



Enterprise AWS Quick Start Guide

v7.6.4

rev. 1.0.6

Copyright © 2002 – 2015 Loadbalancer.org, Inc





Table of Contents

Introduction.....	4
About Enterprise AWS.....	4
Version 7.6.4.....	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
Instance Type.....	7
Deploying Enterprise AWS.....	8
STEP 1 - Create a VPC.....	8
STEP 2 – Accessing & Deploying the AMI.....	10
Checking your Subscriptions.....	14
Accessing the Enterprise AWS WUI.....	14
Using the Enterprise AWS WUI.....	15
Accessing Enterprise AWS using SSH.....	18
Using Linux.....	18
Using Windows.....	18
Accessing Enterprise AWS using SCP.....	21
Using Linux.....	21
Using Windows.....	21
Configuration Examples.....	22
1) Load balancing Web Servers – Single Subnet, 1 arm, Layer 7.....	22
a) Setting up AWS.....	22
b) Setting up the Virtual Service.....	22
c) Setting up the Real Servers.....	22
d) Applying the new Layer 7 Settings.....	23
e) Associating the VIP with an Elastic IP Address (If access from the Internet is required).....	23
2) Load balancing Web Servers - Dual Subnet, 2 arm, Layer 7.....	24
a) Setting up AWS.....	24
b) Configuring the second Network Interface.....	24
c) Setting up the Virtual Service.....	24
d) Setting up the Real Servers.....	25
e) Applying the new Layer 7 Settings.....	25
f) Associating the VIP with an Elastic IP Address (If access from the Internet is required).....	26
3) Load balancing Web Servers - Dual Subnet, 1 arm, Layer 7, with Transparency.....	27
a) Setting up AWS.....	27
b) Setting up the Virtual Service.....	27
c) Setting up the Real Servers.....	28
d) Configuring Layer 7 – Advanced Settings.....	29
e) Applying the new Layer 7 Settings.....	29
f) Associating the VIP with an Elastic IP Address (If access from the Internet is required).....	29
4) Load balancing Web Servers - Single Subnet, 1 arm, Layer 7, with SSL Termination.....	30
a) Setting up AWS.....	30
b) Setting up the Virtual Service.....	30
c) Setting up the Real Servers.....	30
d) Configuring SSL Termination.....	31

e) Applying the new Settings.....	32
f) Associating the VIP with an Elastic IP Address (If access from the Internet is required).....	32
5) Load balancing RD Connection Broker - Dual Subnet, 1 arm, Layer 7.....	34
a) Setting up AWS.....	34
b) Setting up the Virtual Service.....	34
c) Setting up the Real Servers.....	35
d) Applying the new Layer 7 Settings.....	36
e) Associating the VIP with an Elastic IP Address (If access from the Internet is required).....	36
6) Load balancing Web Servers - Single Subnet, 1 arm, Layer 4.....	37
a) Setting up AWS.....	37
b) Setting up the Virtual Service.....	37
c) Setting up the Real Servers.....	38
d) Associating the VIP with an Elastic IP Address (If access from the Internet is required).....	38
Verifying Load Balanced Services.....	39
Diagnosing VIP Connection Problems.....	39
Taking Real Servers Offline (Halting).....	40
Log Files.....	41
Configuring High Availability using two Instances (Master & Slave).....	41
Loadbalancer.org Technical Support.....	45
Appendix.....	46
1. IAM Role Configuration.....	46
2. Configuring Auto-Scaling.....	47
3. Company Contact Information.....	48

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 41.

Version 7.6.4

V7.6.4 includes many updates, a full list can be found [here](#). Enterprise AWS is based on our main hardware/virtual product. Previously, it was based on a completely different code base and development road map. The advantage is that Enterprise AWS now supports many of the same features as the hardware & virtual based products. 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
- The WUI is not accessible on HTTP port 9080, only HTTPS port 9443
- HA (i.e. a master/slave clustered pair) must currently be configured manually
(please see page 41 for more details)

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

<u>Acronym</u>	<u>Terminology</u>
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). This is because the appliance requires a minimum of 2 private IP addresses – one for the interface and one for the load balanced VIP. EC2 classic only supports a single private IP address and a corresponding EIP mapped via NAT.

The easiest way to 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

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.

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>

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 1 - Create a VPC

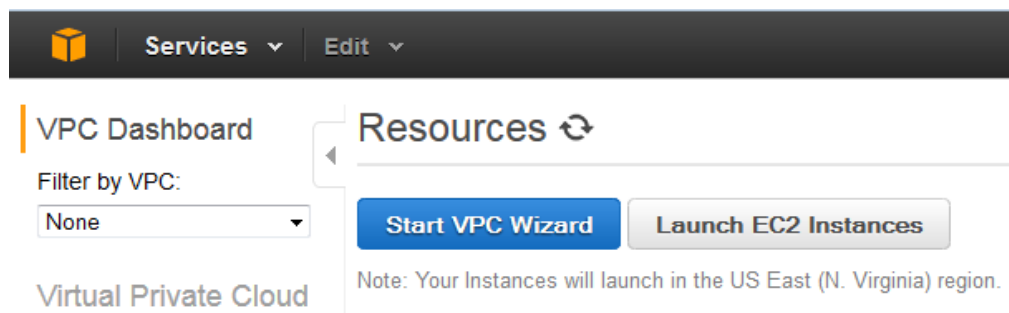
For a manually created VPC, the key steps are:

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:

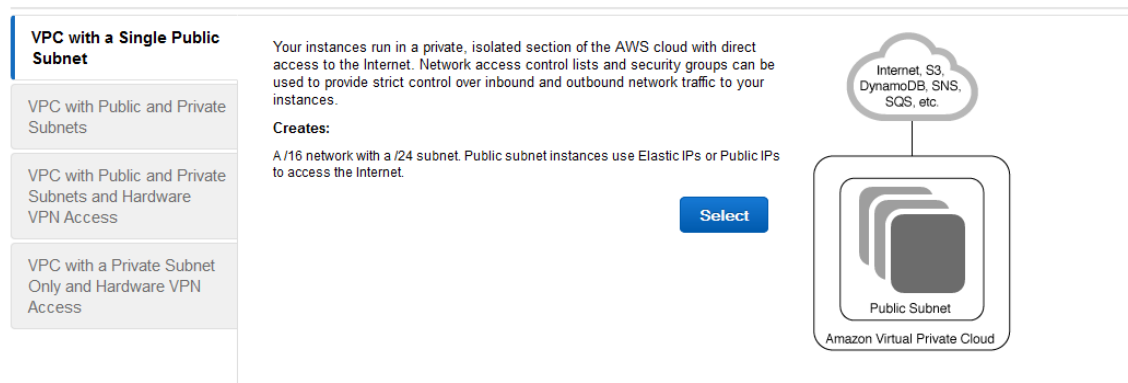
- In the VPC dashboard, click **Start VPC Wizard**



- 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



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

IP CIDR block:*	<input type="text" value="10.0.0.0/16"/>	(65531 IP addresses available)
VPC name:	<input type="text" value="VPC 100"/>	

Public subnet:*	<input type="text" value="10.0.0.0/24"/>	(251 IP addresses available)
Availability Zone:*	<input type="text" value="No Preference"/>	
Subnet name:	<input type="text" value="Public subnet"/>	

You can add more subnets after AWS creates the VPC.

Add endpoints for S3 to your subnets

Subnet:	<input type="text" value="None"/>	
---------	-----------------------------------	--

Enable DNS hostnames:*	<input checked="" type="radio"/> Yes <input type="radio"/> No
Hardware tenancy:*	<input type="text" value="Default"/>
Enable ClassicLink:*	<input type="radio"/> Yes <input checked="" type="radio"/> 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 :

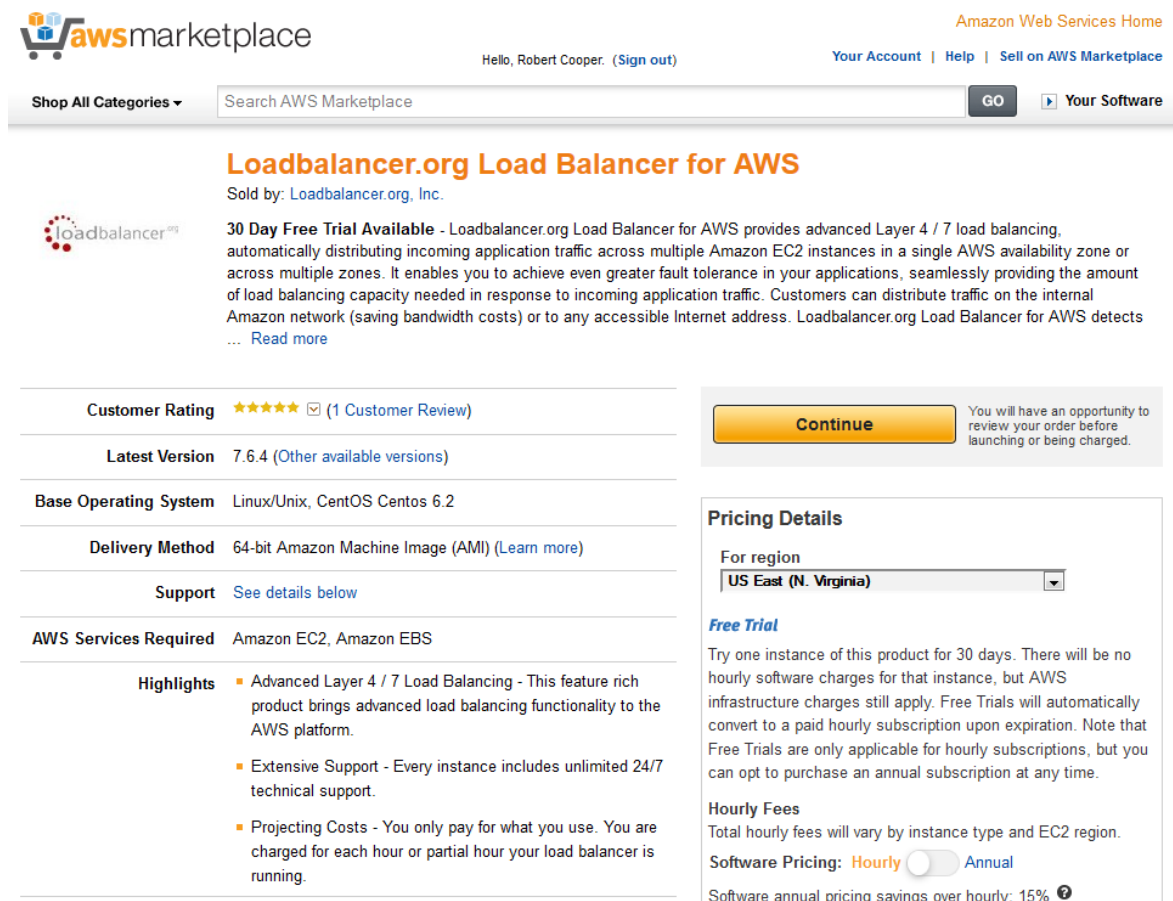
<http://awsdocs.s3.amazonaws.com/VPC/latest/vpc-ug.pdf>

STEP 2 – Accessing & Deploying the AMI

To access and deploy the AMI:

- Browse to the following Amazon Marketplace URL:

https://aws.amazon.com/marketplace/pp/B00S05UW50/ref=sp_mpg_product_title?ie=UTF8&sr=0-2



The screenshot shows the AWS Marketplace product page for 'Loadbalancer.org Load Balancer for AWS'. The page header includes the AWS Marketplace logo, a search bar, and navigation links. The product title is 'Loadbalancer.org Load Balancer for AWS', sold by 'Loadbalancer.org, Inc.'. A '30 Day Free Trial Available' banner is present. The product description states it provides advanced Layer 4 / 7 load balancing. A table lists product details: Customer Rating (5 stars, 1 review), Latest Version (7.6.4), Base Operating System (Linux/Unix, CentOS 6.2), Delivery Method (64-bit Amazon Machine Image (AMI)), Support (See details below), and AWS Services Required (Amazon EC2, Amazon EBS). A 'Continue' button is visible. The 'Pricing Details' section shows the region set to 'US East (N. Virginia)', a 'Free Trial' offer, and pricing options for 'Hourly' and 'Annual' subscriptions. The 'Hourly Fees' section states that total hourly fees vary by instance type and EC2 region. The 'Software Pricing' section shows a toggle for 'Hourly' and 'Annual' pricing, with a note that software annual pricing savings over hourly is 15%.

Amazon Web Services Home

Hello, Robert Cooper. (Sign out) Your Account | Help | Sell on AWS Marketplace

Shop All Categories Search AWS Marketplace GO Your Software

Loadbalancer.org Load Balancer for AWS

Sold by: Loadbalancer.org, Inc.

30 Day Free Trial Available - Loadbalancer.org Load Balancer for AWS provides advanced Layer 4 / 7 load balancing, automatically distributing incoming application traffic across multiple Amazon EC2 instances in a single AWS availability zone or across multiple zones. It enables you to achieve even greater fault tolerance in your applications, seamlessly providing the amount of load balancing capacity needed in response to incoming application traffic. Customers can distribute traffic on the internal Amazon network (saving bandwidth costs) or to any accessible Internet address. Loadbalancer.org Load Balancer for AWS detects ... [Read more](#)

Customer Rating	★★★★★ (1 Customer Review)
Latest Version	7.6.4 (Other available versions)
Base Operating System	Linux/Unix, CentOS 6.2
Delivery Method	64-bit Amazon Machine Image (AMI) (Learn more)
Support	See details below
AWS Services Required	Amazon EC2, Amazon EBS
Highlights	<ul style="list-style-type: none">Advanced Layer 4 / 7 Load Balancing - This feature rich product brings advanced load balancing functionality to the AWS platform.Extensive Support - Every instance includes unlimited 24/7 technical support.Projecting Costs - You only pay for what you use. You are charged for each hour or partial hour your load balancer is running.

Continue You will have an opportunity to review your order before launching or being charged.

Pricing Details

For region
US East (N. Virginia)

Free Trial
Try one instance of this product for 30 days. There will be no hourly software charges for that instance, but AWS infrastructure charges still apply. Free Trials will automatically convert to a paid hourly subscription upon expiration. Note that Free Trials are only applicable for hourly subscriptions, but you can opt to purchase an annual subscription at any time.

Hourly Fees
Total hourly fees will vary by instance type and EC2 region.

Software Pricing: **Hourly** ☐ ☒ Annual

Software annual pricing savings over hourly: 15%

- Click **Continue**
- Select the required pricing options (hourly or annual)
- Click the **Launch with EC2 Console** button next to the required Region

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
<input type="checkbox"/>	General purpose	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
<input checked="" type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	m4.large	2	8	EBS only	Yes	Moderate
<input type="checkbox"/>	General purpose	m4.xlarge	4	16	EBS only	Yes	High
<input type="checkbox"/>	General purpose	m4.2xlarge	8	32	EBS only	Yes	High

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Instance Details](#)

- Select the required instance type – **t2.medium** is recommended, but depends on your requirements
- Click **Next: Configure Instance Details**

1. Choose AMI 2. Choose Instance Type **3. Configure Instance** 4. Add Storage 5. Tag Instance 6. Configure Security Group

Step 3: Configure Instance Details

advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances i

Purchasing option i ☐ Request Spot Instances

Network i [Create new VPC](#)

Subnet i [Create new subnet](#)
251 IP Addresses available

Auto-assign Public IP i

IAM role i

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

- 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”
- Select a suitable **IAM Role**. The role can simply have “**Amazon EC2 Full Access**” for the “**Amazon EC2**” AWS Service Role or for more granular configuration, please refer to section 1 in the Appendix.

N.B. 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 are located (same or different subnet as the load balancer) and the load balancing mode. Please refer to the deployment examples starting on page 22 for more details.

- Click **Next: Add Storage**

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

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 <small>i</small>	Device <small>i</small>	Snapshot <small>i</small>	Size (GiB) <small>i</small>	Volume Type <small>i</small>	IOPS <small>i</small>	Delete on Termination <small>i</small>	Encrypted <small>i</small>
Root	/dev/sda1	snap-05bfa4b6	4	General Purpose (SSD)	12 / 3000	<input checked="" type="checkbox"/>	Not Encrypted

[Add New Volume](#)

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.

[Cancel](#)
[Previous](#)
[Review and Launch](#)
[Next: Tag Instance](#)

- Click **Next: Tag Instance**

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group

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 <small>(127 characters maximum)</small>	Value <small>(255 characters maximum)</small>
Name	Load Balancer

[Create Tag](#) (Up to 10 tags maximum)

[Cancel](#)
[Previous](#)
[Review and Launch](#)
[Next: Configure Security Group](#)

- 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	×

Add Rule



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.

Cancel

Previous

Review and Launch

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

Cancel Launch Instances

- 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 is required to ensure that the load balancer can 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:

The screenshot shows the AWS Marketplace console interface. At the top, there's a navigation bar with the AWS Marketplace logo, a search bar, and links for 'Your Account', 'Help', and 'Sell on AWS Marketplace'. Below the navigation bar, the 'Your Account' section is active, displaying 'Your Software Subscriptions (1)'. A table lists the subscriptions, with one entry for 'Load Balancer.org Enterprise EC2'. The table has columns for 'Products', 'Instances', and 'Actions'. The 'Instances' column shows '1 active' instance with ID 'i-d76acdfb' in a 'running' state. The 'Actions' column includes buttons for 'Usage Instructions', 'Launch more software', and 'Buy annual subscription'. A 'Cancel subscription' button is also visible under the product details.

Accessing the Enterprise AWS WUI

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

i.e.

https://<Public DNS name>:9443

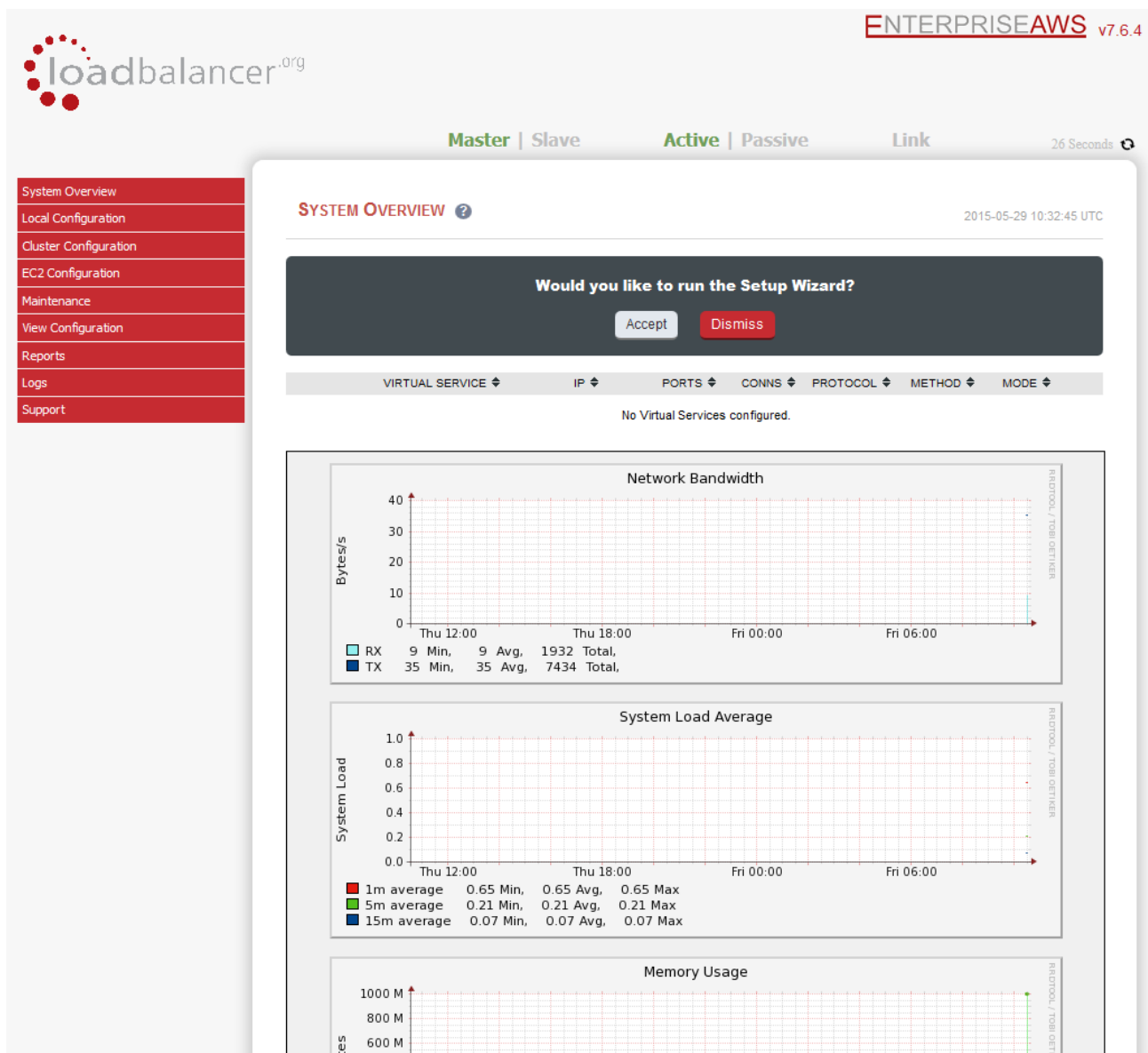
or

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:



Using the Enterprise AWS WUI

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/loadbalanceradministrationv7.pdf>

Local Configuration > Network Interface Configuration

NETWORK INTERFACE CONFIGURATION

IP Address Assignment



eth0

eth0

10.0.0.201/24
10.0.0.220/24

Configure Interfaces

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

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

- Click **Configure Interfaces** to apply any changes

Cluster Configuration > Heartbeat Advanced

HEARTBEAT FAILOVER SCRIPT

```
1 # Heartbeat Failover Commands
2 # Here you can enter commands that run when Heartbeat fails over.
3 # These commands are not replicated across appliances.
4
5
6
7
8
9
```

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]
----------------	-------------------	------------

[Allocate New Elastic IP](#) ?

Notes:

- This menu option is used to define how Elastic IP's relate to private IP's
- Row-1 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 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 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 Enterprise AWS 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'

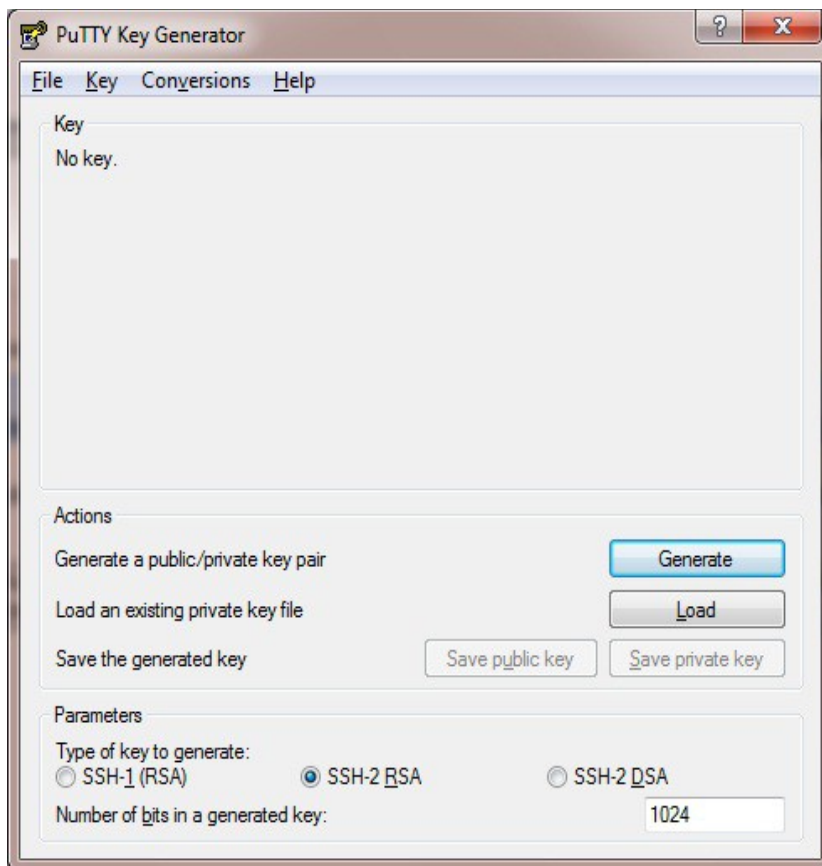
```
ssh -i /path-where-saved/ec2-key-name.pem lbuser@1.2.3.4
```

or

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

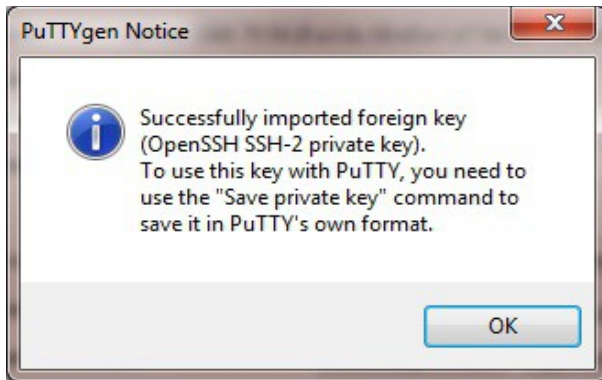
Using Windows

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:

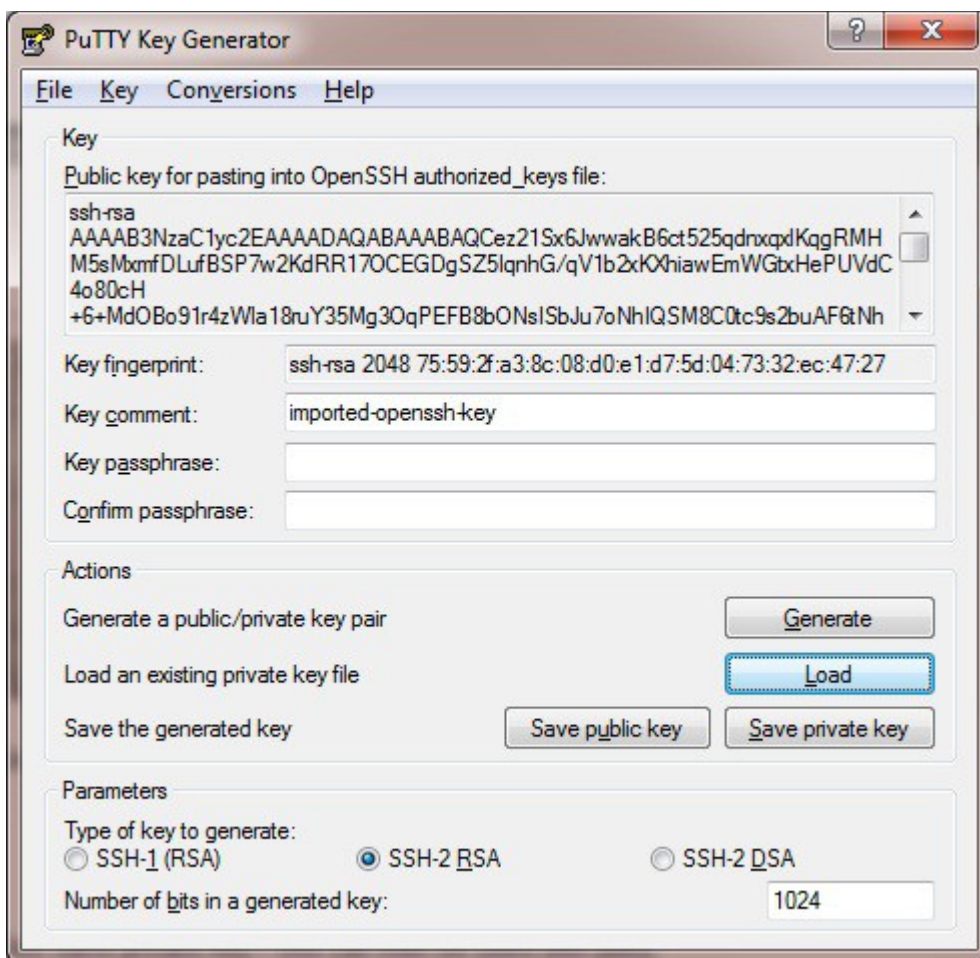


Click **Load**, change the file-type to all files and select the pem file saved earlier when creating your Key Pair.

You should see the following message:

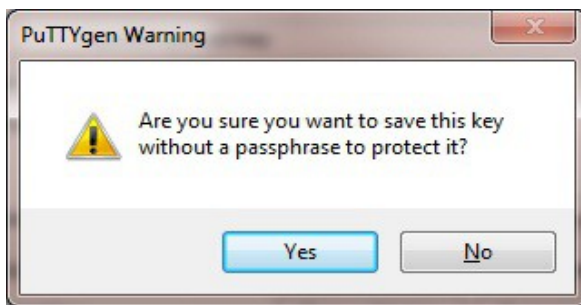


Click **OK**



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

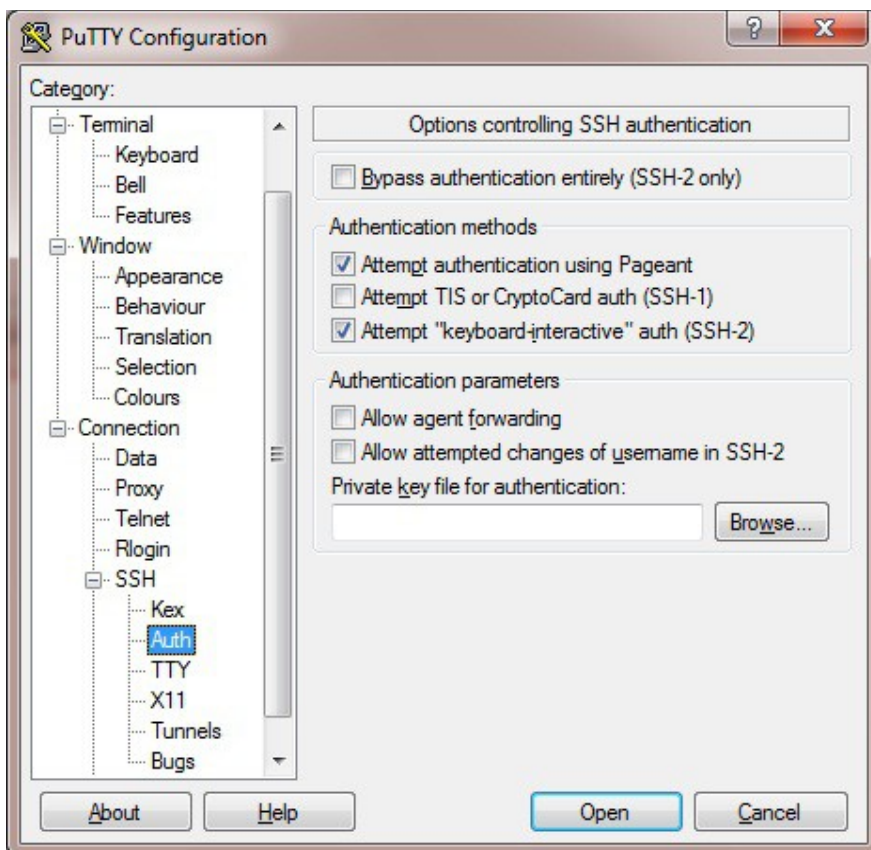
NB. You can also choose to enter an additional pass-phrase for improved security, if you don't, the following message will be displayed:



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

Now close PuTTYgen and start PuTTY

Expand the SSH section as shown below:



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

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

Accessing Enterprise AWS 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
```

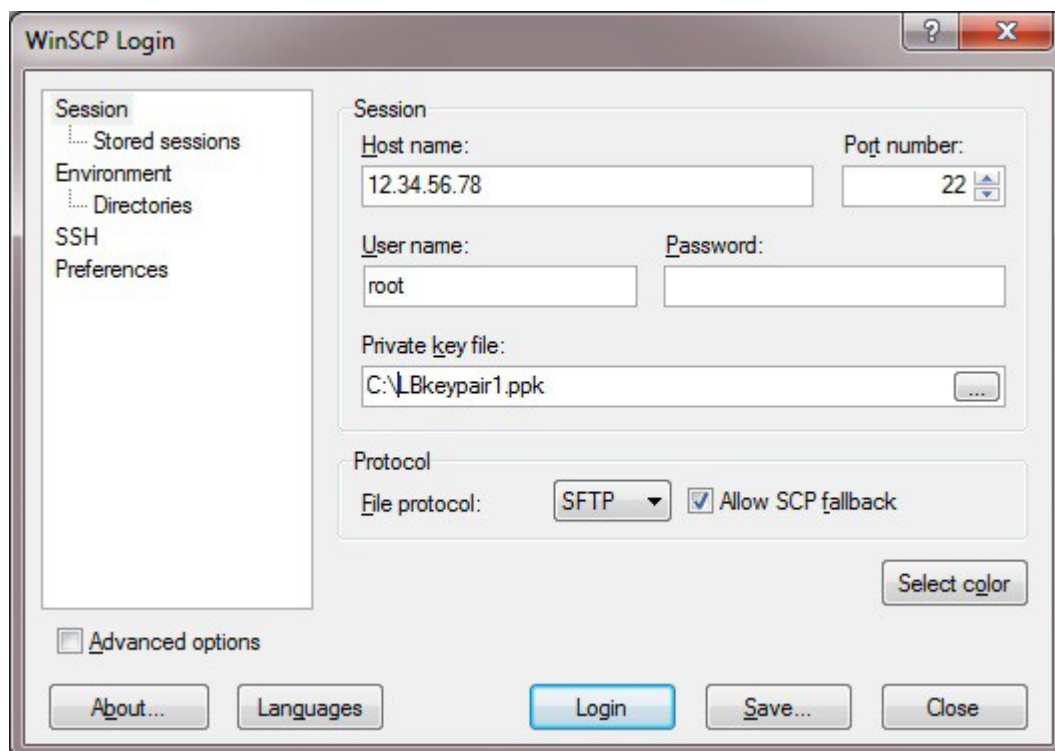
Now start SCP specifying the private key file – login as user 'lbuser'

```
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.



Click **Login**

Configuration Examples

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



NOTE : It's not possible to configure a VIP on the same IP address as any of the network interfaces. This ensures services can move between master and slave appliances.

1) Load balancing Web Servers – Single Subnet, 1 arm, 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	<input type="checkbox"/>		?

Cancel Update

- 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	<input type="text" value="Web1"/>	?
Real Server IP Address	<input type="text" value="10.0.0.23"/>	?
Real Server Port	<input type="text" value="80"/>	?
Weight	<input type="text" value="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 ?

→ [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 - Dual Subnet, 2 arm, 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 servers 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

eth1

eth0

10.0.0.220/24

eth1

10.0.2.220/24

Configure Interfaces

- 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	<input type="text" value="Web-Cluster1"/>		?
Virtual Service	IP Address	<input type="text" value="10.0.0.22"/>	?
	Ports	<input type="text" value="80"/>	?
Layer 7 Protocol	<input type="text" value="HTTP Mode"/>		?
Manual Configuration	<input type="checkbox"/>		?

- 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	<input type="text" value="Web1"/>	?
Real Server IP Address	<input type="text" value="10.0.2.50"/>	?
Real Server Port	<input type="text" value="80"/>	?
Weight	<input type="text" value="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]

Allocate New Elastic IP ?

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



NOTE : If you want to enable transparency for layer 7, please refer to example 3 on the following page. Using a 2 arm load balancer configuration and setting the real servers default gateway to be the load balancers interface in the same subnet is not supported.

3) Load balancing Web Servers - Dual Subnet, 1 arm, 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**
 - 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	igw-4b953a2e	Active	No	✗
	i-3b3f28da Robs AWS Instan...			✗

Buttons: Cancel, Save, Add another route

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	<input type="text" value="Web-Cluster1"/>		?
Virtual Service	IP Address	<input type="text" value="10.0.0.22"/>	?
	Ports	<input type="text" value="80"/>	?
Layer 7 Protocol	<input type="text" value="HTTP Mode"/>		?
Manual Configuration	<input type="checkbox"/>		?

- 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	<input type="text" value="Web1"/>	?
Real Server IP Address	<input type="text" value="10.0.2.50"/>	?
Real Server Port	<input type="text" value="80"/>	?
Weight	<input type="text" value="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]

Allocate New Elastic IP ?

- 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

4) Load balancing Web Servers - Single Subnet, 1 arm, 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	<input type="checkbox"/>		?

CancelUpdate

- 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	<input type="text" value="Web1"/>	
Real Server IP Address	<input type="text" value="10.0.0.23"/>	
Real Server Port	<input type="text" value="80"/>	
Weight	<input type="text" value="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	<input type="text" value="SSL-WEB"/>	
Virtual Service IP address	<input type="text" value="10.0.0.22"/>	
Virtual Service Port	<input type="text" value="443"/>	
Backend Virtual Service IP Address	<input type="text" value="10.0.0.22"/>	
Backend Virtual Service Port	<input type="text" value="80"/>	
Ciphers to use	<input type="text" value="ECDHE-RSA-AES128-GCM"/>	
Do not insert empty fragments	<input checked="" type="checkbox"/>	
SSL Terminator	<input type="radio"/> Pound <input checked="" type="radio"/> STunnel	
Delay DNS Lookups	<input checked="" type="checkbox"/>	
Disable SSLv2 Ciphers	<input checked="" type="checkbox"/>	
Disable SSLv3 Ciphers	<input checked="" type="checkbox"/>	
Allow Client Renegotiation	<input checked="" type="checkbox"/>	
Disable SSL Renegotiation	<input checked="" type="checkbox"/>	
Time To Close	<input type="text" value="0"/>	
Set as Transparent Proxy	<input type="checkbox"/>	

- 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*

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]

Allocate New Elastic IP 

- 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 RD Connection Broker - Dual Subnet, 1 arm, Layer 7

This example uses 2 subnets - one subnet for the load balancer and one subnet for the connection brokers. The load balancer has a single network interface located in the first subnet. Routing rules for the second subnet where the connection brokers 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 connection broker 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**
 - 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	igw-4b953a2e	Active	No	✗
	i-3b3f28da Robs AWS Instan...			✗

Cancel Save

Add another route

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	ConnectionBroker-Cluster1		?
Virtual Service	IP Address	10.0.0.25	?
	Ports	3389	?
Layer 7 Protocol	TCP Mode		?
Manual Configuration	<input type="checkbox"/>		?

Cancel
Update

- Enter an appropriate label for the VIP, e.g. **ConnectionBroker-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 **None**
- 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	ConnectionBroker1	?
Real Server IP Address	10.0.2.50	?
Real Server Port	3389	?
Weight	100	?

Cancel
Update

- Enter an appropriate label for the RIP, e.g. **ConnectionBroker1**
- 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 connection broker 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]
----------------	-------------------	------------

Allocate New Elastic IP ?

- 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

6) Load balancing Web Servers - Single Subnet, 1 arm, Layer 4

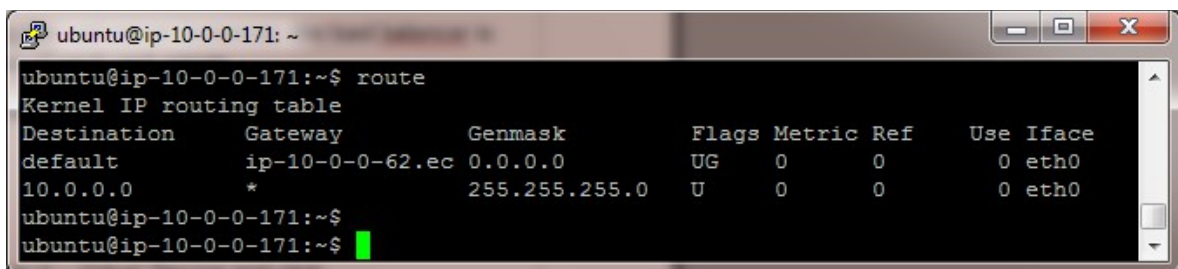
This is a simple layer 4 example using one subnet for both the load balancer and the web servers. The default gateway on the web servers must be set to be the load balancer – this ensures that return traffic goes back to the client via the load balancer, which is a requirement of layer 4 NAT mode.

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: ~  
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	<input type="text" value="Web-Cluster1"/>		?
Virtual Service	IP Address	<input type="text" value="10.0.0.22"/>	?
	Ports	<input type="text" value="80"/>	?
Protocol	<input type="text" value="TCP"/>		?
Forwarding Method	<input type="text" value="NAT"/>		?
		<input type="button" value="Cancel"/>	<input type="button" value="Update"/>

- 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	<input type="text" value="Web1"/>	?
Real Server IP Address	<input type="text" value="10.0.0.31"/>	?
Real Server Port	<input type="text" value="80"/>	?
Weight	<input type="text" value="100"/>	?
Minimum Connections	<input type="text" value="0"/>	?
Maximum Connections	<input type="text" value="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 ?

→ [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

Verifying Load Balanced Services

Various features exist on the load balancer to help monitor load balanced services. These are covered in the sections below.

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
active UP, going down
active DOWN, going up
active or backup DOWN
backup UP
backup UP, going down
backup DOWN, going up
not checked
Note: UP with load-balancing disabled is reported as "NOLB".

Display option:

- [Hide DOWN servers](#)
- [Refresh now](#)
- [CSV export](#)

External resources:

- [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	Downtme	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	0	0			1	-	Y					-
RIP1	0	0	-	0	16	-	0	2	-	56	56	21 696	3 385 782	0	0	0	0	0	0	0	42s UP	L4OK in 0ms	1	Y	-	0	0	0s	0s	-
Backend	0	0	-	0	16	-	0	2	4 000	56	56	21 696	3 385 782	0	0	0	0	0	0	0	42s UP		1	1	1		0	0s	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	0	0	200	0	0	1 464	33 111	0	0	0	0	0	0	0	42s UP		0	0	0	0		0		

Taking Real Servers Offline (Halting)

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.

Remove the network cable from one of the web servers or stop the web service/process, 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).

Replace the network cable, wait a few seconds and then refresh the browsers again. After a few refreshes they should again show different web servers.

The *System Overview* will also show the updated status as these tests are performed:

SYSTEM OVERVIEW ?

2015-04-30 08:35:41 UTC

VIRTUAL SERVICE		IP	PORTS	CONN	PROTOCOL	METHOD	MODE
	HTTP-Cluster	192.168.110.150	80	0	HTTP	Layer 7	Proxy
REAL SERVER		IP	PORTS	WEIGHT	CONN		
	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



NOTE : From v7.6.4 the System Overview supports sorting of VIPs. This can be done by clicking on the column headings or by drag & drop. For more details please refer to chapter 7 in the administration manual.

Log Files

The appliance includes several logs that are very useful when diagnosing issues. These are viewable via the Logs option in the WUI.

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.

The current version (v7.6.4) fully supports this functionality, however it must be configured manually using the following procedure:



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.

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 – Change the instance role to be 'slave'

Once the second instance is up and running:

- Connect to the WUI
- Select the menu option: *Local Configuration > Hostname & DNS*
- Change *role* to **slave**
- Click **Update**

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)

Step 4 – Perform steps (a) to (e) below on BOTH units to allow instance pairing and communication:

a) Open an SSH session to the instance

Please refer to page 18 for details on how to do this under Windows and Linux.

b) Change to the root user - run the following command:

sudo su

c) Edit the file /etc/ssh/sshd_config

N.B. Use an editor such as vi, vim or nano to do this. Under Windows, the editor build into WinSCP can also be used.

Make the following changes -

1) find & change:

AllowUsers lbuser

to

AllowUsers lbuser root

2) find & change:

PasswordAuthentication no

to

PasswordAuthentication yes

3) ensure that:

PermitRootLogin forced-commands-only

is commented out as shown below:

#PermitRootLogin forced-commands-only

Now save and close the file.

d) Restart SSH to apply the changes – run the following command:

service sshd restart

e) Generate new SSH keys and copy to the other instance – run the following commands:

ssh-keygen -t dsa

(accept all defaults)

ssh-copy-id -i /root/.ssh/id_dsa root@<Other Instances private IP address>

e.g.

ssh-copy-id -i /root/.ssh/id_dsa root@10.0.0.188

(when prompted, type 'yes', password = instance-id of the destination instance)

Step 5 – Configure Heartbeat on the Master Instance

- Connect to the WUI on the master unit
- Select the menu option: *Cluster Configuration > Heartbeat Configuration*
- In the *Slave Load Balancer Address* field define the slave appliances private IP address
- Click **Modify Heartbeat Configuration**
- Select the menu option: *View Configuration > Heartbeat Configuration* and verify that the configuration has been updated to include both master and slave, e.g. :

```
# File auto-generated by loadbalancer.org appliance
logfacility local6
uuidfrom nodename
coredumps false
keepalive 3
deadtime 10
warntime 5
initdead 30
auto_failback off
max_rexmit_delay 5000
node lbmaster
node lbslave
udpport 6694
ucast eth0 10.0.1.228
ucast eth0 10.0.1.196
```

The last 2 lines show the master and slave IP addresses

Step 6 – Synchronize the Configuration to the Slave Instance

- On the master, select the menu option: *Maintenance > Backup & Restore > Synchronization* and click the **Synchronize configuration with peer** button

Step 7 – Configure AWS CLI Commands to set the Default Route/Gateway when Failover Occurs

- 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 8 – Restart Heartbeat to apply the new Settings

- Finally, click **Restart Heartbeat** in the blue message box that appears at the top of the screen

The HA Clustered Pair configuration is now complete. The pair keep in regular contact over the network. If the master unit fails, the slave unit will take over. The system overview on master and slave should be as follows:

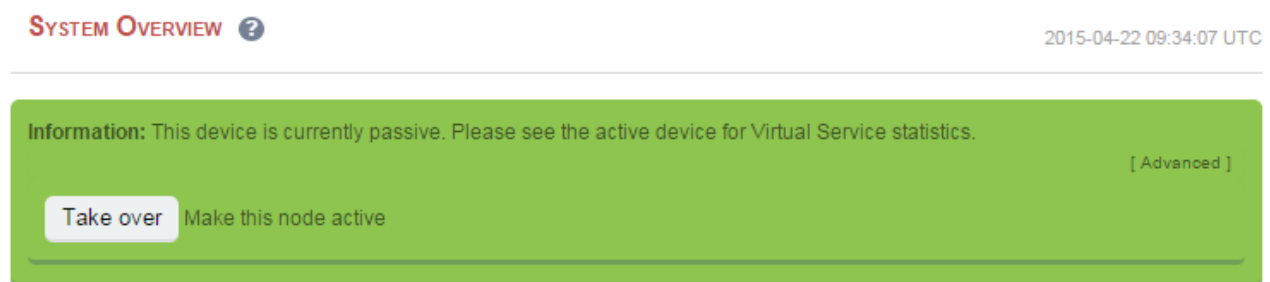
Master Unit:



Slave Unit:



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



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/v7/loadbalanceradministrationv7.6.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, 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": [
    {
      "Sid": "Stmt1424431952000",
      "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": "*"
    }
  ]
}
```

- 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 Auto-Scaling

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 - Create a new Launch AMI & Configure it to Auto-register with the load balancer at boot

This AMI will be used by the Auto-Scaling group when additional servers are required. For a Linux server, the following script should be created in the **init.d** directory to start up automatically on-boot. The script calls the **lbcli** functions on the load balancer which adds it to the load balanced group of servers.

```
#!/bin/bash
#
# chkconfig: 345 80 20
# description: AWS Agent to add autoscaling servers to your Load Balancer
# processname: lbawsscaleagent

# Loadbalancer address
LB_ADDR=192.168.1.52
# Loadbalancer ssh user
LB_USER="root"
VIP_NAME="Vip1"

case "$1" in
  start)
    LOCIP=`/usr/bin/curl -s http://169.254.169.254/latest/meta-data/local-ipv4`
    ssh $LB_USER@$LB_ADDR "lbcli --action add-rip --vip $VIP_NAME --rip_type ipv4 --rip $LOCIP
--layer 7 --ip $LOCIP --port 80 --weight 100"
    ssh $LB_USER@$LB_ADDR "service haproxy reload"
    ;;
  stop)
    LOCIP=`/usr/bin/curl -s http://169.254.169.254/latest/meta-data/local-ipv4`
    ssh $LB_USER@$LB_ADDR "lbcli --action delete-rip --vip $VIP_NAME --rip $LOCIP --layer 7 --ip
$LOCIP--port 80 --rip_type ipv4 --weight 100"
    ssh $LB_USER@$LB_ADDR "service haproxy reload"
    ;;
esac

exit 0
```

N.B. Make sure you correctly configure the values for LB_ADDR and VIP_NAME

Step 2 – Setup the Launch Configuration & Auto-Scaling Group

Now using the EC2 Dashboard, create your launch configuration and auto-scaling group specifying the AMI created in step 1 and your required scaling policies.

N.B. For more information on configuring Auto-scaling, please refer to the following URL:

<http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/GettingStartedTutorial.html>

3. Company Contact Information

Website	URL : www.loadbalancer.org
North America (US)	<p>Loadbalancer.org, Inc. 270 Presidential Drive Wilmington, DE 19807 USA</p> <p>Tel : +1 888.867.9504 (24x7) Fax : +1 302.213.0122 Email (sales) : sales@loadbalancer.org Email (support) : support@loadbalancer.org</p>
North America (Canada)	<p>Loadbalancer.org Ltd 300-422 Richards Street Vancouver, BC V6B 2Z4 Canada</p> <p>Tel : +1 855.681.6017 (24x7) Fax : +1 302.213.0122 Email (sales) : sales@loadbalancer.org Email (support) : support@loadbalancer.org</p>
Europe (UK)	<p>Loadbalancer.org Ltd. Portsmouth Technopole Kingston Crescent Portsmouth PO2 8FA England, UK</p> <p>Tel : +44 (0)330 3801064 (24x7) Fax : +44 (0)870 4327672 Email (sales) : sales@loadbalancer.org Email (support) : support@loadbalancer.org</p>
Europe (Germany)	<p>Loadbalancer.org GmbH Alt Pempelfort 2 40211 Düsseldorf Germany</p> <p>Tel : +49 (0)30 920 383 6494 Fax : +49 (0)30 920 383 6495 Email (sales) : vertrieb@loadbalancer.org Email (support) : support@loadbalancer.org</p>