



Enterprise AWS Configuration Guide

Version 8.6.3 Revision 1.0.0



Table of Contents

1. Introduction	4
2. About Enterprise AWS	4
Main Differences to Our Standard (Non-Cloud) Product	4
Why use Enterprise AWS?	5
3. Accessing AWS	5
4. Deployment Concepts	5
Overview	5
AWS Topology Options	5
Single Availability Zone	6
Dual Availability Zones	7
Creating a VPC	8
VPC IP Address Types	8
Private	8
Public	8
Elastic (EIP)	9
AWS Integration	9
Secondary IP Addresses	9
Elastic IP Addresses	9
Autoscaling	9
IP address Allocation Options & Requirements	10
Internal (Private Network) Deployments	10
Public facing Deployments	10
VPC Network Interfaces (ENI)	11
10GB Support	12
Instance Type	12
5. Deploying Enterprise AWS	12
Create & Configure a VPC	12
Accessing & Deploying the AMI	14
Checking your Subscriptions	19
6. Accessing the Appliance	19
Using the WebUI	19
WebUI Menu Options	20
Appliance Security	21
Security Mode	21
Passwords	21
Checking For Updates	22
Appliance Licensing	22
Enterprise AWS Non-standard WebUI Menu Options	22
Accessing the Appliance using SSH	27
Using Linux	27
Using Windows	27
7. Configuration Examples	30
Deployment Notes	31
IP Addresses	31
Availability Zones	31
Routing Table Target Configuration	31
Real Server Internet access via the Load Balancer Instance	31
1 – Web Servers – 1 subnet, 1 load balancer network interface, layer7	31
2 – Web Servers – 2 subnets, 2 load balancer network interfaces, layer7	33

3 – Web Servers – 2 subnets, 1 load balancer network interface, layer7, transparent	36
4 – Web Servers – 1 subnet, 1 load balancer network interface, layer7, SSL termination	38
5 – RD Session Hosts – 2 subnets, 1 load balancer network interface, layer7	41
6 – Web Servers – 2 subnets, 1 load balancer network interface, layer4	43
8. Configuring High Availability using two Instances (Primary & Secondary)	46
9. Configuring High Availability using two Instances across Availability Zones	50
10. Testing – General Comments	54
Testing Load Balanced Services	54
Diagnosing VIP Connection Problems	54
Taking Real Servers Offline	55
Using Reports & Log Files	56
11. More Information	56
12. Loadbalancer.org Technical Support	56
13. Appendix	57
IAM Role Configuration	57
Configuring the load balancer to auto add/remove auto-scaled Real Servers	58
Configuring Auto-Scaling to auto deploy a new LB.org Instance on Failure	59
Configuring a VPC Endpoint to enable AWS API Access	61

1. Introduction

Amazon Web Services offers an extensive set of global cloud-based services. These services help organizations move faster, lower IT costs, and scale. It allows services to be deployed as and when required. Charges are made for what is used making it an extremely flexible and cost effective solution.

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

2. About Enterprise AWS

Enterprise AWS is a fully featured Application Delivery Controller (ADC) / Load Balancer designed specifically for AWS. The core software is based on customized versions of Centos 6.x/RHEL 6.x, Linux 4.9.x, LVS, HA-Linux, HAProxy, Pound, STunnel & Ldirectord.

Enterprise AWS can be deployed as a single instance or as an HA clustered pair of instances for high availability and resilience. For details of adding a second (Secondary) instance, please refer to [Deployment Concepts - Single Availability Zone](#) and [Configuring HA Using 2 Instances \(Primary & Secondary\)](#). It's also possible to deploy 2 instances in different AZs for high availability, this is achieved using a **Primary 1/Primary 2** model rather than the **Primary/Secondary** model, please refer to [Deployment Concepts - Dual Availability Zones](#) and [Configuring HA Using 2 Instances across Availability Zones](#) for more details.

Enterprise AWS is based on our 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 Our Standard (Non-Cloud) Product

1. The network setup is customized for Amazon EC2 deployment.
2. 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 [Configuration Example 3](#).
 - Also, for a clustered pair of load balancers (Primary & Secondary) the AWS routing table for the Real Server subnet must be dynamically changed when failover from the active to passive device occurs. This can be achieved using the WebUI option: *Cluster Configuration > Heartbeat Advanced*, and the AWS command `aws ec2 replace-route` as explained in [Configuring High Availability using two Instances \(Primary & Secondary\)](#).
3. 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 [Configuration Example 6](#).
 - Also, for a clustered pair of load balancers (Primary & Secondary) the AWS routing table for the Real Server subnet must be dynamically changed when failover from the active to passive device occurs. This can be achieved using the WebUI option: *Cluster Configuration > Heartbeat Advanced*, and the AWS command `aws ec2 replace-route` as explained in [Configuring High Availability using two Instances](#)

(Primary & Secondary).

4. Layer 4 DR mode is only supported for internal clients located in the same VPC as the load balancer. This can be useful for multi-tiered applications. Please refer to [this blog](#) for more information.

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.

Note

For a full feature comparison please refer to our [Enterprise AWS product page](#).

3. Accessing AWS

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: <https://aws.amazon.com/console/>.

4. Deployment Concepts

Overview

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

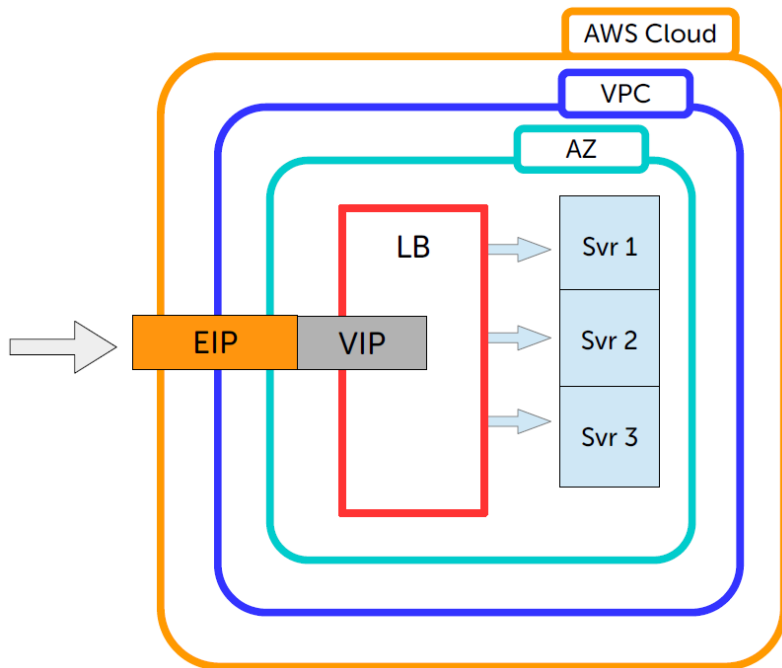
AWS Topology Options

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

Single Availability Zone

Single Appliance

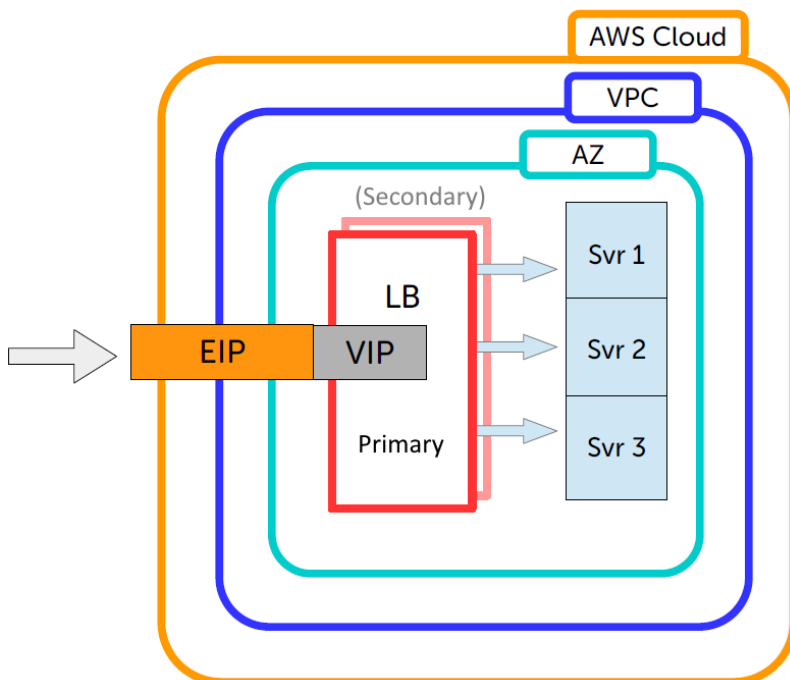
A single instance is deployed.



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

2 Appliances in Active/Passive mode

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



- Under normal conditions the Primary is active and the Secondary is passive. If the Primary fails, the load

balanced services (VIPs) will be automatically brought up on the Secondary. When failover occurs, the EIP is still associated with the same private IP address, but it's now active on the Secondary.

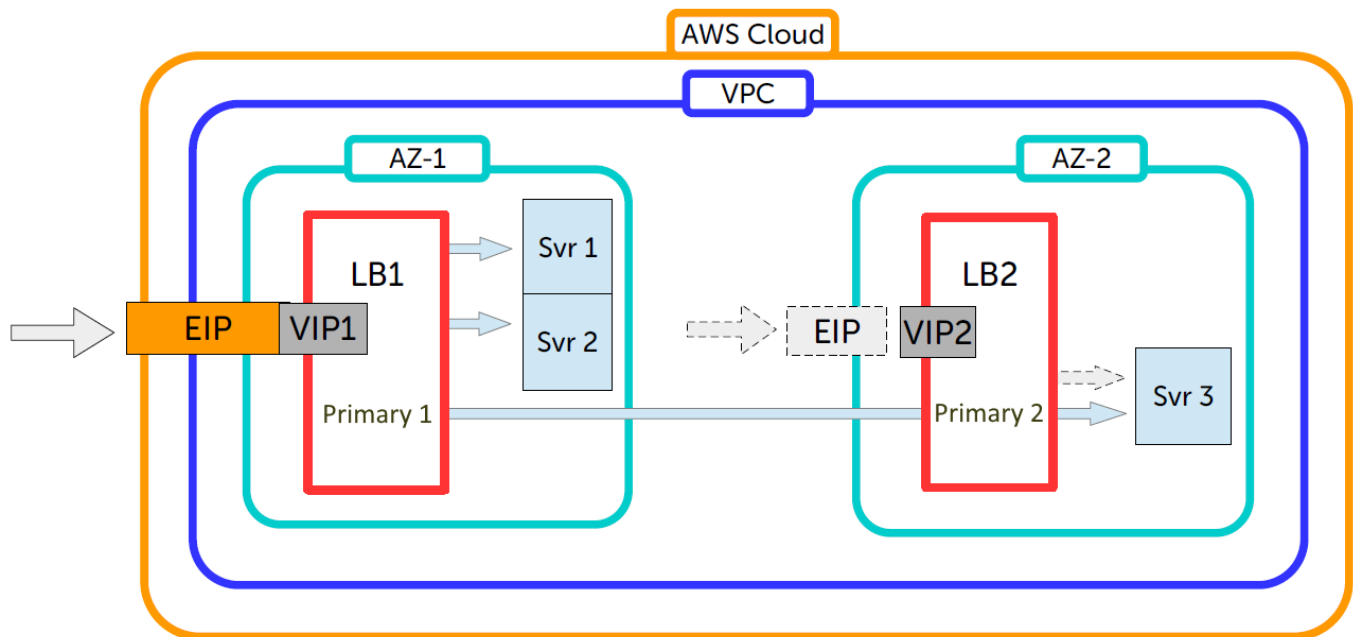
- For a correctly configured pair, changes made to load balanced services on the Primary will be automatically replicated to the Secondary.
- Both Primary and Secondary appliances must be deployed in the same subnet/Availability Zone to allow VIP(s) to be brought up on either appliance.
- Please refer to [Configuring High Availability using two Instances \(Primary & Secondary\)](#) for detailed steps on configuring this mode.

Dual Availability Zones

2 Instances in AZ HA Mode

This mode enables two load balancer instances to be configured in different subnets/Availability Zones. In this mode, for each load balanced service, a VIP is configured on each instance and both are always locally active, but only one is made available via the associated EIP. Regular checks ensure that the EIP is up, and if not, the EIP is automatically associated with the other instance thereby ensuring availability.

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



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

- Please refer to [Configuring HA Using 2 Instances across Availability Zones](#) for detailed steps on configuring this mode.

Creating a VPC

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

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

Note | For more information about VPCs please click [here](#).

VPC IP Address Types

There are 3 IP address types as detailed below:

Private

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

Public

Internet routable IP address assigned by the system for all instances. Traffic routed to the Public IP is translated via 1:1 Network Address Translation (NAT) and forwarded to the Private IP address of an instance. The mapping of a Public IP to Private IP of an instance is the default launch configuration for all instance types. Public IP Addresses

are released when instances are stopped or terminated. When an instance is powered on again or restarted, it is allocated a different public IP address. If you require a persistent public IP address that can be associated to and from instances as you require, use an Elastic IP address instead.

Elastic (EIP)

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

Note

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

AWS Integration

Enterprise AWS has been designed to leverage the AWS API to automatically complete tasks in AWS that would otherwise need to be done manually.

To achieve this, each instance must have access to the API. This can be provided by defining a VPC Endpoint to enable connectivity from the subnet where the instance is located. Please refer to [Configuring a VPC Endpoint to enable AWS API Access](#).

API calls are used to provide automatic integration in the following areas.

Secondary IP Addresses

When VIPs are added that are not bound to the Primary IP address of the instance, additional secondary private IP addresses must be added. These IP's must be added using the appliance's WebUI and also via the AWS portal to ensure that AWS is also aware of the new allocation.

If the appliance has access to the AWS API Endpoint when the VIP is created, secondary IP addresses are automatically assigned to the instance's Primary network interface when VIPs are created using the appliance WebUI. This is achieved through AWS API calls.

If the appliance does not have access to the AWS API when the VIP is created, the IP address must also be manually assigned to the network interface using the AWS EC2 dashboard.

Elastic IP Addresses

The same concept applies to EIPs. This relates to allocating and deleting EIPs and also association and disassociation.

Autoscaling

Enterprise AWS support integration with AWS Autoscaling. When configured, the load balancer will use AWS API calls to monitor the specified autoscaling group for any changes in servers. If a server is added or removed from the autoscaling group, the same will be automatically applied to the Virtual Service.

Layer 7 VIPs also support the ability to set the port used to connect to the backend server. This is set when the autoscaling service adds a new server. If empty this will default to the frontend listen port.

For details on how to enable Autoscaling for a particular VIP, please refer to [Configuring the load balancer to auto add/remove auto-scaled Real Servers](#).

IP address Allocation Options & Requirements

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

Internal (Private Network) Deployments

In this scenario **Virtual Services (VIPs)** can be configured in the following ways:

- **For a Single Appliance**

1. Using the primary private IP address of the instance.
2. Using additional secondary private IP(s).

AND

Configuring a **VPC Endpoint** to enable access from the subnet where the instance is located, to the Amazon API for your particular region, e.g. for **US-east**, the AWS Service Endpoint to access the API is <https://ec2.us-east-1.amazonaws.com>. When the Endpoint is configured, make sure that the associated Security Group allows inbound access on port 443.

Note

If the Endpoint is not configured there will be no access to the AWS API. In this case, the secondary IP address(es) will need to be added via the appliance WebUI and manually via the AWS portal.

- **For a Clustered Pair (Primary & Secondary)**

1. Using additional secondary private IP(s) - this ensures that the VIP is able to 'float' between the Primary & Secondary instance.

AND

Configuring a VPC Endpoint to enable access from the subnet where the instances are located, to the Amazon API for your particular region, e.g. for **US-east**, the AWS Service Endpoint to access the API is <https://ec2.us-east-1.amazonaws.com>. When the Endpoint is configured, make sure that the associated Security Group allows inbound access on port 443.

Note

Access to the AWS API **MUST** be provided for this method to be configured and to operate correctly. If there is no access, it will not be possible to move the floating IP address for the VIP from one instance to the other.

Public facing Deployments

In this scenario **Virtual Services (VIPs)** can be configured in the following ways:

- **For a Single Appliance**

1. Using the primary private IP address of the instance and associating an EIP.
2. Using additional secondary private IP(s) and associating EIP(s).

AND

Configuring a VPC Endpoint to enable access from the subnet where the instance is located, to the

Amazon API for your particular region, e.g. for US-east, the AWS Service Endpoint to access the API is <https://ec2.us-east-1.amazonaws.com>. When the Endpoint is configured, make sure that the associated Security Group allows inbound access on port 443.

Note

If the Endpoint is not configured:

- There will be no access to the AWS API. In this case, the secondary IP address(es) will need to be added via the appliance WebUI and manually via the AWS portal.
- EIP allocation & association must be configured via the AWS portal.

• For a Clustered Pair (Primary & Secondary)

1. Using additional secondary private IP(s) and associating EIP(s).

AND

Configuring a VPC Endpoint to enable access from the subnet where the instances are located, to the Amazon API for your particular region, e.g. for US-east, the AWS Service Endpoint to access the API is <https://ec2.us-east-1.amazonaws.com>. When the Endpoint is configured, make sure that the associated Security Group allows inbound access on port 443.

Note

Access to the AWS API **MUST** be provided for this method to be configured and to operate correctly. If there is no access, it will not be possible to move the floating IP address for the VIP from one appliance to the other or re-associate the EIP.

• For Dual AZ (Primary 1 & Primary 2)

1. Using additional secondary private IP(s) and associating EIP(s).

AND

Assigning a standard, dynamic public IP address to each instance when deployed, this ensures that each appliance is able to access the internet to perform the EIP health check of the peer appliance, more information on the options available to provide instance Internet access is available [here](#).

AND

Configuring a VPC Endpoint to enable access from the subnets where the instances are located, to the Amazon API for your particular region, e.g. for US-east the AWS Service Endpoint to access the API is <https://ec2.us-east-1.amazonaws.com>. When the Endpoint is configured, make sure that the associated Security Group allows inbound access on port 443.

Note

Access to the AWS API **MUST** be provided for this method to be configured and to operate correctly. If there is no access, it will not be possible to re-associate the EIP.

VPC Network Interfaces (ENI)

By default, a single ENI (Elastic Network Interface) is allocated when an instance is launched. A private IP address within the IP address range of its VPC is auto assigned to the ENI. Multiple private IP addresses can be assigned to

each ENI, the limit is determined by instance type as defined [here](#).

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 located 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 t3.medium. This can be changed as required. A comparison of the various types is available [here](#).

5. Deploying Enterprise AWS

Create & Configure a VPC

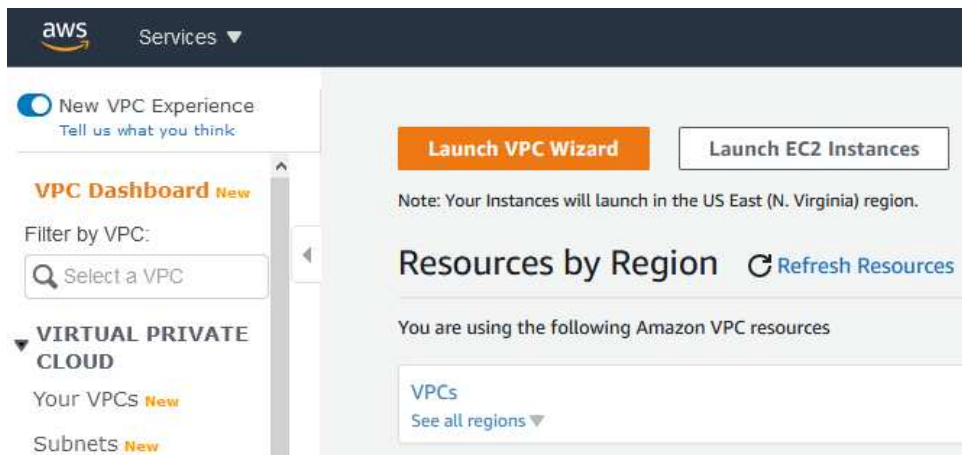
For a manually created VPC, the key steps are:

1. Create a **VPC** – this is an isolated portion of the AWS cloud.
2. Create and attach an **Internet gateway** – this connects the VPC directly to the Internet and provides access to other AWS products.
3. Create an Amazon **VPC subnet** – this is a segment of a VPC's IP address range where you can launch Amazon EC2 instances.
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 inbound and outbound traffic.

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

To create a VPC using the wizard:

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



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

Note

This wizard option is appropriate in many cases. It creates a VPC with a single public subnet and auto configures the gateway, subnets and routing table. Additional subnets can easily 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.

Important:

If you are using a Local Zone with your VPC [follow this link](#) to create your VPC.

Select

Public Subnet

Amazon Virtual Private Cloud

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

Step 2: VPC with a Single Public Subnet

IPv4 CIDR block:* (65531 IP addresses available)

IPv6 CIDR block: ☒ No IPv6 CIDR Block
☐ Amazon provided IPv6 CIDR block
☐ IPv6 CIDR block owned by me

VPC name:

Public subnet's IPv4 CIDR:* (251 IP addresses available)

Availability Zone:* ▼

Subnet name:

You can add more subnets after AWS creates the VPC.

Service endpoints

Enable DNS hostnames:* ☒ Yes ☐ No

Hardware tenancy:* ▼

Enable ClassicLink:* ☐ Yes ☒ No

4. Click **Create VPC**.

Note | For more information about VPCs please click [here](#).

Accessing & Deploying the AMI

To access and deploy the AMI:

1. In the EC2 Dashboard, click **Launch Instance**.
2. Select *AWS Marketplace*.
3. Search for "Loadbalancer.org".
4. Click **Select** next to the required AMI, either:
 - **Advanced Load Balancer ADC – 10G** (Fully featured appliance licensed for 20Gbps throughput)
 - **Advanced Load Balancer ADC – 1G** (Fully featured appliance licensed for 4Gbps throughput)
 - **Advanced Load Balancer ADC – BYOL** (Fully featured appliance, one time payment (AWS infrastructure costs still apply), purchase and apply either the 1G or the 10G license)

Note

The BYOL version will work completely unrestricted for 30 days without any license applied. During this period, only AWS usage charges will apply. After the 30 days, the trial will still function, but no configuration changes will be possible until the license is applied.

5. Review pricing details and if happy to proceed click **Continue**.
6. Select the required instance type – **t3.medium** is the default.

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instance needs.

Filter by: All instance families Current generation [Show/Hide Columns](#)

Currently selected: t3.medium (- ECUs, 2 vCPUs, 2.5 GHz, -, 4 GiB memory, EBS only)

Note: The vendor recommends using a **t3.medium** instance (or larger) for the best experience with

	Family	Type
<input checked="" type="checkbox"/>	t2	t2.nano
<input type="checkbox"/>	t2	t2.micro Free tier eligible
<input type="checkbox"/>	t2	t2.small
<input type="checkbox"/>	t2	t2.medium
<input checked="" type="checkbox"/>	t2	t2.large
<input checked="" type="checkbox"/>	t2	t2.xlarge
<input checked="" type="checkbox"/>	t2	t2.2xlarge
<input checked="" type="checkbox"/>	t3	t3.nano
<input checked="" type="checkbox"/>	t3	t3.micro
<input checked="" type="checkbox"/>	t3	t3.small
<input checked="" type="checkbox"/>	t3	t3.medium

7. 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 a

Number of instances ⓘ [Launch into Auto Scaling Group](#) ⓘ

Purchasing option ⓘ ☐ Request Spot instances

Network ⓘ ⓘ [Create new VPC](#)

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

Auto-assign Public IP ⓘ

Placement group ⓘ ☐ Add instance to placement group

Capacity Reservation ⓘ

Domain join directory ⓘ ⓘ [Create new directory](#)

IAM role ⓘ ⓘ [Create new IAM role](#)

CPU options ⓘ ☐ Specify CPU options

Shutdown behavior ⓘ

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

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

9. Select a suitable **IAM Role**. The role can simply have "**Amazon EC2 Full Access**" for the "**Amazon EC2**" AWS Service Role or for more granular configuration please refer [IAM Role Configuration](#).

Note

Configuring an **IAM role** for the instance is optional. However, we always recommend that one is assigned. This allows the instance to make AWS API calls to automatically configure the required AWS settings. If not set, these AWS settings would need to be manually configured.

10. Configure the remaining options according to your requirements

- **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 are located (same or different subnet as the load balancer) and the load balancing mode selected. Please refer to [Configuration Example 4](#) & [Configuration Example 7](#) for more details.
- **Advanced Details** – Configure according to your requirements.

11. 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.

Volume 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>	Throughput (MB/s) <small>i</small>	Delete on Termination <small>i</small>	Encryption <small>i</small>
Root	/dev/sda1	snap-082b2a0fbca3bcee1	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypt <small>v</small>

[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: Add Tags](#)

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

Step 5: Add Tags

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>
<p><i>This resource currently has no tags</i></p> <p>Choose the Add tag button or click to add a Name tag.</p> <p>Make sure your IAM policy includes permissions to create tags.</p>	

[Add Tag](#) (Up to 50 tags maximum)

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

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

Step 5: Add Tags

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	LB1

Add another tag (Up to 50 tags maximum)

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

14. 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: Advanced Load Balancer ADC - BYOL-8-5-AutogenByAWSMP-

Description: This security group was generated by AWS Marketplace and is based on recom

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	9443	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Custom UDP F	UDP	6694	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

[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](#)

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

- Management (SSH) – TCP port 22
- Management (WebUI) – TCP port 9443
- Heartbeat between Primary and Secondary appliances – UDP port 6694

Note

By default, rules with source of 0.0.0.0/0 allow all IP addresses access to the instance, these should be locked down to allow access only from known / trusted IPs.

- Additional rules must be added to provide access to the application(s) being load balanced. These should also be locked down to know IPs / IP ranges where possible.
 - For example, if you're load balancing HTTP & HTTPS traffic, add TCP ports 80 & 443

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

15. Click **Review and Launch**.

16. Check all settings and click **Launch**.

Select an existing key pair or create a new key pair
X

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

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

Note

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

18. If using an existing key pair, check (tick) the acknowledgment check-box.

19. Click the **Launch Instances** button, the instance will now launch.

20. If you're deploying layer 4 NAT mode services or Layer 7 with TProxy enabled, you'll need to disable the **Source/Destination Check** for the instance. This is because the instance must be able to send and receive traffic when the source or destination is not itself.

This can be done by right-clicking the instance and selecting: *Networking > Change Source/Dest. Check* and ensuring it's disabled (*Stop* should be checked) as shown below:

Source / destination check [Info](#)

Each EC2 instance performs source and destination checks by default. The instance must be the source or destination of all the traffic it sends and receives.

Instance ID


 i-00cd455a64a24330b (LB1)

Network interface [Info](#)

 eni-04d6e7d6a8180bd15 (LB1)

Source / destination checking [Info](#)

☒ Stop


 If this is a NAT instance, you must stop source / destination checking. A NAT instance must be able to send and receive traffic when the source or destination is not itself.


Checking your Subscriptions

Current subscriptions can be managed using the *Your Marketplace Software* option in the [AWS Marketplace](#) console as shown below:

Manage subscriptions [Info](#)


Actions ▼


 **You will need a License Manager SLR to see license entitlements**
Without AWS License Manager service linked roles (SLRs) you will not be able to see any of your AWS Marketplace license entitlements below. Please go to AWS License Manager Console to onboard with the AWS License Manager SLRs.

Set up SLR 

Your subscriptions

All delivery methods ▼

 < 1 > 



Advanced Load Balancer ADC - BYOL

By Loadbalancer.org

Amazon Machine Image Support eligible BYOL

Access level

Agreement

Launch new instance

Manage

6. Accessing the Appliance

Using the WebUI

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

<https://<Public DNS name>:9443>

or

<https://<Public IP address>:9443>

© Copyright Loadbalancer.org • Documentation • Enterprise AWS Configuration Guide

19

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

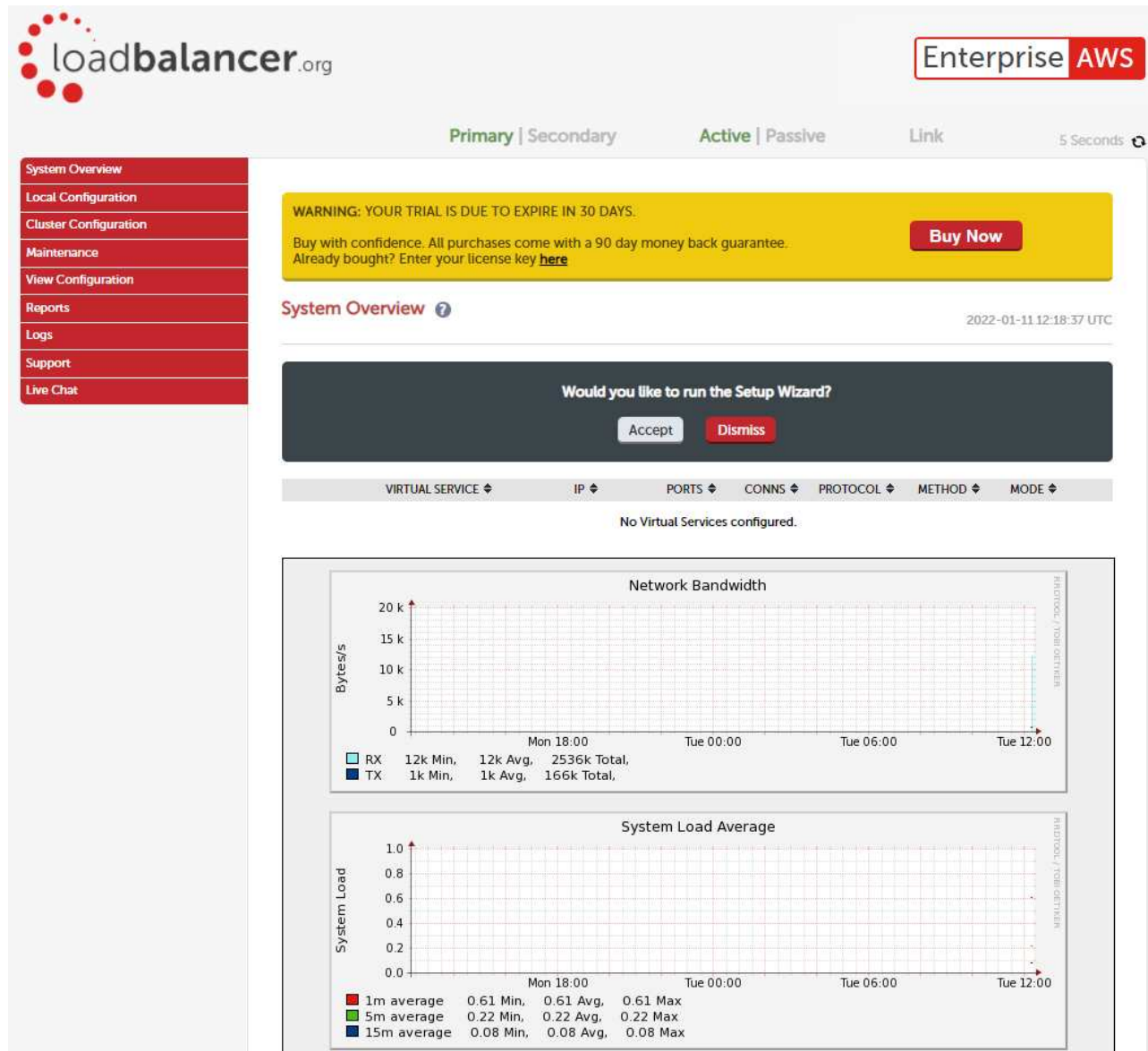
Username: loadbalancer

Password: <EC2 Instance-ID>

Note

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

Once logged in, the WebUI is displayed:



WebUI Menu Options

System Overview - Displays a graphical summary of all VIPs, RIPs and key appliance statistics

Local Configuration - Configure local host settings such as IP address, DNS, system time etc.

Cluster Configuration - Configure load balanced services such as VIPs & RIPs

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

View Configuration - Display the saved appliance configuration settings

Reports - View various appliance reports & graphs

Logs - View various appliance logs

Support - Create a support download, contact the support team & access useful links

Live Chat - Start a Live Chat session with one of our Support Engineers

Appliance Security

Note

For full details of all security related features, please refer to [Appliance Security Features](#).

Security Mode

To control how the appliance is accessed and which features are enabled, 3 security modes are provided:

- **Secure - (default)** - in this mode:
 - the WebUI is accessible on HTTPS port **9443**. If you attempt to access the WebUI on HTTP port **9080** you will be redirected to HTTPS port **9443**
 - access to the *Local Configuration > Execute shell command* menu option is disabled
 - the ability to edit the firewall script & the lockdown wizard is disabled
 - 'root' user console & SSH password access are disabled
- **Custom** - in this mode, the security options can be configured to suit your requirements
- **Secure - Permanent** - this mode is the same as **Secure** but the change is *irreversible*

Important

Only set the security mode to **Secure - Permanent** if you are 100% sure this is what you want!

To configure the Security Mode:

1. Using the WebUI, navigate to: *Local Configuration > Security*.
2. Select the required *Appliance Security Mode* - if **Custom** is selected, configure the additional options according to your requirements.
3. Configure the *HTTPS Port for Web User Interface*, *Web Interface SSL Certificate* and *Ciphers to use* according to your requirements.
4. Click **Update**.

Passwords

The password for the 'loadbalancer' WebUI user account and the 'root' Linux user account are set during the Network Setup Wizard. These can be changed at any time.

1 - the 'root' Linux account

As explained above, 'root' user console & SSH password access are disabled by default. If enabled, the 'root' password can be changed at the console, or via an SSH session using the following command:

```
# passwd
```

Note

For the AWS and Azure cloud products it's not possible to directly login as root. If root access is required, once you've logged into the console/SSH session using the credentials defined during instance deployment, run the following command:

```
$ sudo su
```

2 - the 'loadbalancer' WebUI account

This can be changed using the WebUI menu option: *Maintenance > Passwords*.

Checking For Updates

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

Appliance Licensing

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

Enterprise AWS Non-standard WebUI Menu Options


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

The menu options that are different are detailed below. For all others please refer to our [Administration Manual](#).

1) Local Configuration > Network Interface Configuration

Network Interface Configuration

IP Address Assignment


eth0

eth0

10.0.0.102/24

10.0.0.202/24

MTU

1500

 bytes

Configure Interfaces

Notes

1. Shows the private IP addresses allocated to the instance.
2. The first address in the list is auto-allocated when launched - it's not possible to change the auto-allocated IP address.
3. Multiple IP addresses can be assigned as shown.
4. Additional IP addresses added here after the first one in the list are shown as "Secondary Private IPs" in the AWS/EC2 Dashboard.
5. Click **Configure Interfaces** to apply any changes.

2) 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

1. Enables commands to be run at failover from Primary to Secondary appliance if configured. This includes Amazon CLI tools commands. For more information of the various CLI commands available please click [here](#).
2. Please refer to [Configuring HA Using 2 Instances \(Primary & Secondary\)](#) for more details on configuring 2 appliances in a Primary/Secondary HA configuration.

3) EC2 Configuration > EC2 Network Configuration

Associated Elastic IP's ?

Elastic IP		Private IP	Use with AZ HA	
52.211.158.247	→	10.0.0.160	<input checked="" type="checkbox"/>	[Disassociate]

Available Elastic IP's

52.209.141.104	eipalloc-6de42109	[Delete]
----------------	-------------------	------------

[Allocate New Elastic IP ?](#)

Notes

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

4) EC2 Configuration > EC2 Zone HA Configuration

This menu option is used to configure Zone HA , where 2 Primary instances are deployed, each in a different AZ in a Primary 1/Primary 2 configuration. Please refer to [Configuring HA Using 2 Instances across Availability Zones](#) for more details on configuring dual AZ HA mode

Synchronization Tab

SYNCHRONISE WITH PEER ?

Generate a new TLS key pair and copy to peer

IP address of peer in another Availability Zone

Password for *loadbalancer* user on peer

Add new node

Notes

- This is used to configure a Primary/Secondary pair. The IP address of the Secondary instance and the password for the *loadbalancer* user must be entered, then when **Add new node** is clicked, new keys and signed certificates will be generated and synchronized with the node specified. These keys are used to verify the peer when monitoring an Elastic IP across Availability Zones.
- Please refer to [Configuring HA Using 2 Instances across Availability Zones](#) for more details on configuring dual AZ HA mode.

Security Tab

Root key installed (Delete)
Root certificate installed (Delete)
Server certificate installed (Delete)
Server key installed (Delete)

Notes

- This is used to verify that the various keys & certificates have been generated and also allows them to be deleted.
- If deleted, the keys & certificates will need to be re-generated using the Synchronization Tab as described above.
- Please refer to [Configuring HA Using 2 Instances across Availability Zones](#) for more details on configuring dual AZ HA mode.

Configuration Tab

Synchronisation
Security
Configuration

CONFIGURATION

Listen port

9444

?

Check Interval

5

?

Failure Count

3

?

Max Association Retry

10

?

Update

Notes

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

5) EC2 Configuration > EC2 Zone HA Status

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

The Primary 1 instance:

AZ HA STATUS

Elastic IP	Private IP	Status	Action
52.211.158.247	10.0.0.160	Local	[Disassociate]

The Primary 2 instance:

AZ HA STATUS

Elastic IP	Private IP	Status	Action
52.211.158.247	10.0.1.160	Peer	[Associate]

Notes

1. The VIP (10.0.0.160) on the Primary 1 instance is currently associated with the EIP.
2. Please refer to [Configuring HA Using 2 Instances across Availability Zones](#) for more details on configuring dual AZ HA mode.

Accessing the Appliance using SSH

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

Note

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

Using Linux

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

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

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

e.g.

Using the IP address:

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

Or using the fqdn:

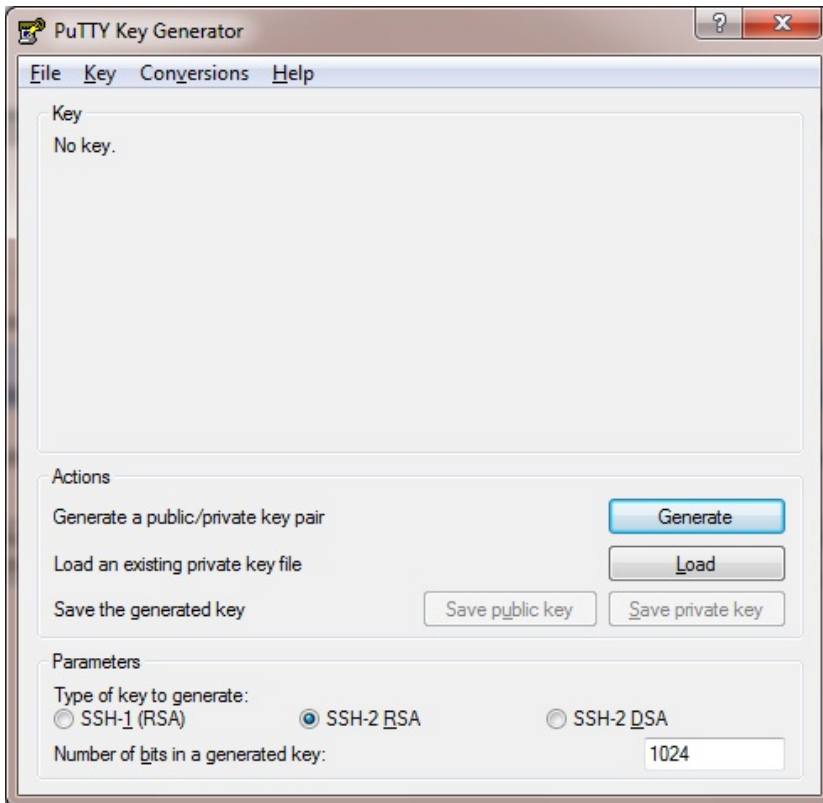
```
# ssh -i /path-where-saved/private-key-file.pem lbuser@fqdn
```

Using Windows

Note

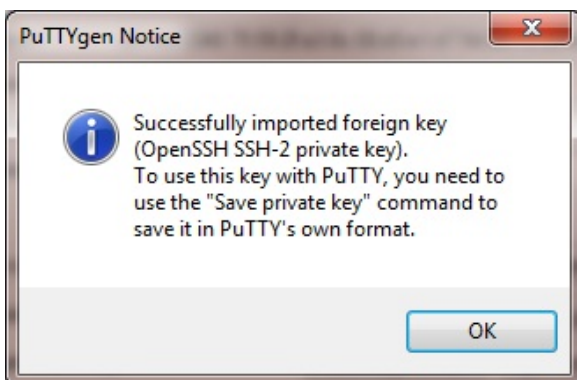
If you create the keypair before deploying the load balancer instance using the *Network & Security > Key Pair* option in the EC2 console, the output file format to be set to ppk. In this case Puttygen is not needed.

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

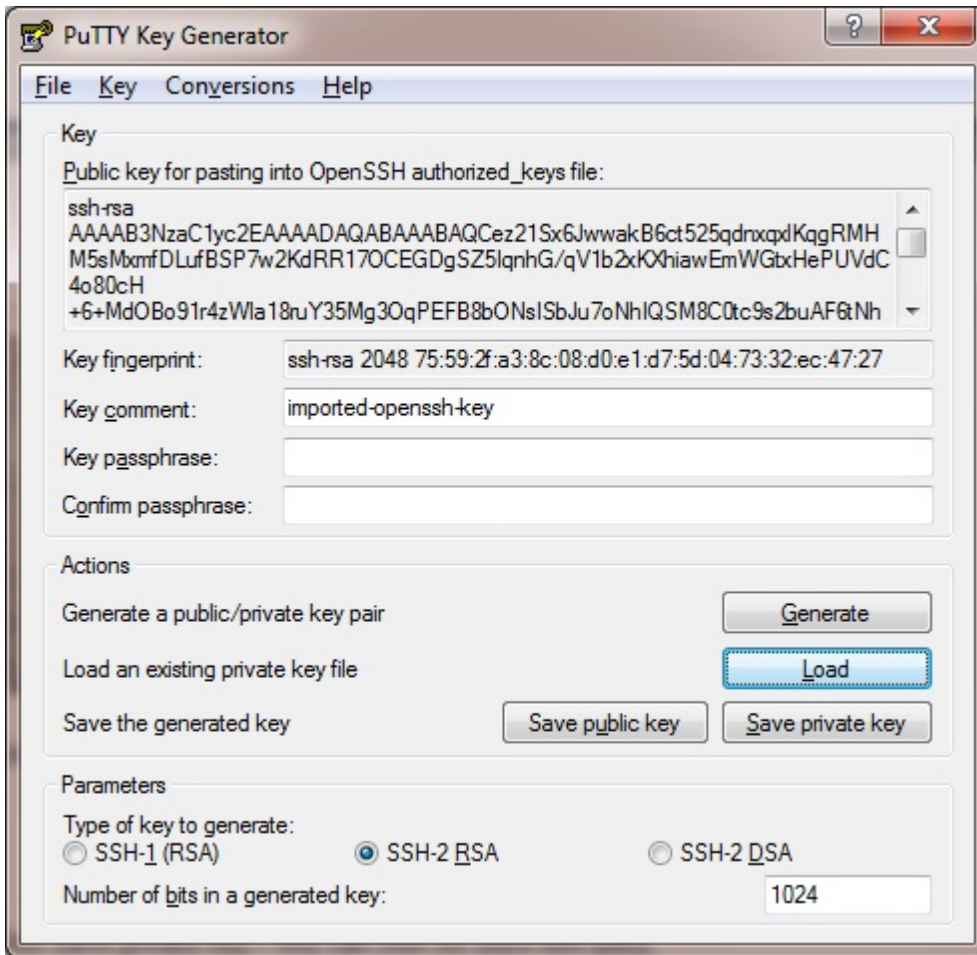


Click **Load**, change the file-type to all files and select the pem file saved earlier when creating your Key Pair.

You should see the following message:

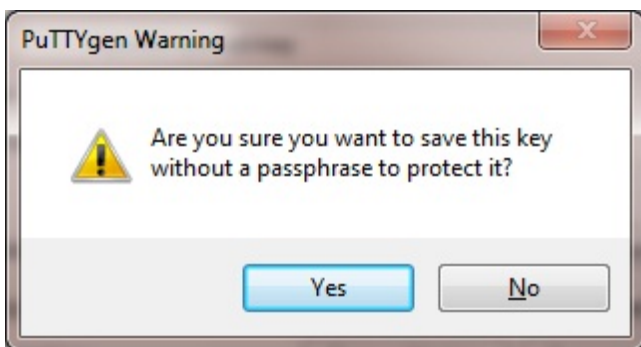


Click **OK**.



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

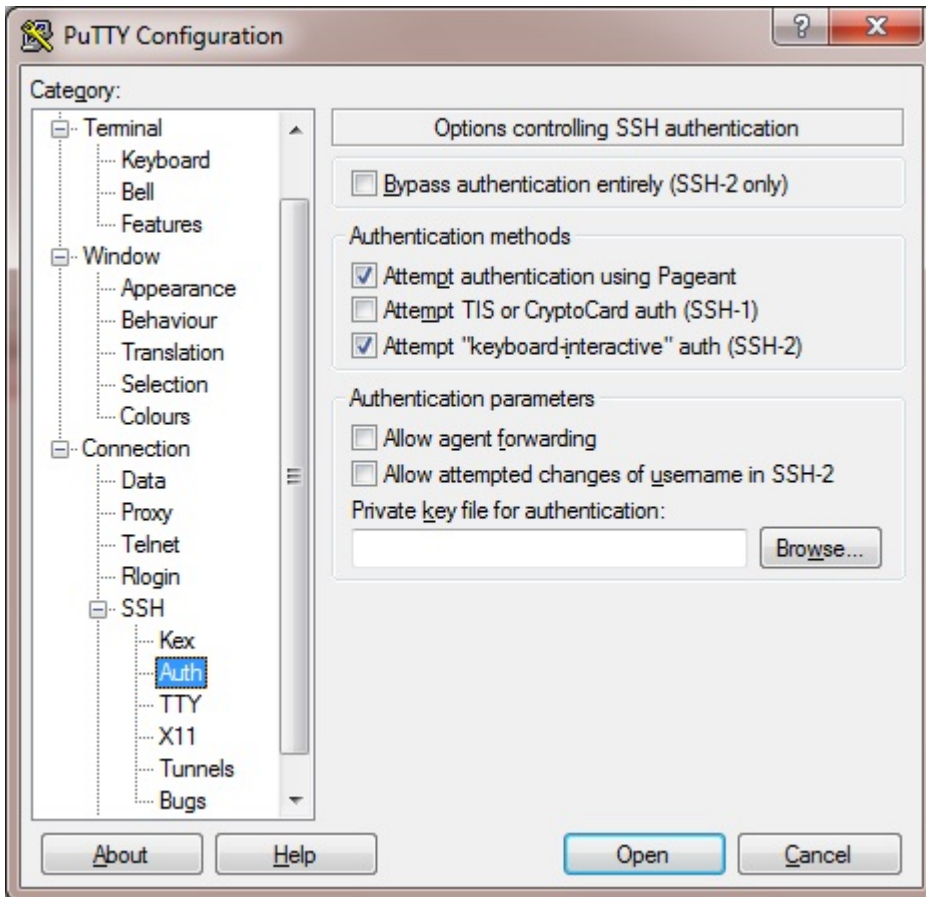
You can also choose to enter an additional pass-phrase for improved security, if you don't, the following message will be displayed:



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

Now close PuTTYgen and start PuTTY.

Expand the SSH section as shown below:



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

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

Note

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

```
$ sudo su
```

7. Configuration Examples

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

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

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

Deployment Notes

IP Addresses

Various conditions must be met depending on the deployment type, please refer to [IP address Allocation Options & Requirements](#) for more details.

Availability Zones

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

It's also possible to place one load balancer instance in AZ-1 and a second instance in AZ-2, then create a Primary 1/Primary 2 HA pair. Please refer to [Deployment Concepts - Dual Availibty Zones](#) and [Configuring HA Using 2 Instances across Availability Zones](#) for more information.

Routing Table Target Configuration

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

Real Server Internet access via the Load Balancer Instance

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

To enable *autonat*:

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



1 – Web Servers – 1 subnet, 1 load balancer network interface, layer7

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

a) Setting up AWS

1. Deploy the load balancer instance as described in [Deploying Enterprise AWS](#).
2. Deploy your required web server instances into the same VPC & subnet as the load balancer.

b) Setting up the Virtual Service

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

2. Enter the following details:

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

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

c) Setting up the Real Servers

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

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"/>	?
Re-Encrypt to Backend	<input type="checkbox"/>	?
Weight	<input type="text" value="100"/>	?

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

d) Applying the new Layer 7 Settings

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

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

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

Associated Elastic IP's

Elastic IP		Private IP	Use with AZ HA	
3.215.68.164 ▼	→	10.0.0.22 ▼	<input type="checkbox"/>	[Associate]

Available Elastic IP's

3.215.68.164	eipalloc-00d5d9cacb8749c81	[Delete]
54.166.28.242	eipalloc-068837fd23e0883fe	[Delete]

Allocate New Elastic IP 

2. Under the Associated Elastic IP's section click [Associate] next to the VIP's 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 – Web Servers – 2 subnets, 2 load balancer network interfaces, layer7

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


1. Deploy the load balancer instance as described in [Deploying Enterprise AWS](#).
2. Add a second subnet to your VPC, skip this step if you already have one.
3. Add a second Network Interface, associate it with the second subnet and attach it to the load balancer instance.
4. Deploy your required web server instances into the second subnet.

b) Configuring the second Network Interface

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

IP Address Assignment

eth0

eth1

eth0

10.0.0.220/24

eth1

10.0.2.220/24

Configure Interfaces

2. Click **Configure Interfaces**.

c) Setting up the Virtual Service

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

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

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

d) Setting up the Real Servers

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

2. Enter the following details:

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"/>	?
Re-Encrypt to Backend	<input type="checkbox"/>	?
Weight	<input type="text" value="100"/>	?

3. Enter an appropriate label for the RIP, e.g. **Web1**.

4. Change the *Real Server IP Address* field to the required IP address, e.g. **10.0.2.50**.

5. Set the *Real Server Port* field to the required port, e.g. **80**.

6. Click **Update**.

7. Repeat the above steps to add your other web server(s).

e) Applying the new Layer 7 Settings

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

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

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

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]

?

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

Note

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 [Configuration Example 4](#) instead if you require layer 7 with transparency.

3 – Web Servers – 2 subnets, 1 load balancer network interface, layer7, transparent

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

a) Setting up AWS

1. Deploy the load balancer instance as described in [Deploying Enterprise AWS](#).
2. Disable the Source/Destination Check for the load balancer instance. This is required to ensure that the load balancer is available as a target when configuring routing (see step 5 below). This is done by right-clicking the instance and selecting: *Networking > Change Source/Dest. Check* and ensuring it's disabled.
3. Add a second subnet to your VPC, skip this step if you already have one.
4. Deploy your required web server instances into the second subnet.
5. Add a default route to the second subnets routing table (the subnet where the web servers are located), set the target to be the interface on the load balance.
 - Under the VPC dashboard, select *Route Tables*.
 - Select the route table that relates to the second subnet.
 - Select the *Routes* tab, and click **Edit routes**.
 - Click **Add 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-06e68cfb73e850255 LB1" as shown below:

[Route Tables](#) > Edit routes

Edit routes

Destination	Target
10.0.0.0/16	local
<input type="text" value="0.0.0.0/0"/>	<input type="text" value="i-06e68cfb73e850255"/>
<input type="button" value="Add route"/>	<div>i-06e68cfb73e850255 LB1</div>

Important

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

- Click **Save Routes**

b) Setting up the Virtual Service

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 7 – Virtual Services* and click **Add a New Virtual Service**.
2. 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		
Layer 7 Protocol	<input type="text" value="HTTP Mode"/>	?
Manual Configuration	<input type="checkbox"/>	?
		<input type="button" value="Cancel"/> <input type="button" value="Update"/>

3. Enter an appropriate label for the VIP, e.g. **Web-Cluster1**.
4. Set the *Virtual Service IP address* field to the required IP address, e.g. **10.0.0.22**.
5. Set the *Virtual Service Ports* field to the required IP address, e.g. **80**.
6. Leave *Layer 7 Protocol* set to **HTTP Mode**.
7. Click **Update**.
8. Now click **Modify** next to the newly created VIP.
9. Scroll down to the *Other* section and click **[Advanced]**.
10. Enable (check) *Transparent Proxy*.
11. Click **Update**.

c) Setting up the Real Servers

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

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"/>	?
Re-Encrypt to Backend	<input type="checkbox"/>	?
Weight	<input type="text" value="100"/>	?
		<input type="button" value="Cancel"/> <input type="button" value="Update"/>

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

d) Applying the new Layer 7 Settings

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

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

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

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](#) ?

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

4 – Web Servers – 1 subnet, 1 load balancer network interface, layer7, SSL termination

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

Note

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

a) Setting up AWS

1. Deploy the load balancer instance as described in [Deploying Enterprise AWS](#).
2. Deploy your required web server instances into the same VPC & subnet as the load balancer.

b) Setting up the Virtual Service

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 7 – Virtual Services* and click **Add a New Virtual Service**.
2. 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		
Layer 7 Protocol	<input type="text" value="HTTP Mode"/>	?
Manual Configuration	<input type="checkbox"/>	?
		<input type="button" value="Cancel"/> <input type="button" value="Update"/>

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

c) Setting up the Real Servers

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

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"/>	?
Re-Encrypt to Backend	<input type="checkbox"/>	?
Weight	<input type="text" value="100"/>	?
		<input type="button" value="Cancel"/> <input type="button" value="Update"/>

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

7. Repeat the above steps to add your other web server(s).

d) Upload an SSL Certificate

1. Using the WebUI, navigate to *SSL Termination* and click **Add a new SSL Certificate**.
2. Select *Upload prepared PEM/PFX file*.
3. Enter an appropriate *Label* (name) for the certificate, e.g. **Cert1**.
4. Browse to and select the relevant certificate file.
5. for PFX files, enter the *PFX File Password*.
6. Click **Add Certificate**.

Note

You can also create a CSR on the load balancer. If this is required, select the *Create A New SSL Certificate (CSR)* option instead of *Upload prepared PEM/PFX file* in step 2 above. For additional information please refer to [Generating a CSR on the Load Balancer](#).

e) Configuring SSL Termination

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

Label	<input type="text" value="SSL-Web-Cluster1"/>	?
Associated Virtual Service	<input type="text" value="Web-Cluster1"/>	?
Virtual Service Port	<input type="text" value="443"/>	?
SSL Operation Mode	<input type="text" value="High Security"/>	
SSL Certificate	<input type="text" value="Cert1"/>	?
Source IP Address	<input type="text"/>	?
Enable Proxy Protocol	<input checked="" type="checkbox"/>	?
Bind Proxy Protocol to L7 VIP	<input type="text" value="Web-Cluster1"/>	?

2. Set the *Associated Virtual Service* drop-down to the VIP created in step (b) above (the *Label* field will be auto-populated).
3. Leave *SSL Operation Mode* set to **High Security**.
4. Select the *SSL Certificate* uploaded in step (d) above.
5. Click **Update**.

f) Applying the new Settings

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

2. Once the configuration is complete, use the **Restart STunnel** button at the top of the screen to apply the changes.

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

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

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 ?

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

5 – RD Session Hosts – 2 subnets, 1 load balancer network interface, layer7

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

a) Setting up AWS

1. Deploy the load balancer instance as described in [Deploying Enterprise AWS](#).
2. Add a second subnet to your VPC, skip this step if you already have one.
3. Deploy your required session host server instances into the second subnet.

b) Setting up the Virtual Service

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

Label	<input type="text" value="SessionHost-Cluster1"/>	?
Virtual Service		
IP Address	<input type="text" value="10.0.0.25"/>	?
Ports	<input type="text" value="3389"/>	?
Protocol		
Layer 7 Protocol	<input type="text" value="TCP Mode"/>	?
Manual Configuration	<input type="checkbox"/>	?
		<input type="button" value="Cancel"/> <input type="button" value="Update"/>

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

c) Setting up the Real Servers

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

Label	<input type="text" value="SessionHost1"/>	?
Real Server IP Address	<input type="text" value="10.0.2.50"/>	?
Real Server Port	<input type="text" value="3389"/>	?
Re-Encrypt to Backend	<input type="checkbox"/>	?
Weight	<input type="text" value="100"/>	?
		<input type="button" value="Cancel"/> <input type="button" value="Update"/>

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

d) Applying the new Layer 7 Settings

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

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

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

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 ?](#)

2. Under the Associated Elastic IP's section click **[Associate]** next to the VIP's 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 – Web Servers – 2 subnets, 1 load balancer network interface, layer4

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

a) Setting up AWS

1. Deploy the load balancer instance as described in [Deploying Enterprise AWS](#)
2. Disable Source/Destination Check For the load balancer instance. This is required for layer 4 NAT mode services. This is because the instance must be able to send and receive traffic when the source or destination is not itself. This can be done by right-clicking the instance and selecting: *Networking > Change Source/Dest. Check* and ensuring it's disabled.
3. Add a second subnet to your VPC, skip this step if you already have one.
4. Deploy your required web server instances into the second subnet.
5. 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 routes**.

- Click **Add 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-06e68cfb73e850255 LB1" as shown below:

[Route Tables](#) > Edit routes

Edit routes

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	i-06e68cfb73e850255
Add route	i-06e68cfb73e850255 LB1

Important

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

- Click **Save Routes**.

b) Setting up the Virtual Service

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

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

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

c) Setting up the Real Servers

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

Label	<input type="text" value="Web1"/>	?
Real Server IP Address	<input type="text" value="10.0.1.20"/>	?
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"/>	?

3. Enter an appropriate label for the RIP, e.g. **Web1**.
4. Change the *Real Server IP Address* field to the required IP address, e.g. **10.0.1.20**.
5. Set *Real Server Port* to **80**.
6. Click **Update**.
7. Repeat the above steps to add your other web servers(s).

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

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

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]

?

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

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

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

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



The screenshot shows a configuration interface with two rows. The first row has a label 'Auto-NAT', a dropdown menu showing 'eth0', and a help icon. The second row has a label 'Multi-threaded', a checkbox that is checked, and a help icon. A green 'Update' button is located at the bottom right of the configuration area.

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

8. Configuring High Availability using two Instances (Primary & Secondary)

Enterprise AWS supports HA mode using two instances configured as a clustered pair. In this mode, one device is active (typically the Primary appliance) and the other is passive (typically the Secondary appliance). If the active device fails for any reason, the passive device will take over. The Primary and Secondary appliances are deployed in the *same subnet/Availability Zone* to allow VIP(s) to be brought up on either appliance.

Note

Various conditions must be met depending on the deployment type, please refer to [IP address Allocation Options & Requirements](#) for more details.

Note

This procedure assumes the first appliance is already up and running and that it will be the Primary of the clustered pair.

Step 1 – Configure a VPC Endpoint to enable Amazon API access from both appliances

1. Please refer to [Configuring a VPC Endpoint to enable AWS API Access](#) for details on configuring an Endpoint.

Step 2 – Deploy a second Instance & Configure the Source/Dest. Check

1. Please refer to the steps in [Deploying Enterprise AWS](#).
2. If you have layer 4 NAT mode services or Layer 7 services with TProxy enabled, you must disable the **Source/Destination Check** for the instance. Right-click the instance and select: *Networking > Change Source/Dest. Check* and ensure it's disabled (*Stop* should be checked).

Step 3 – Prepare both instances for pairing

1. Using the WebUI, navigate to: *Local Configuration > Security* and change the *Appliance Security Mode* to **Custom** and ensure that *Disable Console Access* and *Disable SSH Password Access* are both un-checked on **both** appliances.
2. Now navigate to: *Local configuration > Execute Shell Command* (you may need to refresh the WebUI to see

the option in the menu) and run the following command on **both** appliances:

```
lb_enable_root enable
```

Step 4 – Update Security Group Settings

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

Rule 1:

```
Type: Custom Inbound UDP rule
Protocol: UDP
Port Range: 6694
Source: local subnet
```

Rule 2:

```
Type: Custom Inbound ICMP rule
Protocol: Echo Request
Port Range: N/A
Source: local subnet
```

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

Step 5 – 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 Secondary device, these routing rules must then be changed to route via the Secondary instance. To set this up:

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

```
aws ec2 replace-route --route-table-id rtb-01e14bef4c71fa663 --destination-cidr-block
0.0.0.0/0 --instance-id i-0ec274bb1ed990132 --region us-east-1
```

Note

change **rtb-01e14bef4c71fa663** to the RT-ID of the table associated with your real servers subnet.

change **i-0ec274bb1ed990132** to the Instance-Id of your Primary instance.

change **us-east-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 Primary instance. It's run automatically each time the Primary becomes active.

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

```
aws ec2 replace-route --route-table-id rtb-01e14bef4c71fa663 --destination-cidr-block 0.0.0.0/0 --instance-id i-0ca0bbe09f770161a --region us-east-1
```

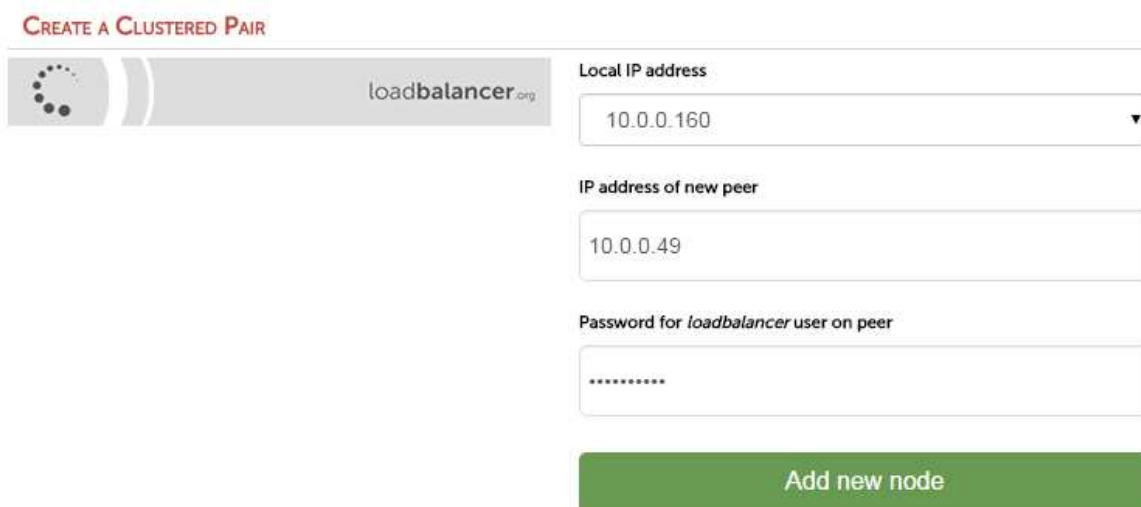
Note

- change **rtb-01e14bef4c71fa663** to the RT-ID of the table associated with your real servers subnet
- change **i-0ca0bbe09f770161a** to the Instance-Id of your Secondary instance
- change **us-east-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 Secondary instance. It's run automatically each time the Secondary becomes active

Step 6 – Configure High-Availability

1. Open the WebUI on the Primary unit.
2. Select the menu option: *Cluster Configuration > High Availability Configuration*.



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

Commit changes

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

Reload HAProxy

Restart Heartbeat

High Availability Configuration - primary

<div> <div>P</div> <div>10.0.0.160</div> <div>loadbalancer.org</div> </div>	<div>Break Clustered Pair</div>
<div> <div>S</div> <div>10.0.0.49</div> <div>loadbalancer.org</div> </div>	

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

Step 7 – Verify Synchronization State

- Once all services have restarted, the synchronization process will be complete.
- Verify that the status on the Primary & Secondary is as follows:

Primary Unit:

Primary | Secondary

Active | Passive

Link

Secondary Unit:

Primary | **Secondary**

Active | **Passive**

Link

Note | If no services have been configured, 'Active' will be grayed out on both instances.

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

SYSTEM OVERVIEW ?

2017-12-13 13:27:40 UTC

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

[Advanced]

Note | For other possible states, please refer to [Clustered Pair Diagnostics](#).

Step 8 – For public facing deployments Associate an EIP with the VIP on the Active Appliance

If this is a public facing deployment, on the active appliance allocate a new Elastic IP (if required) and associate the VIP address with the EIP.

9. Configuring High Availability using two Instances across Availability Zones

Enterprise AWS also supports HA mode using two instances deployed in different AZs. In this mode, VIPs are configured on both instances and are always locally active, but only one is made available via the associated EIP. For more information on how this mode works, please refer to [Deployment Concepts - Dual Availability Zones](#).

Note

Various conditions must be met depending on the deployment type, please refer to [IP address Allocation Options & Requirements](#) for more details.

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

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

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

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

1. Now deploy 2 instances – one in subnet 1, the other in subnet 2 and associate an EIP with each instance. For more information on deploying instances, refer to the steps in [Deploying Enterprise AWS](#).
2. Right-click each instance and select: *Networking > Change Source/Dest. Check* and ensure this is disabled.

Step 3 – Update Security Group Settings

1. Ensure that the security group used by each instance has the following rule defined - this is required to allow HA to be configured:

Type: Custom TCP rule
Protocol: TCP
Port Range: 9443
Source: Anywhere (or lockdown further if preferred, e.g. to the IP address/subnet of the peer appliance)

2. Ensure that the security group used by each instance has the following rule defined - this is required to allow the Zone HA check service to contact the peer node:

Type: Custom TCP rule
Protocol: TCP
Port Range: 9444
Source: Anywhere (or lockdown further if preferred, e.g. to the IP address/subnet of the peer appliance)

Step 4 – Configure Zone HA settings to enable the 2 instances to Communicate

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

The screenshot shows the 'Synchronisation' tab of the 'EC2 Zone HA Configuration' page. At the top, there are three tabs: 'Synchronisation' (active), 'Security', and 'Configuration'. Below the tabs, there is a red header 'SYNCHRONISE WITH PEER' with a help icon. The main content area contains the following elements: a text label 'Generate a new TLS key pair and copy to peer', a text label 'IP address of peer in another Availability Zone' followed by a text input field containing '52.53.54.55', a text label 'Password for loadbalancer user on peer' followed by a password input field with masked characters '.....', and a large green button at the bottom labeled 'Add new node'.

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

Step 5 – Configure Zone HA Settings

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

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

The screenshot shows the 'Configuration' tab of the 'EC2 Zone HA Configuration' page. At the top, there are three tabs: 'Synchronisation', 'Security', and 'Configuration' (active). Below the tabs, there is a red header 'CONFIGURATION'. The main content area contains a table with four rows of configuration parameters. Each row has a label, a text input field with a value, and a help icon. The parameters are: 'Listen port' with value '9444', 'Check Interval' with value '5', 'Failure Count' with value '3', and 'Max Association Retry' with value '10'. At the bottom right of the table, there is a green button labeled 'Update'.

Listen port	9444	?
Check Interval	5	?
Failure Count	3	?
Max Association Retry	10	?

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

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

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

Step 7 – Configure Failover Scripts

1. On the *first* instance, edit the file `/etc/loadbalancer.org/scripts/azhaFailover` and add any commands you would like to run (e.g. route customization) when the *first* instance becomes live.
2. On the *second* instance, edit the file `/etc/loadbalancer.org/scripts/azhaFailover` and add any commands you would like to run (e.g. route customization) when the *second* instance becomes live.

Note

Please refer to [Configuring HA Using 2 Instances \(Primary & Secondary\)](#) for an example of how to use the "aws ec2 replace-route" command.

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

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

Note

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

Elastic IP		Private IP	Use with AZ HA	
52.18.181.235 ▾	→	10.0.0.160 ▾	<input checked="" type="checkbox"/>	[Associate]

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

Elastic IP		Private IP	Use with AZ HA	
52.211.145.138 ▾	→	10.0.0.160 ▾	<input type="checkbox"/>	[Associate]
52.18.181.235	→	10.0.0.160	<input checked="" type="checkbox"/>	[Disassociate]

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

Elastic IP		Private IP	Use with AZ HA	
52.18.181.235	→	10.0.0.160	<input checked="" type="checkbox"/>	[Disassociate]

Step 9 – Associate EIPs to Private IPs on the SECOND Instance

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

Note

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

Checking EIP Status

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

AZ HA STATUS

Elastic IP	Private IP	Status	Action
52.18.181.235	10.0.0.160	Local	[Disassociate]

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

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

AZ HA STATUS

Elastic IP	Private IP	Status	Action
52.18.181.235	10.0.1.160	Peer	[Associate]

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

Testing EIP failover

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

Note

This can take up to 30 seconds to complete.

Manually moving the EIP to the other Instance

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

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

Or

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

Note

Using **[Dissociate]** is the slower of the 2 options because the other device has to first detect that the EIP is down which will cause some initial delay, whereas the first option forces an immediate EIP re-association.

10. Testing – General Comments

Note

For more information on testing and verifying load balanced services, please refer to [Testing Load Balanced Services](#).

Testing Load Balanced Services

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

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

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

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

Diagnosing VIP Connection Problems

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



The image shows a status bar with three sections: 'Primary | Secondary' in green, 'Active | Passive' in green, and 'Link' in grey.

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

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:cf:18:03 brd ff:ff:ff:ff:ff:ff
    inet 192.168.110.85/18 brd 192.168.127.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet 192.168.110.90/18 brd 192.168.127.255 scope global secondary eth0
        valid_lft forever preferred_lft forever
```

The above example shows that the interface address (192.168.110.85) and the VIP address (192.168.110.90) 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).

	VIRTUAL SERVICE	IP	PORTS	CONNS	PROTOCOL	METHOD	MODE	
↑	HTTP-Cluster	192.168.110.150	80	0	HTTP	Layer 7	Proxy	
⚠	RDP-Cluster	192.168.110.150	3389	0	TCP	Layer 7	Proxy	
↓	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, the Layer 7 statistics page can be used. To access the page, navigate to: *Reports > Layer 7 Status* - a new tabbed window will be displayed:

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 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	Downtime	Thrthle
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	0				1	-	Y		0	0		-
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	0	0s	-		
Backend		0	0	0	16	0	2	4 000	56	56	21 696	3 385 782	0	0	0	0	0	0	0	0	42s UP		1	1	1		0	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	Downtme	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			

Taking Real Servers Offline

- 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.
- 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.
- 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:

	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 WebUI. Details of both can be found in [Appliance Monitoring](#).

11. More Information

Please refer to our website for all the latest [Manuals](#) and [Deployment Guides](#).

12. Loadbalancer.org Technical Support

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

13. Appendix

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 EIPs are configured via the load balancer's WebUI, they are also auto-configured in EC2. To configure the required IAM role:

1. In the AWS Console, under the **Security, Identity & Compliance** section select the **IAM** Option.
2. Select **Policies** in the Dashboard.
3. Click **Create Policy**.
4. Select the **JSON** tab.
5. Copy and paste the complete policy definition shown on the following page into the JSON window, replacing all existing text.
6. Click **Review Policy**.
7. Verify that you're happy with the configuration.
8. Type a suitable *Name & Description* for the new Policy.
9. Click **Create Policy**.
10. Select **Roles** in the Dashboard.
11. Click **Create Role**.
12. For *Choose the service that will use this role*, select **EC2**.
13. For *Select your user case*, select ***EC2 *(Allows EC2 instances to call AWS services on your behalf)**
14. Click **Next: Permissions**.
15. To view the Policy just created, change the *Filter* to **Customer Managed**.
16. Now check (tick) the policy just created.
17. Click **Next: Review**.
18. Type a suitable *Name & Description* for the new Role.
19. Click **Create Role**.

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

```

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

```

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		
Manual Configuration	<input type="checkbox"/>	?
Label	<input type="text" value="webCluster1"/>	?
IP Address	<input type="text" value="10.0.20.135"/>	?
Ports	<input type="text" value="80"/>	?
Autoscaling Group Name	<input type="text" value="ASG-1"/>	?
Autoscaling backend server port	<input type="text"/>	?
Protocol [Advanced]		
Layer 7 Protocol	<input type="text" value="HTTP Mode"/> ▼	?

Note

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

Now save the updated configuration and restart services as prompted.

Note

For more information on configuring Auto-scaling in AWS, please click [here](#).

Configuring Auto-Scaling to auto deploy a new LB.org Instance on Failure

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

Step 1 – Deploy a Load Balancer instance

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

Step 2 – Create an image of the instance

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

1. Right click the running instance and select: *Images & templates > Create Image*.
2. Enter an appropriate name & description for the image – e.g. **AS-LB-Recovery, LB recovery image**.
3. Click **Create Image** to start the image creation process, this should be completed within a few minutes, creation status can be checked under: *Images > AMIs*.

Step 3 – Configure a Launch Configuration

A launch configuration is an instance configuration template that an Auto Scaling group uses to launch EC2 instances.

1. Using the VPC Dashboard, under *Auto Scaling* select **Launch Configurations**.
2. Click the **Create Launch Configuration** button.
3. Define a name for the Launch Configuration, e.g. **LB-LC1**.
4. Under *My AMIs* Choose the AMI created in **Step 2** above.
5. Select a suitable *Instance type* - typically this will be the same as the original load balancer.
6. Under *IAM instance profile* select the IAM role that was used for the original load balancer.
7. To enable the same Elastic IP Address (EIP) to be attached to the new instance, expand the *Advanced details* section and complete steps a) and b) below:

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

```
#!/bin/bash
# set EIP_ID to the allocation ID of your Elastic IP Address
EIP_ID="eipalloc-0018577077972aa37"
# set LB_REGION to the appropriate region
LB_REGION="us-east-1"
export INSTANCE_ID=$(curl -s http://169.254.169.254/latest/meta-data/instance-id)
export IPV4_ADDRESS=$(curl -s http://169.254.169.254/latest/meta-data/local-ipv4)
/usr/local/bin/aws ec2 associate-address --instance-id ${INSTANCE_ID} --region ${LB_REGION} \
--allow-reassociation --allocation-id ${EIP_ID} --private-ip-address ${IPV4_ADDRESS} \
> /var/log/lbas.log 2>&1
```

Note

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

- change EIP_ID in line 3 to the allocation ID of your EIP – this can be found in the information pane for the EIP
- change LB_REGION 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:

8. Select the same Security group that was used for the original load balancer instance.
9. Configure the required Key pair options.
10. Click Create Launch Configuration.

Step 4 – Create an Auto Scaling Group

1. Using the VPC Dashboard, under *Auto Scaling* select **Auto Scaling Groups**.
2. Click the **Create Auto Scaling group** button.
3. Enter an appropriate *Group name*, e.g. **AS1**.
4. Under *Launch Template* click **Switch to Launch Configuration**.
5. Select the *Launch Configuration* created in the previous step.
6. Click **Next**.

7. Select your *VPC* and *Subnet(s)*.
8. Click **Next**.
9. Under *load balancing*, leave **No Load Balancer** selected.
10. Under *Health Checks*, change the *health check grace period* if required.
11. Click **Next**.
12. Configure the group size and scaling policies to suit your environment.
13. Click **Next**.
14. Configure any required notifications and Click **Next**.
15. Configure any required Tags and Click **Next**.
16. Review all settings and click **Create Auto Scaling group**.

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

Note

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

Testing

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

Completely Terminating the Instance

Don't 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.

Configuring a VPC Endpoint to enable AWS API Access

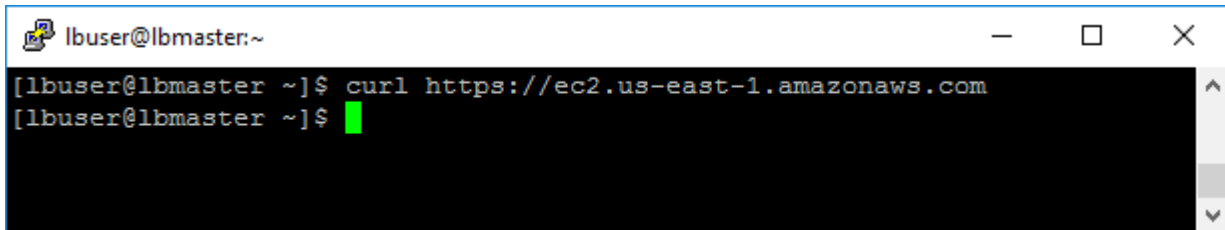
As explained in [IP address Allocation Options & Requirements](#), Endpoints are required to enable access to the AWS API. The steps required to configure an Endpoint are as follows:

1. Open the VPC Dashboard and select **Endpoints**.
2. Click **Create Endpoint**.
3. Leave *Service Category* set to **AWS Service**.
4. Under *Service Name* select the EC2 service, for us-east, the full name is **com.amazonaws.us-east-1.ec2**.

<input type="radio"/>	com.amazonaws.us-east-1.dynamodb	amazon	Gateway
<input type="radio"/>	com.amazonaws.us-east-1.ebs	amazon	Interface
<input checked="" type="radio"/>	com.amazonaws.us-east-1.ec2	amazon	Interface
<input type="radio"/>	com.amazonaws.us-east-1.ec2messages	amazon	Interface

5. Select the *VPC* where the load balancer instances are located.

6. Select the *subnet(s)* where the load balancer instances are located.
7. Leave *Enable DNS name* ticked (checked).
8. Select/configure a *Security group* that allows inbound access on TCP port 443.
9. Configure the *Policy* to suit your requirements.
10. Click **Create endpoint**.
11. To check that the load balancer instance(s) have the required access, SSH into each appliance and test the connection using curl as shown below:

A terminal window titled 'lbuser@lbmaster:~' with standard window controls. The prompt is '[lbuser@lbmaster ~]\$'. The command 'curl https://ec2.us-east-1.amazonaws.com' has been entered and executed. The prompt is now '[lbuser@lbmaster ~]\$' followed by a green cursor. There is no visible output from the command, indicating a successful connection.

```
lbuser@lbmaster:~  
[lbuser@lbmaster ~]$ curl https://ec2.us-east-1.amazonaws.com  
[lbuser@lbmaster ~]$
```

This shows a successful connection to the AWS API Endpoint.

About Loadbalancer.org

Loadbalancer.org's mission is to ensure that its clients' businesses are never interrupted. The load balancer experts ask the right questions to get to the heart of what matters, bringing a depth of understanding to each deployment. Experience enables Loadbalancer.org engineers to design less complex, unbreakable solutions - and to provide exceptional personalized support.



United Kingdom

Loadbalancer.org Ltd.
Compass House, North Harbour
Business Park, Portsmouth, PO6 4PS
UK: +44 (0) 330 380 1064
sales@loadbalancer.org
support@loadbalancer.org

United States

Loadbalancer.org, Inc.
4550 Linden Hill Road, Suite 201
Wilmington, DE 19808, USA
TEL: +1 833.274.2566
sales@loadbalancer.org
support@loadbalancer.org

Canada

Loadbalancer.org Appliances Ltd.
300-422 Richards Street, Vancouver,
BC, V6B 2Z4, Canada
TEL: +1 866 998 0508
sales@loadbalancer.org
support@loadbalancer.org

Germany

Loadbalancer.org GmbH
Tengstraße 2780798,
München, Germany
TEL: +49 (0)89 2000 2179
sales@loadbalancer.org
support@loadbalancer.org