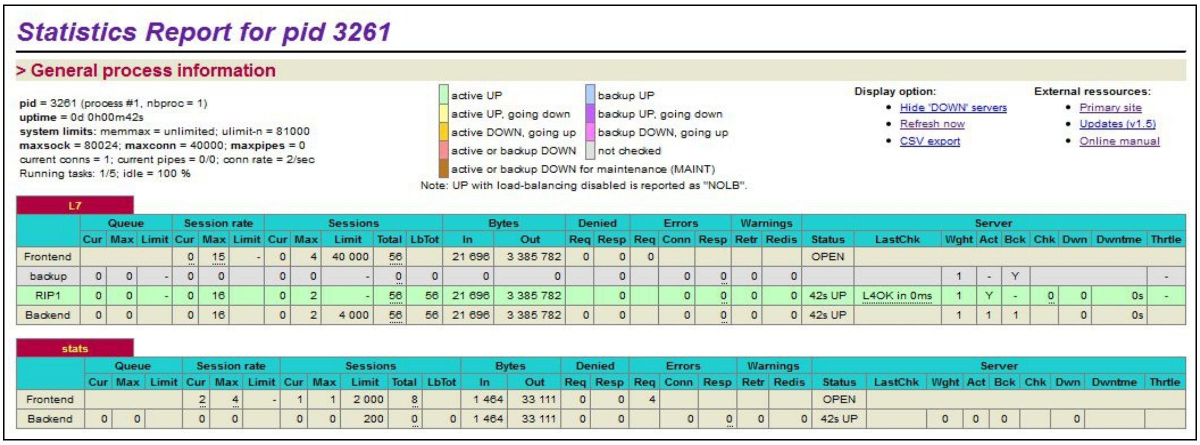
   → for single subnet Layer 4 mode make sure that the default gateway on all real servers is set to be the load balancer

   → 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:



Statistics Report for pid 3261

> General process information

pid = 3261 (process #1, nbproc = 1)
uptime = 0d 0h00m42s
system limits: memmax = unlimited; ulimit-n = 81000
maxsock = 80024; **maxconn** = 40000; **maxpipes** = 0
current conns = 1; current pipes = 0/0; conn rate = 2/sec
Running tasks: 1/5; idle = 100 %

active UP	backup UP
active UP, going down	backup UP, going down
active DOWN, going up	backup DOWN, going up
active or backup DOWN	not checked
active or backup DOWN for maintenance (MAINT)	

Note: UP with load-balancing disabled is reported as "NOLB".

Display option:
- Hide 'DOWN' servers
- Refresh now
- CSV export

External ressources:
- Primary site
- Updates (v1.5)
- Online manual

L7

	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				0	15	-	0	4	40 000	56		21 696	3 385 782	0	0	0					OPEN								
backup	0	0	-	0	0		0	0	-	0	0	0	0		0		0	0	0	0			1	-	Y				-
RIP1	0	0	-	0	16		0	2	-	56	56	21 696	3 385 782		0		0	0	0	0	42s UP	L4OK in 0ms	1	Y	-	0	0	0s	-
Backend	0	0		0	16		0	2	4 000	56	56	21 696	3 385 782	0	0		0	0	0	0	42s UP		1	1	1		0	0s	

stats

	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				2	4	-	1	1	2 000	8		1 464	33 111	0	0	4					OPEN								
Backend	0	0		0	0		0	0	200	0	0	1 464	33 111	0	0		0	0	0	0	42s UP		0	0	0		0		

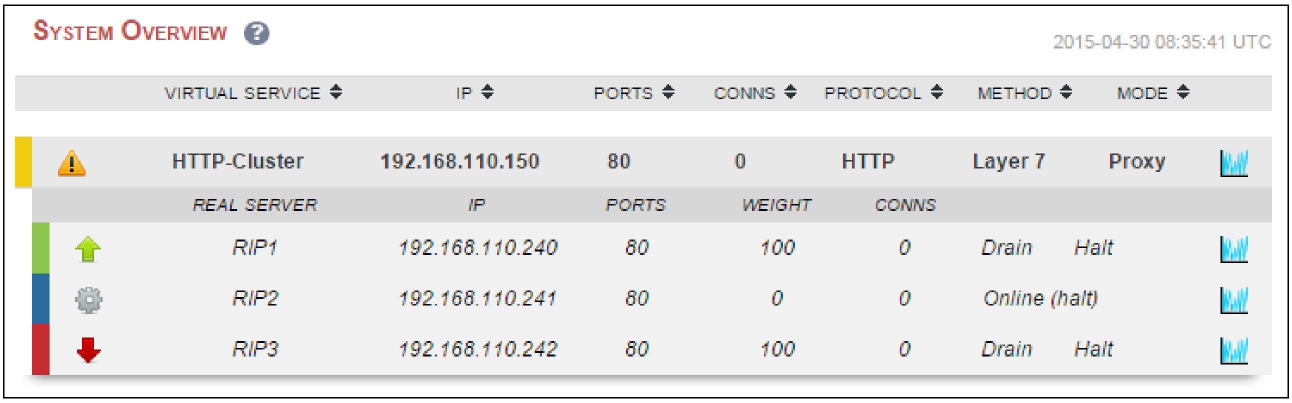
Taking Real Servers Offline

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:



SYSTEM OVERVIEW

2015-04-30 08:35:41 UTC

	VIRTUAL SERVICE	IP	PORTS	CONNS	PROTOCOL	METHOD	MODE
⚠	HTTP-Cluster	192.168.110.150	80	0	HTTP	Layer 7	Proxy
	REAL SERVER	*IP*	*PORTS*	*WEIGHT*	*CONNS*		
⬆	*RIP1*	*192.168.110.240*	*80*	*100*	*0*	*Drain*	*Halt*
⚙	*RIP2*	*192.168.110.241*	*80*	*0*	*0*	*Online (halt)*	
⬇	*RIP3*	*192.168.110.242*	*80*	*100*	*0*	*Drain*	*Halt*

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 WUI. Details of both can be found in the administration manual.

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 HA pair, you'll need 3 EIPS's - 1 for the primary interface on each instance and 1 for the VIP.

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

- Please refer to the steps on pages 10-13.
- Right-click the instance and select: *Networking > Change Source/Dest. Check* and ensure this is disabled

Step 2 – Prepare both instances for pairing

- Using the WUI option: *Local configuration > Execute Shell Command* run the following command on both appliances:

```
lb_enable_root enable
```

Step 3 – Verify Security Group Settings

Ensure that the security group used by both instances has the following 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)

N.B. 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:

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

  ec2-replace-route rtb-15127270 -r 0.0.0.0/0 -i 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
- On the slave instance select the menu option: *Cluster Configuration > Heartbeat Advanced* and add the following line:

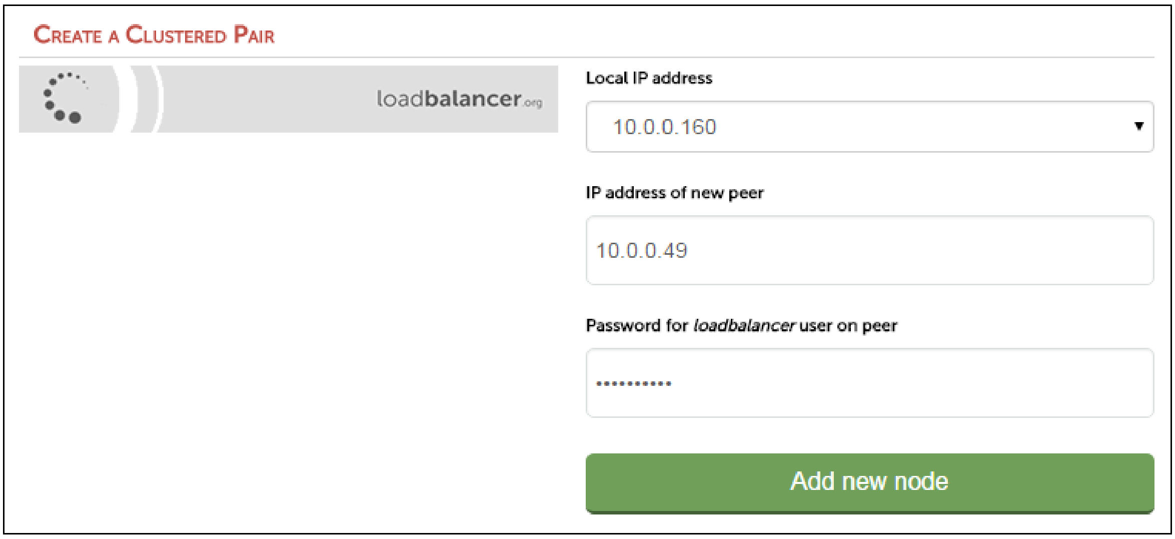
  ec2-replace-route rtb-15127270 -r 0.0.0.0/0 -i i-f45ejc53 --region eu-west-1

  *(change **rtb-15127270** to the Route Table ID of the table associated with your real servers subnet)*
  *(change **i-f40efc59** 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 master instance. It's run automatically each time the slave becomes active

Step 5 – Configure High-Availability

- Connect to the WUI on the master unit
- Select the menu option: *Cluster Configuration > High Availability Configuration*



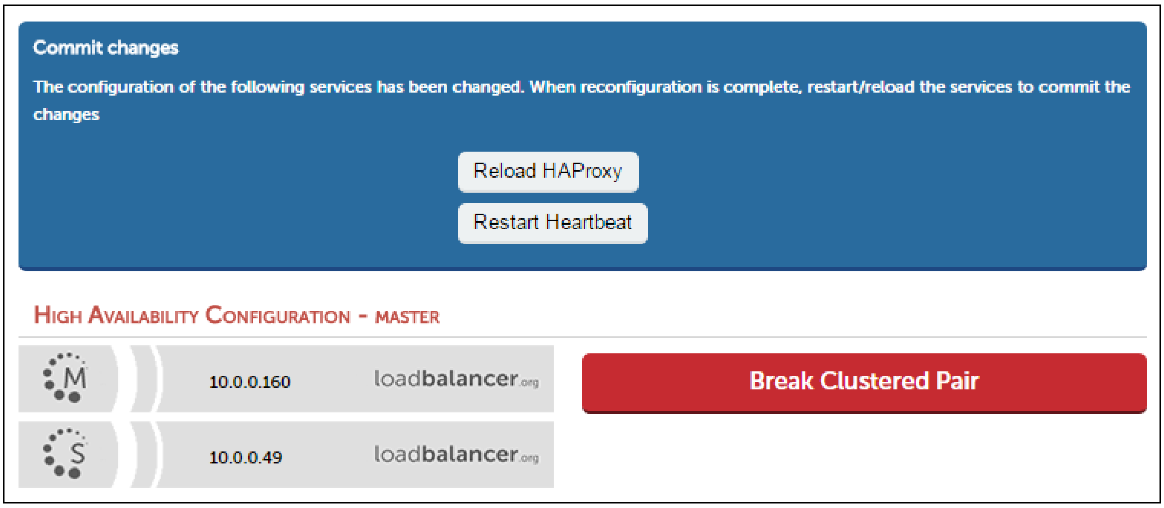
CREATE A CLUSTERED PAIR

Local IP address: 10.0.0.160

IP address of new peer: 10.0.0.49

Password for *loadbalancer* user on peer: ••••••••••

- In the *IP address of new peer* field, enter the slave appliances private IP address
- In the Password for *loadbalancer user on peer* field enter the *Instance-ID* of the slave appliance
- Click **Add new node**
- Once the pairing configuration has finished, any required service restart messages and the confirmed pair message will be displayed as shown below:



Commit changes

The configuration of the following services has been changed. When reconfiguration is complete, restart/reload the services to commit the changes

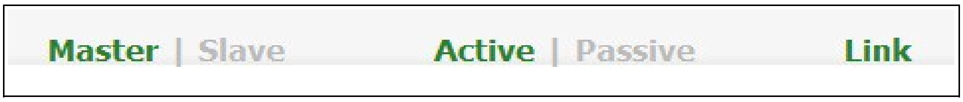
HIGH AVAILABILITY CONFIGURATION - MASTER

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

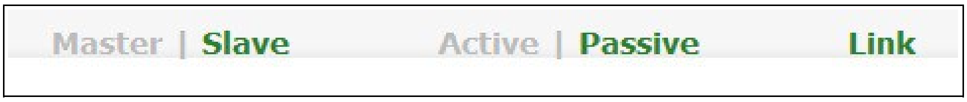
Step 6 – Verify Synchronization State

- Once all services have restarted, the sychronization process will be complete
- Verify that the status on the master & slave is as follows:

Master Unit:

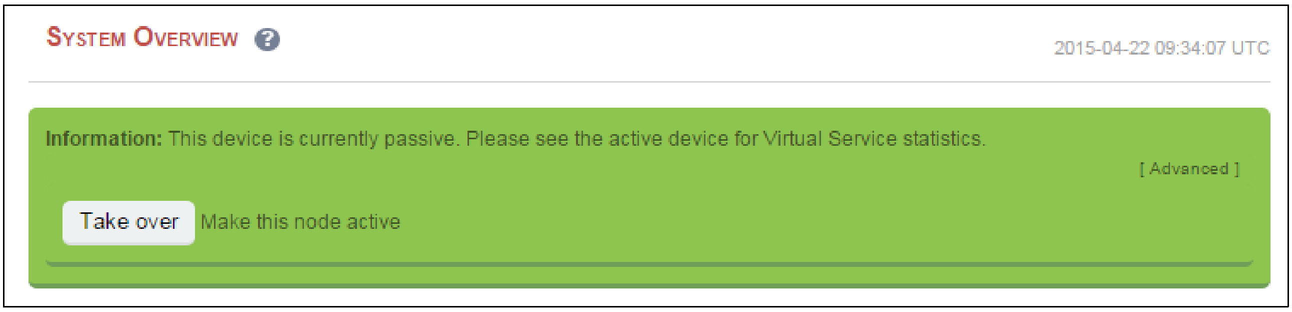


Slave Unit:



N.B. 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



SYSTEM OVERVIEW

2015-04-22 09:34:07 UTC

Information: This device is currently passive. Please see the active device for Virtual Service statistics.

Make this node active

Other possible 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
Master | Slave	**Active** | **Passive**	**Link**	this is a master unit, a slave unit has been defined, but the link is down (e.g. serial cable unplugged) so the state cannot be determined. In this case the floating IP's may be active on both units. ***Action****: check & verify the heartbeat configuration, check heartbeat logs & if required restart heartbeat on both units*

Loadbalancer.org Technical Support

If you have any questions regarding the appliance don't hesitate to contact the support team support@loadbalancer.org or your local reseller.

For more details please refer to the administration manual:

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

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 WUI, they are also auto-configured in EC2. To configure the required IAM role:

- In the AWS Console, under the **Security & Identity** section select the **Identity & Access Management** Option
- Select **Policies** in the Dashboard
- Click **Create Policy**
- Click **Select** next to *Create Your Own Policy*
- Enter a suitable *Policy Name*
- Copy and paste the following policy definition:

```
{
  "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": "*"
    }
  ]
}
```

- Click **Create Policy**
- Now select **Roles** in the Dashboard

- Click **Create New Role**
- Specify a suitable name and click **Next Step**
- Click **Select** next to *Amazon EC2*
- Select the Policy just created
- Click **Next Step** and then click **Create Role**

Use this new role when setting up your instances

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

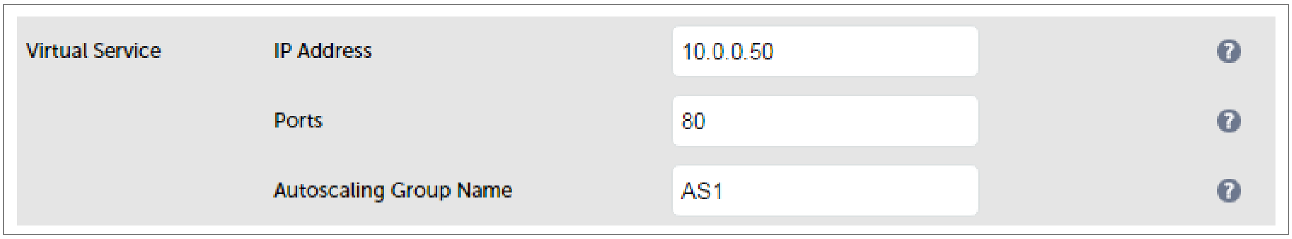
Using the EC2 Dashboard, create your launch configuration and auto-scaling group according to your requirements.

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:



Virtual Service	IP Address	10.0.0.50	?
	Ports	80	?
	Autoscaling Group Name	AS1	?

Now save the updated configuration and restart services as prompted.

N.B. 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 - Install cloud-init on your load balancer

This enables settings of the new auto scaled instance to be configured at initial launch.

- Connect via SSH – see page 19 of this guide for more details
- Login as **lbuser** , no password will be required
- Once logged in, enter the following command

```
sudo yum install cloud-init
```

When prompted, enter 'Y' to continue with the installation of the required packages

Step 3 - Create an image of the instance

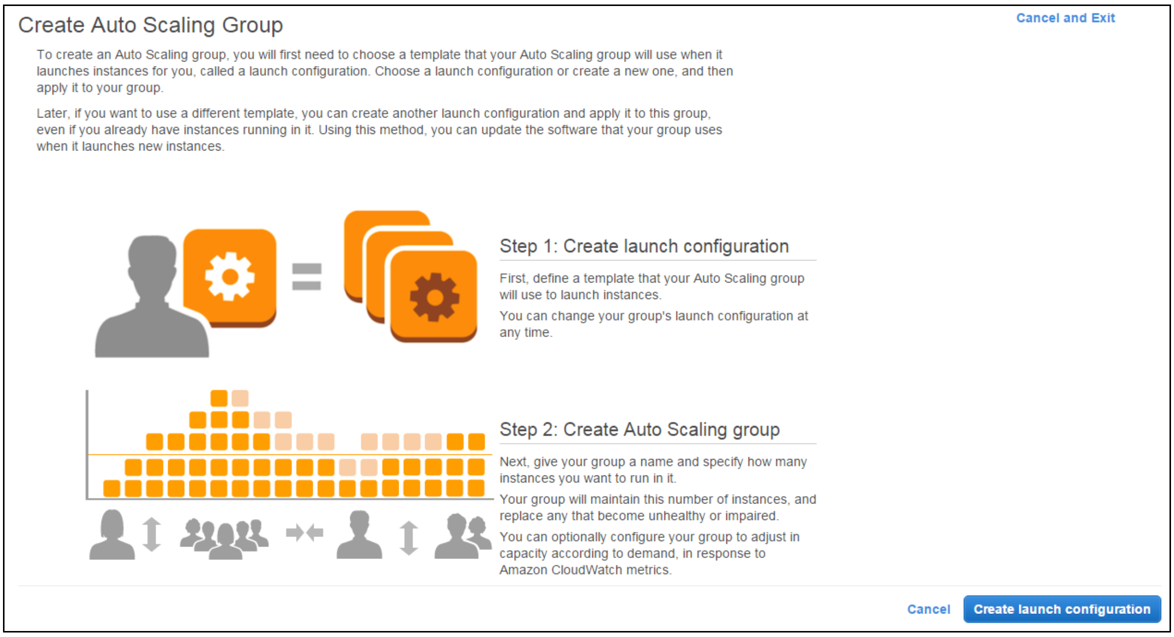
This will be the source image when new instances are deployed.

- Right click the running instance and select: *Image > Create Image*
- Enter an appropriate name & description for the image – e.g. **AS-LB-Recovery, LB recovery image**
- 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 4 - Configure AWS Auto Scaling

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

- Under *AUTO SCALING* select **Launch Configurations**
- Click the **Create Auto Scaling Group** button
- If no *Launch Configuration* exists, you'll be prompted to create one as shown below (Step 1)



Create Auto Scaling Group

To create an Auto Scaling group, you will first need to choose a template that your Auto Scaling group will use when it launches instances for you, called a launch configuration. Choose a launch configuration or create a new one, and then apply it to your group.

Later, if you want to use a different template, you can create another launch configuration and apply it to this group, even if you already have instances running in it. Using this method, you can update the software that your group uses when it launches new instances.

Step 1: Create launch configuration

First, define a template that your Auto Scaling group will use to launch instances.

You can change your group's launch configuration at any time.

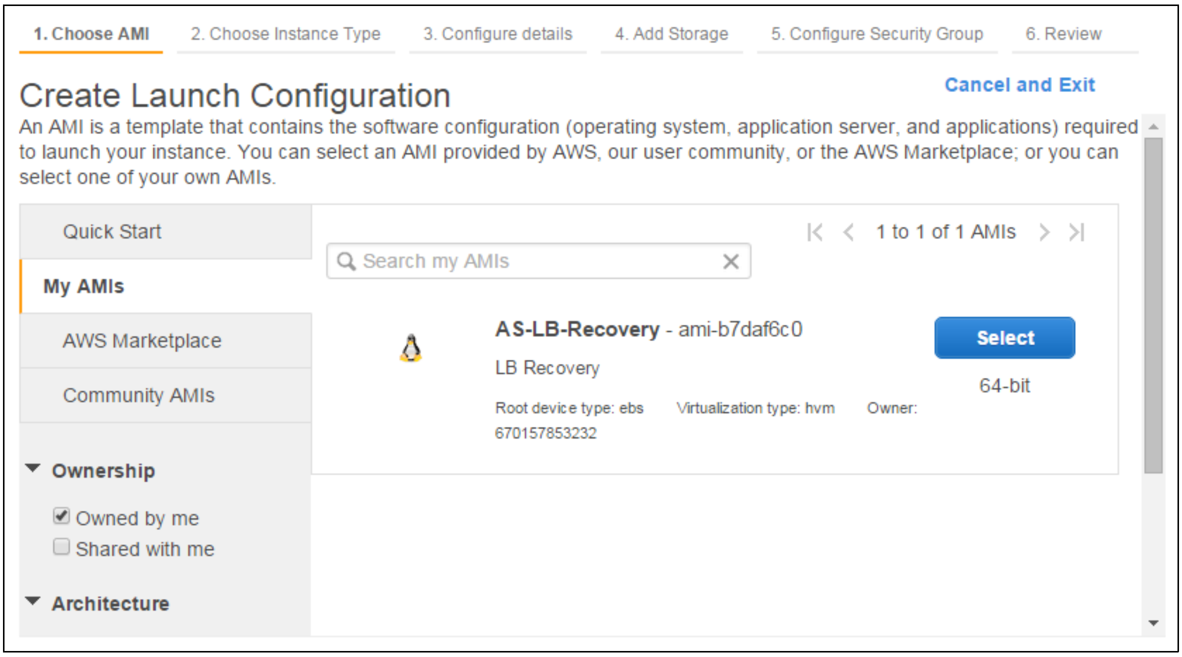
Step 2: Create Auto Scaling group

Next, give your group a name and specify how many instances you want to run in it.

Your group will maintain this number of instances, and replace any that become unhealthy or impaired.

You can optionally configure your group to adjust in capacity according to demand, in response to Amazon CloudWatch metrics.

- Click the **Create Launch Configuration** button
- Select **My AMIs**



Create Launch Configuration

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Ownership

☑ Owned by me
□ Shared with me

Architecture

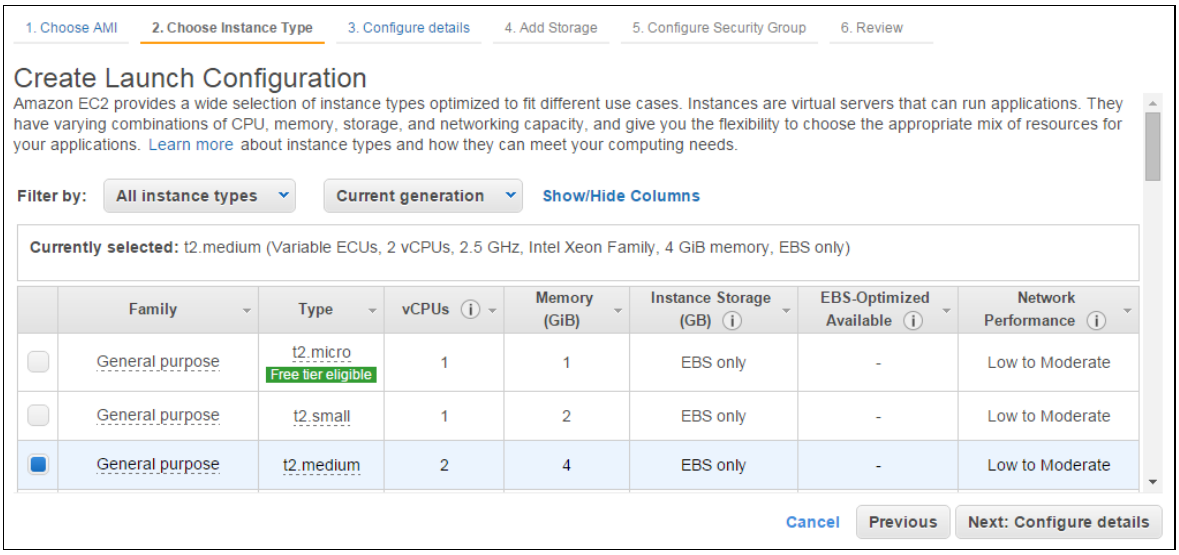
AS-LB-Recovery - ami-b7daf6c0

LB Recovery

Root device type: ebs Virtualization type: hvm Owner: 670157853232

64-bit

- Click the **Select** button next to the AMI just created



Create Launch Configuration

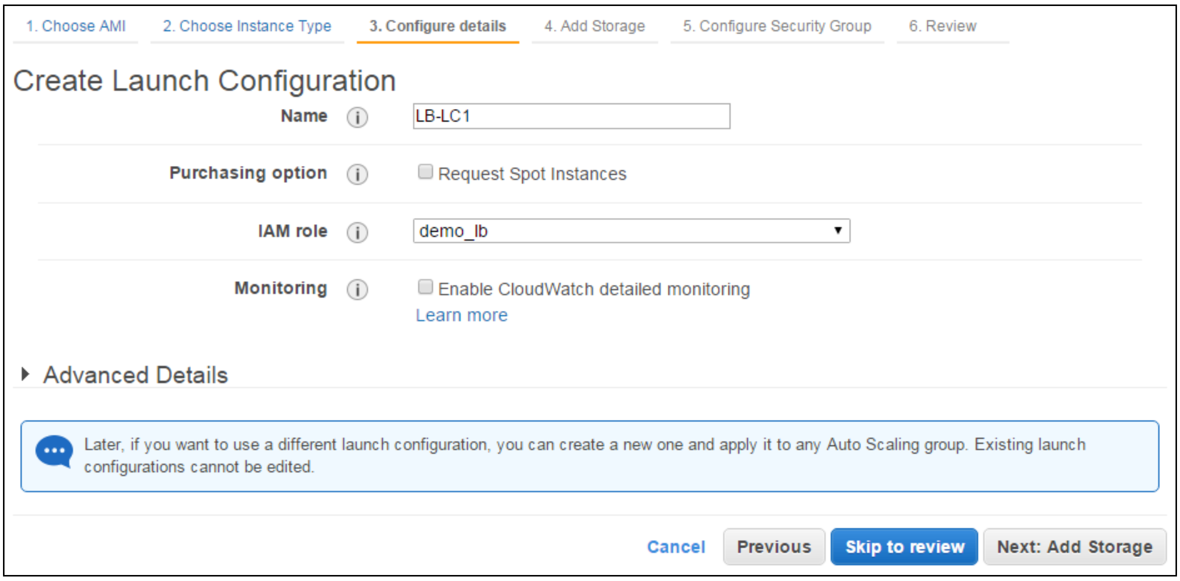
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance types | Current generation | Show/Hide Columns

Currently selected: t2.medium (Variable ECUs, 2 vCPUs, 2.5 GHz, Intel Xeon Family, 4 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
□	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
■	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate

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



Create Launch Configuration

Name: LB-LC1

Purchasing option: □ Request Spot Instances

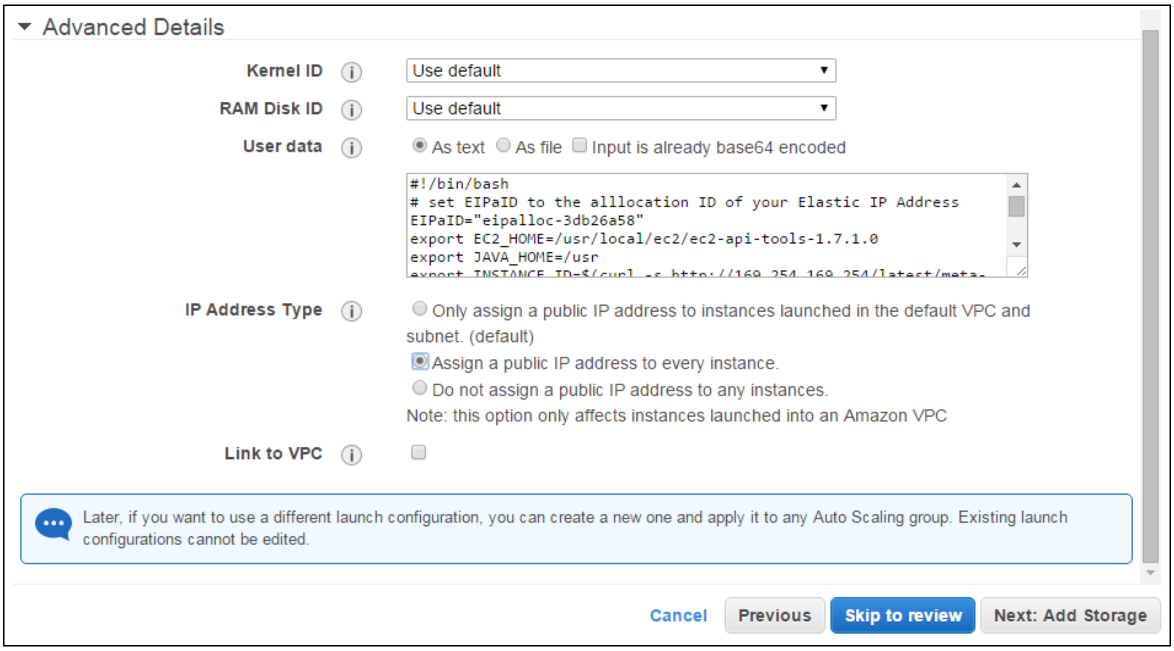
IAM role: demo_lb

Monitoring: □ Enable CloudWatch detailed monitoring
Learn more

Advanced Details

Later, if you want to use a different launch configuration, you can create a new one and apply it to any Auto Scaling group. Existing launch configurations cannot be edited.

- Define a name for the Launch Configuration, e.g. **LB-LC1**
- Select the same IAM role as was used for the original load balancer
- 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:



Kernel ID ⓘ Use default ▾

RAM Disk ID ⓘ Use default ▾

User data ⓘ ● As text ○ As file □ Input is already base64 encoded

```
#!/bin/bash
# set EIPaID to the alllocation ID of your Elastic IP Address
EIPaID="eipalloc-3db26a58"
export EC2_HOME=/usr/local/ec2/ec2-api-tools-1.7.1.0
export JAVA_HOME=/usr
export INSTANCE_ID=$(curl -s http://169.254.169.254/latest/meta-
```

IP Address Type ⓘ

○ Only assign a public IP address to instances launched in the default VPC and subnet. (default)

● Assign a public IP address to every instance.

○ Do not assign a public IP address to any instances.

Note: this option only affects instances launched into an Amazon VPC

Link to VPC ⓘ □

Later, if you want to use a different launch configuration, you can create a new one and apply it to any Auto Scaling group. Existing launch configurations cannot be edited.

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 EC2_HOME=/usr/local/ec2/ec2-api-tools-1.7.1.0
export JAVA_HOME=/usr
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/ec2/ec2-api-tools-1.7.1.0/bin/ec2-associate-address -i $INSTANCE_ID
--region $LBregion --allow-reassociation -a $EIPid -private-ip-address $IPV4 >
/var/log/lbas.log 2>&1
```

NOTE: make the following changes to the above script to suit your environment:

i) change **EIPid** in line 3 to the allocation ID of your EIP - this can be found in the lower information pane for the EIP

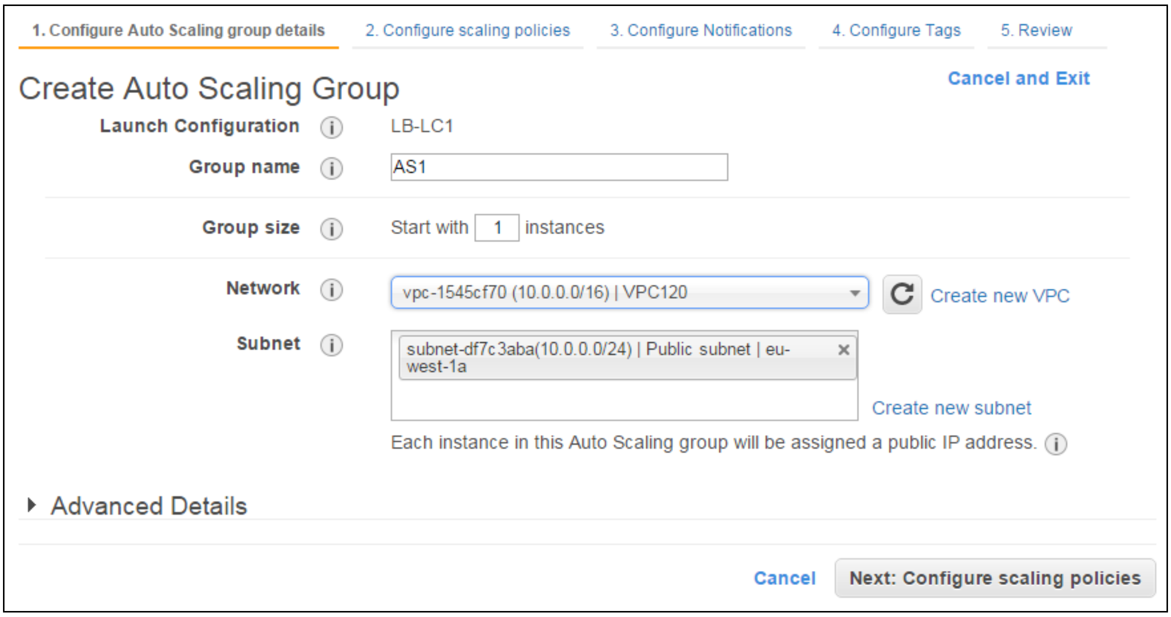
ii) change **LBregion** in line 5 to the appropriate region

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

Now continue as follows:

- Click **Next: Add Storage**
- Click **Next: Configure Security Group**
- Select the same security group as used for the original load balancer instance
- Click **Review**
- Click **Create Launch Configuration**
- Configure the required *key pair* option
- Click **Create Launch Configuration**

You'll now be prompted to enter details for the Auto Scaling group:


1. Configure Auto Scaling group details
2. Configure scaling policies
3. Configure Notifications
4. Configure Tags
5. Review
Cancel and Exit

Create Auto Scaling Group

Launch Configuration: LB-LC1

Group name: AS1

Group size: Start with 1 instances

Network: vpc-1545cf70 (10.0.0.0/16) | VPC120

Subnet: subnet-df7c3aba(10.0.0.0/24) | Public subnet | eu-west-1a

Each instance in this Auto Scaling group will be assigned a public IP address.

Create new VPC
Create new subnet
Advanced Details
Cancel
Next: Configure scaling policies

- Enter an appropriate *Group name* , e.g. **AS1**
- Select the correct VPC for your environment
- Select the correct subnet for your environment
- If required, expand *Advanced Details* and change the *Health Check grace period* from the 300s default value
- Click **Next: Configure scaling policies**
- Leave the option set to *Keep this group at its initial size* and click **Next: Configure Notifications**
- Configure any required notifications and Click **Next: Configure Tags**
- Define any required Tags, e.g. **Name = LB - autoscaled** , etc.
- Click **Review**
- 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 WUI 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 Alt Pempelfort 2 40211 Düsseldorf Germany Tel : +49 (0)211 9793 7203 Fax : +49 (0)30 920 383 6495 Email (sales) : vertrieb@loadbalancer.org Email (support) : support@loadbalancer.org