LOADBALANCER

MANUALS

Enterprise AWS Configuration Guide

Version 8.11.3 Revision 1.0.0

Table of Contents

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 an advanced load balancing solution within the Amazon cloud.

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 LBOS-7 which is a customized Linux build maintained by Loadbalancer.org, LVS, Ldirectord, Linux-HA, HAProxy & STunnel.

Enterprise AWS provides advanced Layer 4/7 load balancing which automatically distributes incoming application traffic across EC2 instances either in a single Availability Zone, or across multiple Zones. Traffic can be distributed on the internal Amazon network (reducing bandwidth costs) or to any accessible internet address.

Full support for Service Discovery, including integration with AWS Auto Scaling which ensures that servers or containers are automatically added to the load balanced cluster as backend capacity is increased to meet inbound traffic demands.

Support for TCP and UDP enables load balancing of virtually any protocol. SSL termination and re-encryption - coupled with a built-in, OWASP top 10 compliant WAF - enables you to create a secure, robust interface for your infrastructure. URL re-writing/content switching can be used to direct traffic based on defined rules.

Advanced, customizable health checks ensure rapid detection of unhealthy instances and enable traffic to be rerouted to healthy instances.

Enterprise AWS also includes GSLB for multi-site deployments and comes with a fully featured API.

Enterprise AWS can be deployed in the following ways:

- As an HA pair (Primary & Secondary) in a single Availability Zone
- As an HA pair (Primary 1 and Primary 2) split between 2 Availability Zones
- As a single instance

> **Note** For more information on appliance deployment options, please refer to Deployment Concepts.

2.1. Regions Supported

The appliance is available in the following regions:

Region Name	Code
US East (Ohio)	us-east-2
US East (N. Virginia)	us-east-1
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
Africa (Cape Town)	af-south-1
Asia Pacific (Hong Kong)	ap-east-1
Asia Pacific (Jakarta)	ap-southeast-3
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Osaka)	ap-northeast-3
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
Canada (Central)	ca-central-1
Europe (Frankfurt)	eu-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2
Europe (Milan)	eu-south-1
Europe (Paris)	eu-west-3
Europe (Stockholm)	eu-north-1
Middle East (Bahrain)	me-south-1
South America (São Paulo)	sa-east-1
AWS GovCloud (US-East)	us-gov-east-1
AWS GovCloud (US-West)	us-gov-west-1

2.2. Instance Types Supported

Type	vCPUs	Memory (GB)	Storage (GB)	Network Performance
t3.micro	2	1	EBS only	Up to 5 Gigabit
t3.medium	2	4	EBS only	Up to 5 Gigabit
c5.xlarge	4	8	EBS only	Up to 10 Gigabit
c5.2xlarge	8	16	EBS only	Up to 10 Gigabit
c5.12xlarge	48	96	EBS only	12 Gigabit

Type	vCPUs	Memory (GB)	Storage (GB)	Network Performance
c5.24xlarge	96	192	EBS only	25 Gigabit

When deploying a new instance, the default type is **t3.medium**. This can be changed depending on your computing, memory, networking and storage needs.

> **Note** Resources required for a particular deployment depend on multiple factors including the application being load balanced, the number of end-users, the anticipated throughput and whether you'll be load balancing at layer 4 or layer 7. Therefore it's not realistic to make generic recommendations. If you need assistance in determining the resources required for your deployment, please contact support.

For more information on AWS instance types, please refer to: https://aws.amazon.com/ec2/instance-types/.

2.3. Enhanced Networking

For the load balancer to support it's maximum instance type speed, SR-IOV (single root I/O virtualization) is enabled by default. SR-IOV is a method of device virtualization that provides higher I/O performance and lower CPU utilization when compared to traditional virtualized network interfaces. Enhanced networking provides higher bandwidth, higher packet per second (PPS) performance, and consistently lower inter-instance latencies.

2.4. Licensing & Costs

Loadbalancer.org offers a flexible range of licensing options to suit your requirements:

- **Annual Subscription** - For long term deployments, an annual subscription can be purchased that saves 15% when compared to an hourly subscription.
- **Pay-As-You-Go (PAYG - Hourly)** - Pay by the hour for flexibility and control of your costs.
- **Bring-Your-Own-License BYOL** - Purchase a perpetual license for long term deployments or when you already have a license for a different platform/cloud provider and want to move your appliance to AWS by utilizing the Freedom License.

> **Note** For more information on the Freedom License, please contact sales.

Up-to-date pricing information is available here.

2.5. Main Differences to our Standard (Non-Cloud) Product

Enterprise AWS is based on our standard 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.

1. The network setup is customized for Amazon EC2 deployment.
2. The primary ENI IP address is allocated by DHCP and cannot be changed.
3. Layer 7 SNAT mode (with TProxy/transparency) & Layer 4 NAT mode where the default gateway of the Real

Servers is configured to be an IP on the the load balancer is not supported.

- Instead, the routing table for the Real Server subnet should be modified so that return traffic passes back via the load balancer. For more information, please refer to Configuration Example 2.
- For an HA pair of load balancers (Primary & Secondary) in a single AZ, the AWS routing table for the Real Server subnet must be changed when failover occurs. This ensures that return traffic always passes back via the active load balancer. This can be achieved using the WebUI option: ***Cluster Configuration > Heartbeat Advanced***, and the AWS CLI command **aws ec2 replace-route**. For more information, please refer to Configuring HA using 2 Instances (Primary & Secondary) in a Single AZ.
- For an HA pair of load balancers (Primary 1 & Primary 1) split between 2 AZs, if the same Real Servers are used with both load balancer instances, the routing table for the respective subnets must be changed when EIP failover occurs. This is to ensure that return traffic always passes back via the load balancer with the active EIP association. This can be achieved by editing the AZ HA failover script on both instances (/etc/loadbalancer.org/scripts/azhaFailover) and using the AWS CLI command **aws ec2 replace-route**. For more information, please refer to Configuring HA using 2 Instances (Primary 1 & Primary 2) Split Between 2 AZs.

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. For more information, please refer to this blog.

3. Why use Enterprise AWS?

3.1. Product Features & Support

- **Comprehensive features** - Enterprise AWS supports all features provided by the various elastic load balancers as well the following additional features within a single product:
  - Ability to load balance both EC2 based and non-EC2 based servers.
  - Customizable timeouts for custom applications.
  - Comprehensive back-end server health-check options.
  - Enables fallback servers to be configured and invoked when all load balanced servers/services fail.
  - Multiple persistence methods.
  - Full integration with Remote Desktop Services Connection Broker.
  - Multiple load balanced services running on multiple IP addresses.
  - GSLB for multisite load balancing.
  - Fully featured WAF (Web Application Firewall).
- **Ease of use** - The interface is virtually identical to our hardware/virtual product which is very simple and intuitive to use. This also makes migrations to AWS much easier for existing customers.
- **Freedom license** - Our freedom license enables customers to migrate from one environment to another (e.g. virtual to cloud) at no additional cost - and with free migration assistance.
- **Expert assistance is available 24 x 7** - Our very experienced support team can assist when needed.

> **Note** For a detailed list of features and a comparison to the various elastic load balancers, please

refer to our Enterprise AWS Product Page.

3.2. Rapid Deployment

Enterprise AWS is very quick and simple to deploy. A simple PoC consisting of a VPC with private and public subnets, Enterprise AWS and a Linux test web server can be configured in less than 30 minutes.

> **Note** For more details, please refer to Configuration Example 1.

3.3. Use Cases

Enterprise AWS can be used to provide advanced load balancing for many cloud based applications and services. For example:

- Load balance web servers to improve website performance and availability.
- Protect front-end application servers from a wide range of attacks, including the OWASP Top 10 by deploying the built-in WAF.
- Leverage the built in GSLB features to provide load balancing and failover across multiple locations.
- Provide advanced load balancing for cloud based medical systems that utilize protocols such as DICOM and HL7.

4. Deployment Concepts

4.1. AWS - Key Concepts

Regions & Availability Zones

Each Region is designed to be isolated from the other Regions. This achieves the greatest possible fault tolerance and stability. Each Region has multiple, isolated locations known as Availability Zones.

VPC & Subnets

A virtual private cloud (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS Cloud. You can specify an IP address range for the VPC, add subnets, add gateways, and associate security groups.

A subnet is a range of IP addresses in your VPC. You launch AWS resources, such as Amazon EC2 instances, into your subnets. You can connect a subnet to the internet, other VPCs, and your own data centers, and route traffic to and from your subnets using route tables.

If a subnet is associated with a route table that has a route to an Internet Gateway, it's known as a public subnet. If a subnet is associated with a route table that does not have a route to an internet gateway, it's known as a private subnet.

> **Note** For more information on VPCs, please refer to What is an AWS VPC.

ACL's & Security Groups

Security groups allow or deny specific inbound and outbound traffic and are associated with network interfaces. If an instance has a single ENI, the security group effectively applies to the instance. If the instance has multiple ENI's, a different security group can be associated with each ENI.

Network ACLs allow or deny specific inbound and outbound traffic at the subnet level.

Route Tables

A route table contains a set of rules, called routes, that determine where network traffic from your subnet or gateway is directed. There are several types or route table - for full details, please refer to VPC Route Tables.

Internet Gateway

An internet gateway enables resources in your public subnets (such as EC2 instances) to connect to the internet if the resource has a public IPv4 address or an IPv6 address. Similarly, resources on the internet can initiate a connection to resources in your subnet using the public IPv4 address or IPv6 address.

NAT Gateway

A NAT gateway is a Network Address Translation (NAT) service. You can use a NAT gateway so that instances in a private subnet can connect to services outside your VPC but external services cannot initiate a connection with those instances.

Using the Load Balancer instance as the NAT Gateway

If your load balancer instance is located in a public subnet and your Real Servers are located in a private subnet, Internet access for the Real Servers can be provided via the load balancer. To configure this:

1. Disable the Source/Destination check for the load balancer instance(s):
   - Right-click the instance and select: ***Networking > Change Source/Dest. Check***.
   - Tick (check) the ***Stop*** checkbox.
   - Click **Save**.
2. Enable Auto-NAT on the load balancer:
   - Using the WebUI, navigate to: ***Cluster Configuration > Layer 4 - Advanced Configuration > Auto-NAT***.
   - Set *Auto-NAT* to **eth0**.

   - Click **Update**.

Elastic Network Interfaces (ENIs)

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.

Elastic IPs

An Elastic IP address is a reserved public IP address that you can assign to any EC2 instance in a particular region, until you choose to release it.

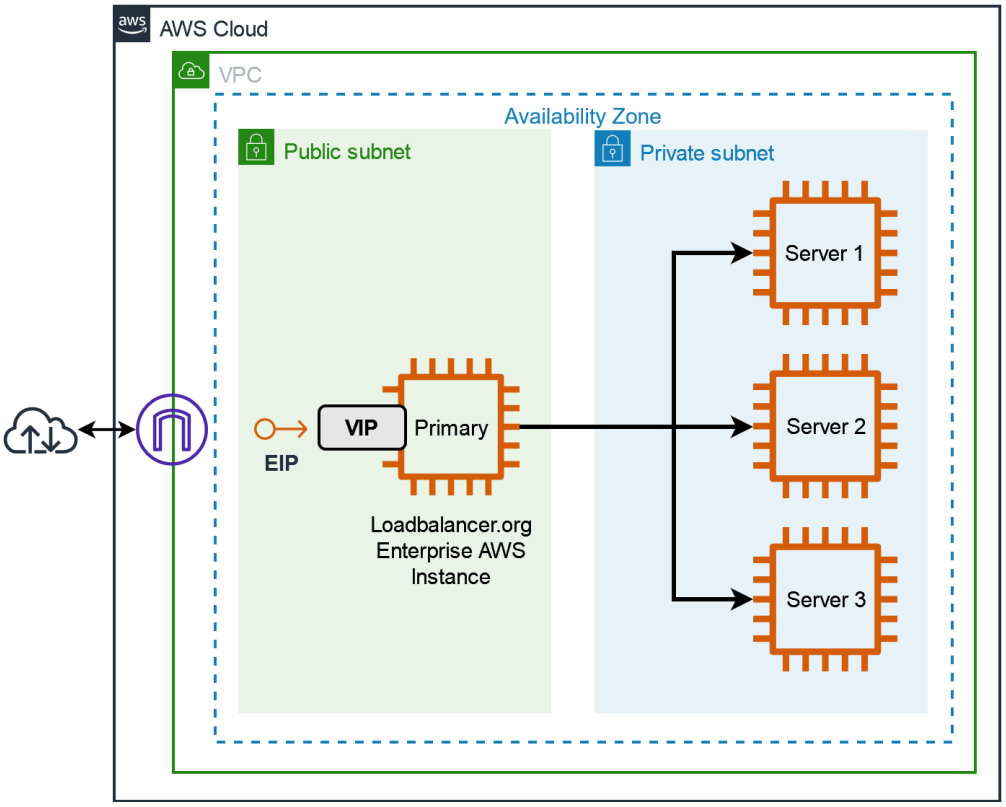
4.2. Enterprise AWS deployment Options

There are several ways in which the load balancer can be deployed. The options available depend on whether you intend to deploy a single appliance or two appliances for HA and whether you're deploying to single or dual availability zones. The options are explained below.

Single Availability Zone

Single Appliance

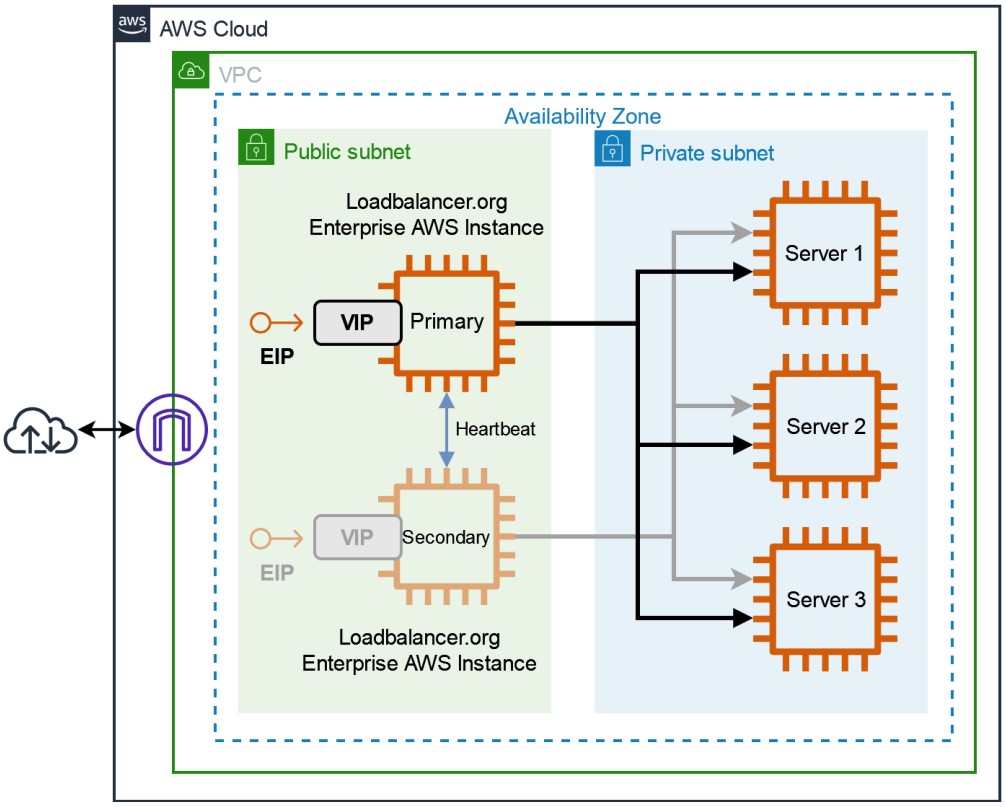
A single instance is deployed.



- 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 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.



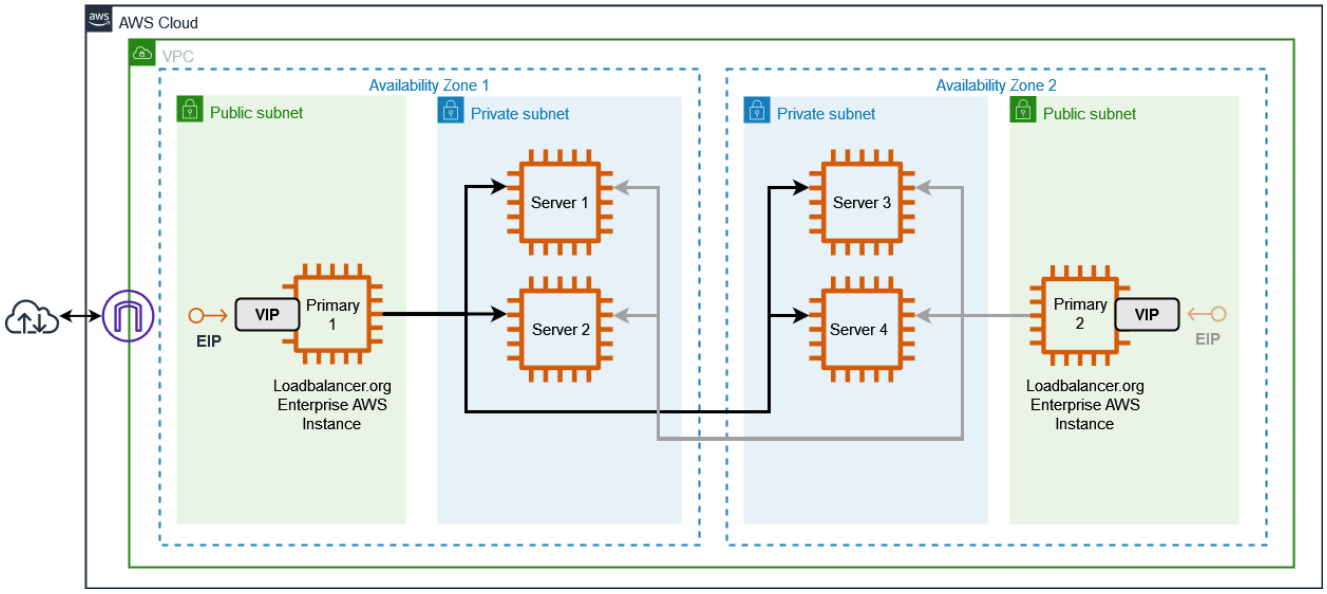
- Under normal conditions the Primary is active and the Secondary is passive. If the Primary fails, the load balanced services (VIPs) will be automatically brought up on the Secondary. When failover occurs, the EIP is still associated with the same private IP address, but it's now active on the Secondary.
- 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 HA using 2 Instances (Primary & Secondary) in a Single AZ for detailed steps on configuring and using this mode.

Dual Availability Zones

2 Instances in AZ HA Mode

This mode enables two load balancer instances to be configured in different 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 Real Servers, the example below shows one possible scenario.



- A VIP is configured on each load balancer instance. In the above example, all four Real Servers are associated with both VIPs.
- The VIP on Primary 1 and Primary 2 are both locally active, but the EIP is only associated with one of the instances - in the above example, the EIP is associated with the VIP on Primary 1.
- Primary 2 regularly checks that the EIP is up via Primary 1 and if not, the EIP is associated with Primary 2 after the check timeout has been reached.
- Under normal circumstances, whether the EIP is associated with the VIP on Primary 1 or Primary 2, all four Real Servers are available.
- In the above example, if AZ-1 fails, Primary 1, Server 1 and Server 2 will be unavailable. This will trigger Primary 2 to associate the EIP with it's own VIP and services will continue to be provided from Server 3 and Server 4.
- The WebUI can be used to force the EIP to be associated with the VIP on Primary 2 rather than the VIP on Primary 1.
- Please refer to Configuring HA using 2 Instances (Primary 1 & Primary 2) Split Between 2 AZs for detailed steps on configuring and using this mode.

4.3. AWS API 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. API calls are used to provide automatic integration.

When VIPs are added that are not bound to the Primary IP address of the instance, additional secondary private IP addresses must be added. When these IP's are added using the appliance's WebUI they are automatically added as secondary IPs to the instance in AWS provided the instance has access to the AWS API.

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 to the autoscaling group, the server will also be added as a Real Server for the Virtual Service. The same applies when servers are removed. For more information, please refer to Autoscaling.

> **Note** An appropriate IAM role must also be configured. For more information, please refer to IAM Role.

4.4. Amazon API Access Requirements

Single Appliance Deployments

- If your VIPs will be configured on the primary private IP address of the appliance then no access to the Amazon API is required.
- If your VIPs will be configured on additional secondary IP address, then access to the Amazon API is required.
  - If the appliance is deployed in a public subnet, access to the Amazon API is provided by default.
  - If the appliance is deployed in a private subnet, a VPC Endpoint must be created to enable access to the API.

Clustered Pair Deployments (Primary & Secondary)

- VIPs cannot be configured on the primary private IP address of the appliance and secondary IPs must be used instead. Therefore access to the Amazon API is required in all cases.
  - If the appliances are deployed in a public subnet, access to the Amazon API is provided by default.
  - If the appliances are deployed in a private subnet, a VPC Endpoint must be created to enable access to the API.

Dual AZ Deployments (Primary 1 & Primary 2)

- Access to the Amazon API is required in all cases. This enables the EIP(s) to be moved between instances if a failover occurs.

> **Note** For details on creating a VPC Endpoint, please refer to Configuring an AWS Service Endpoint.

5. Security Best Practices

> **Note** General AWS security best practices can be found at https://aws.amazon.com/architecture/security-identity-compliance/.

In addition to following the AWS guidelines, the following best practices also apply:

5.1. Login Credentials

By default, you can login to the WebUI using the username **loadbalancer** and the Enterprise AWS instance ID as the password. Whilst the instance ID is relatively secure, since it can be viewed by others who have access to the AWS portal but may not need access to the load balancer, it is strongly recommended to change it once deployed.

5.2. Opening Ports

By default, when Enterprise AWS is deployed there is a predefined security group that can be selected. As described here this group opens ports 9443 (WebUI), 22 (SSH) and 6694 (Heartbeat). You'll also need to include the port(s) used by your application. Only open ports that are needed and also lock down the source address as far as possible.

5.3. Administrative Access

By default, the WebUI is accessed on port 9443. This can be changed if required - for more information, please refer to Appliance Security. SSH can also be used as detailed in Accessing the Appliance using SSH.

As mentioned in Appliance Security, full root access can be enabled. This should only be used when needed and not for day-to-day administration tasks.

5.4. Key Pair Storage & Protection

A key pair, consisting of a public key and a private key, is a set of security credentials that you use to prove your identity when connecting to the Enterprise AWS instance. Amazon EC2 stores the public key on your instance, and you store the private key. The private key allows you to securely SSH into your instance. Anyone who possesses your private key can connect to your instances, so it's important that you store your private key in a secure place.

The AWS Secret Manager can be used to store and protect your keypairs. For more information, please refer to this AWS Security Blog.

5.5. IAM Role

Enterprise AWS requires additional permissions to allow it to make EC2 API requests. These requests enable EC2 console functions to be called automatically. for example, when secondary IPs configured via the load balancer's WebUI, they are also automatically configured in EC2.

To configure an IAM role to enable the required permissions:

1. Open the IAM Management Console.
2. Under *Access management*, select *Policies* and click **Create policy**.
3. Select the JSON tab and copy and paste the complete policy definition shown below into the JSON window, replacing all existing text.
4. Click **Next: Tags**.
5. Click **Next: Review**.
6. Enter a suitable *Name*, e.g. **LB-API-Policy**.
7. Click **Create policy**.
8. Under *Access management*, select *Roles* and click **Create role**.
9. Leave *Trusted entity type* set to **AWS Service**.
10. Set *Use case* to **EC2**.

11. Click **Next**.
12. Under *Add Permissions*, select the policy just created, e.g. **LB-API-Policy**.
13. Click **Next**.
14. Enter a suitable role name, e.g. **LB-API-Role**.
15. Click **Create role**.

IAM Policy Definition:

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

> **Note** The IAM role should be configured in advance. The allows the IAM role to be selected during instance deployment.

6. Deploying Enterprise AWS

6.1. Minimum deployment

A minimal public facing deployment can be configured that uses the following AWS resources:

- VPC
- Public Subnet
- Internet Gateway
- Route Table
- Security Group
- Load balancer instance
- 2 or more Real Server instances (e.g. AWS Linux based web servers)
- EBS Volumes (1 per instance)
- Elastic Network Interfaces (1 per instance)
- Elastic IP

6.2. Knowledge requirements

A successful deployment requires familiarity with the configuration of Enterprise AWS using the WebUI aa described in the manual and also with the following AWS services as a minimum:

- Amazon EC2
- Amazon VPC
- Amazon IAM

It's anticipated that Enterprise AWS will be deployed by AWS cloud Engineers who are familiar with the above.

> **Note** If you're new to AWS, please refer to Getting Started with AWS.

6.3. Other Pre-Requisites

1. Have an amazon AWS account. An account can be created here: https://aws.amazon.com/.
2. An IAM role should be created before the instance is deployed. For details, please refer to IAM Role.
3. For private subnet based deployments, an AWS service Endpoint should be created to enable the load balancer instance to access the AWS API. For more information, please refer to Configuring an AWS Service Endpoint.

6.4. Create & Configure a VPC

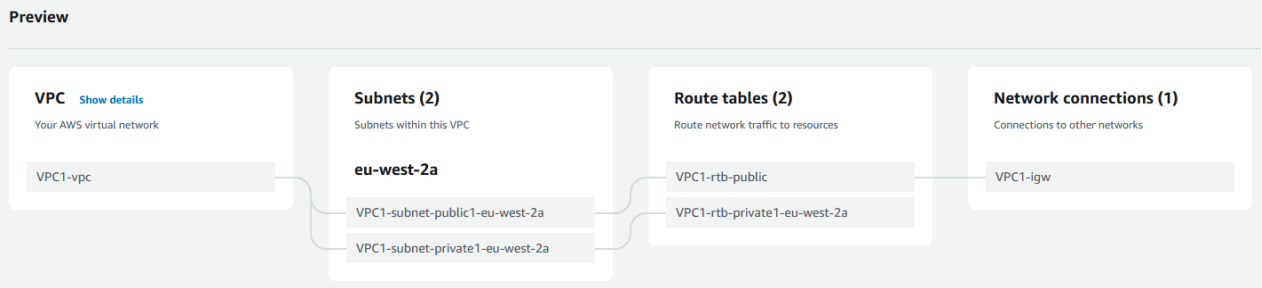
To create a VPC:

1. Open the VPC Management Console.
2. Click **Create VPC**.
3. Select **VPC and more**.

4. Enter a suitable *name* for the VPC, e.g. **VPC1**.
5. Specify an *IPv4 CIDR block*, e.g. **10.0.0.0/16**.
6. Select **Amazon-provided IPv6 CIDR block** if an IPv6 block is required.
7. Set the required *Tenancy*.
8. Set the *Number of Availability Zones (AZs)* as required.
9. Set the *Number of Public subnets* as required.
10. Set the *number of Private subnets* as required.
11. Set *NAT gateways* as required.

> **Note** If you want to provide your private subnet backend (real) servers with access to the Internet, this can be achieved by using the load balancer appliance located in the public subnet. An additional NAT gateway is not needed. For details, please refer to NAT Gateway.

12. Set *VPC endpoints* as required.
13. A visual representation of the VPC will be shown, e.g. :



14. Click **Create VPC**.

> **Note** For more information about VPCs, please refer to What is Amazon VPC?

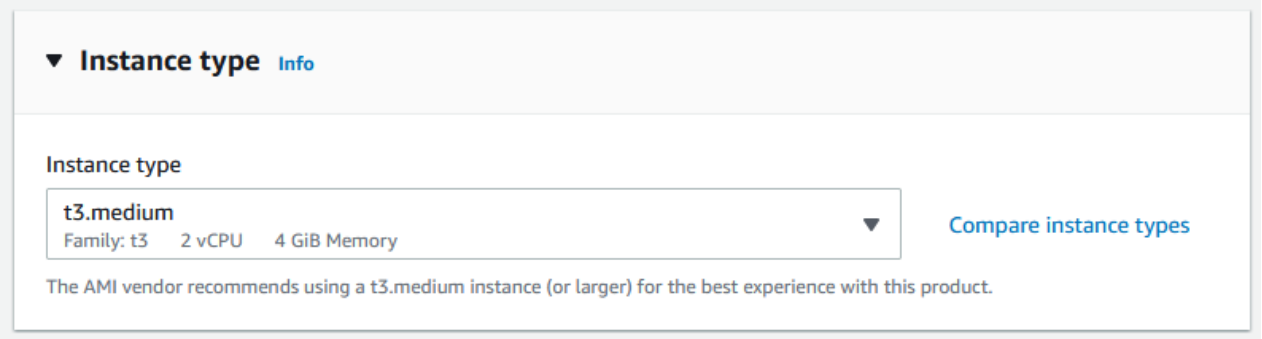
6.5. Accessing & Deploying the AMI

To access and deploy the AMI:

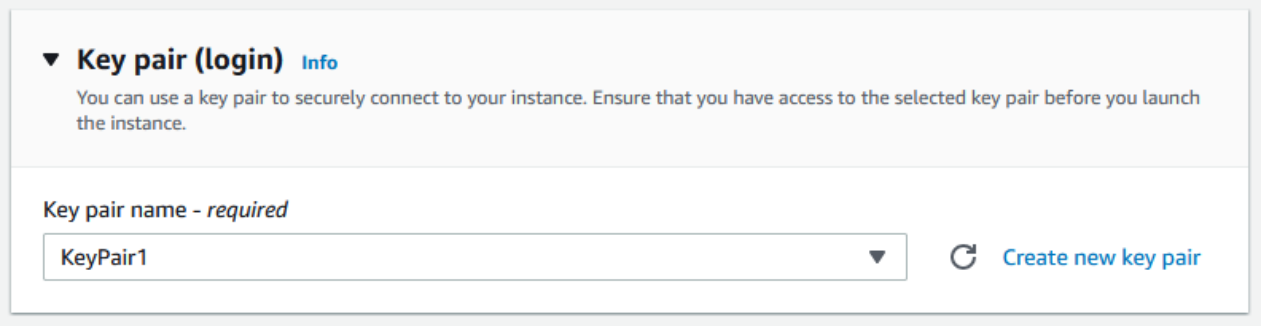
1. In the EC2 Management Console, click **Launch Instance**.
2. Enter a suitable *Name* for the instance, e.g. **LB1**.
3. Search for "Loadbalancer.org".
4. Select the *AWS Marketplaec AMIs* tab.
5. Click **Select** next to the required AMI, either:
   - **Load Balancer Enterprise ADC**
   - **Load Balancer Enterprise ADC - BYOL**

> **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.

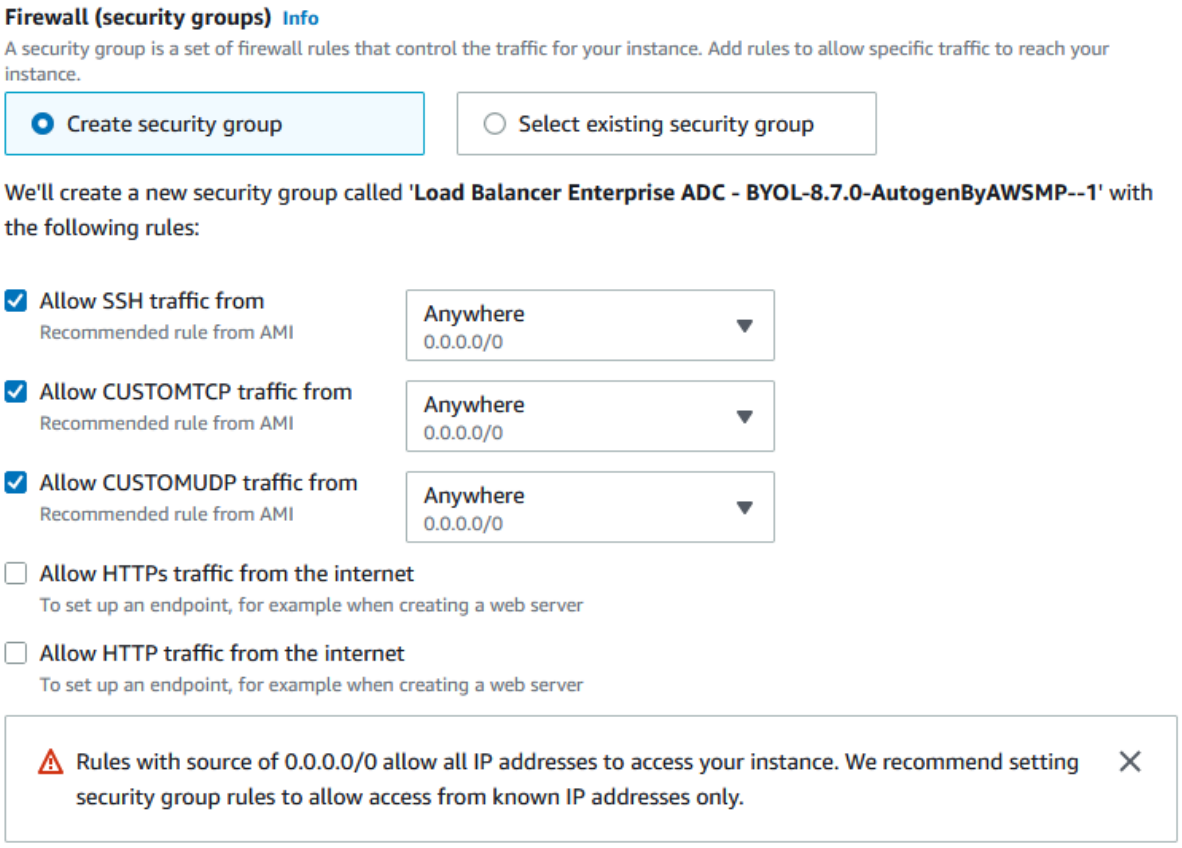
6. Review the details and if you're happy to proceed click **Continue**.
7. Select the required instance type - **t3.medium** is the default.



8. Select the required Key pair.



9. Configure the required *Network Settings*.
   - Select the required *VPC*.
   - Select the required *Subnet*.
   - Configure *Auto-assign public IP* according to your requirements.
   - Configure the required *Firewall (Security Group)* settings. By default the following rules are set if a new group is created:



- These rules are required to enable the following:
  - TCP port 22 - Appliance Management (SSH)
  - TCP port 9443 - Appliance Management (WebUI)
  - UDP port 6694 - Heartbeat between Primary and Secondary appliances (please refer to Configuring HA using 2 Instances (Primary & Secondary) in a Single AZ).
- Additional rules should 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.

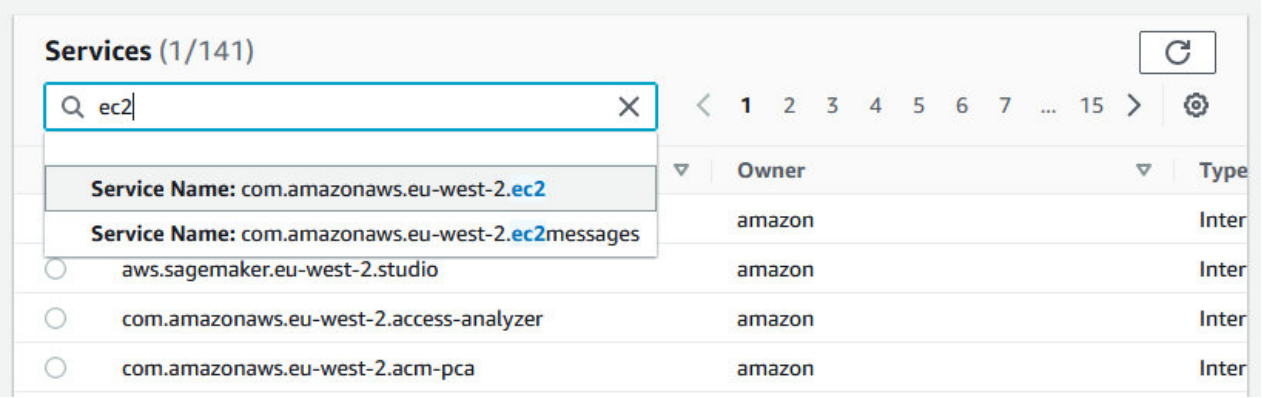
10. Configure the required ***Storage***. The default value is recommended.
11. Expand the ***Advanced Details*** section.
    - Set the ***IAM instance Profile*** to the **IAM Role** created previously - for details see IAM Role.
    - Configure other options according to your requirements.

6.6. Configuring an AWS Service Endpoint

As explained in Amazon API Access Requirements, an AWS Service Endpoint is required to enable access to the AWS API when the appliance is deployed in a private subnet. To configure an Endpoint:

1. Open the VPC Management Console and select **Endpoints**.
2. Click **Create endpoint**.
3. Enter an appropriate ***Name***, e.g. **LB-API-endpoint**.

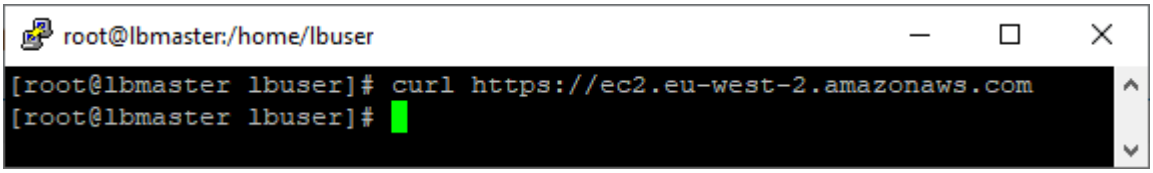
4. Leave *Service Category* set to **AWS Services**.
5. Under *Services* filter for 'ec2' and select the com.amazonaws.<region>.ec2 service, e.g. for Europe (London) the full name is **com.amazonaws.eu-west-2.ec2**.



6. Select the *VPC* where the load balancer instance(s) is located.
7. Select the *subnet(s)* where the load balancer instance(s) is located.
8. Select/configure a *Security group* that allows access to the endpoint network interface.

> **Note** Enable inbound access on port 443 and make sure the the source is the entire VPC. Specifying just the subnet address where the load balancer instance is located will not work.

9. Configure the *Policy* to suit your requirements.
10. Click **Create endpoint**.
11. To verify that the load balancer instance(s) has the required access, SSH into each appliance and test the connection using curl as shown below:

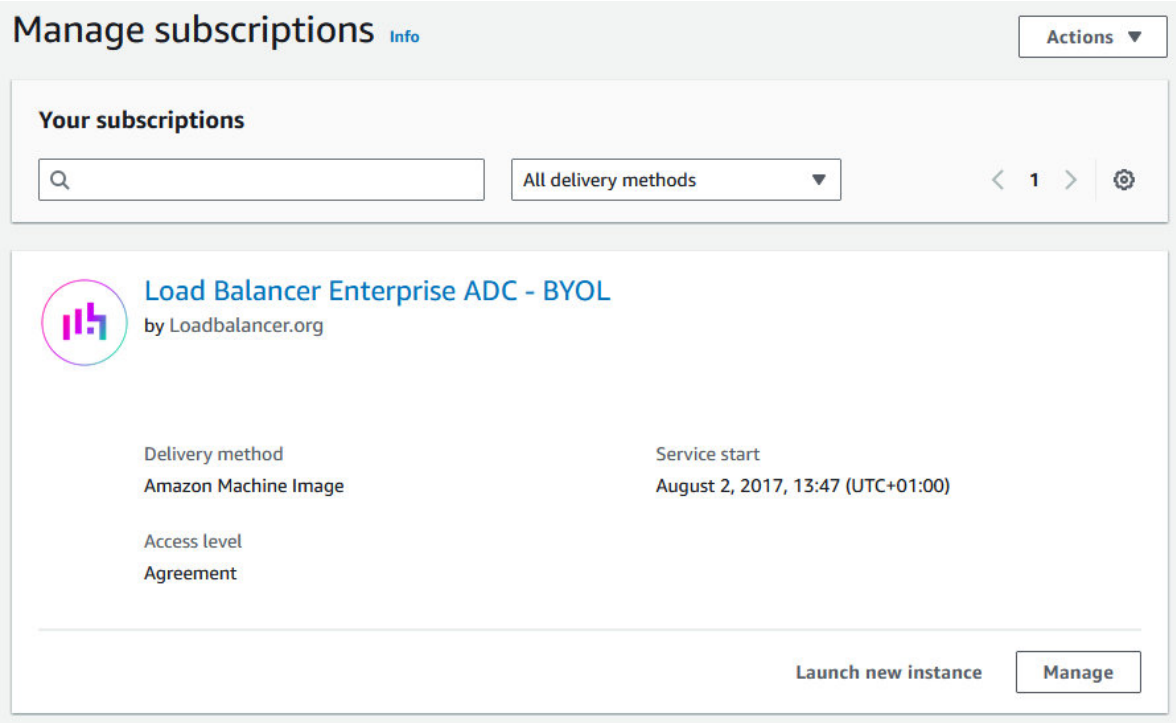


This shows a successful connection to the AWS Service Endpoint.

> **Note** For more information, please refer to AWS Service Endpoints.

6.7. Checking your Subscriptions

Current subscriptions can be managed using the *Your Marketplace Software* option in the AWS Marketplace console as shown below:



7. Accessing the Appliance

7.1. Accessing the Appliance WebUI

Using a browser, navigate to the public IP address or public DNS name on port 9443:

https://<Public IP address>:9443

or

https://<Public DNS name>:9443

> **Note** You'll receive a warning about the WebUI's SSL certificate. This is due to the default self signed certificate that is used. If preferred, you can upload your own certificate - for more information, please refer to Appliance Security Features.

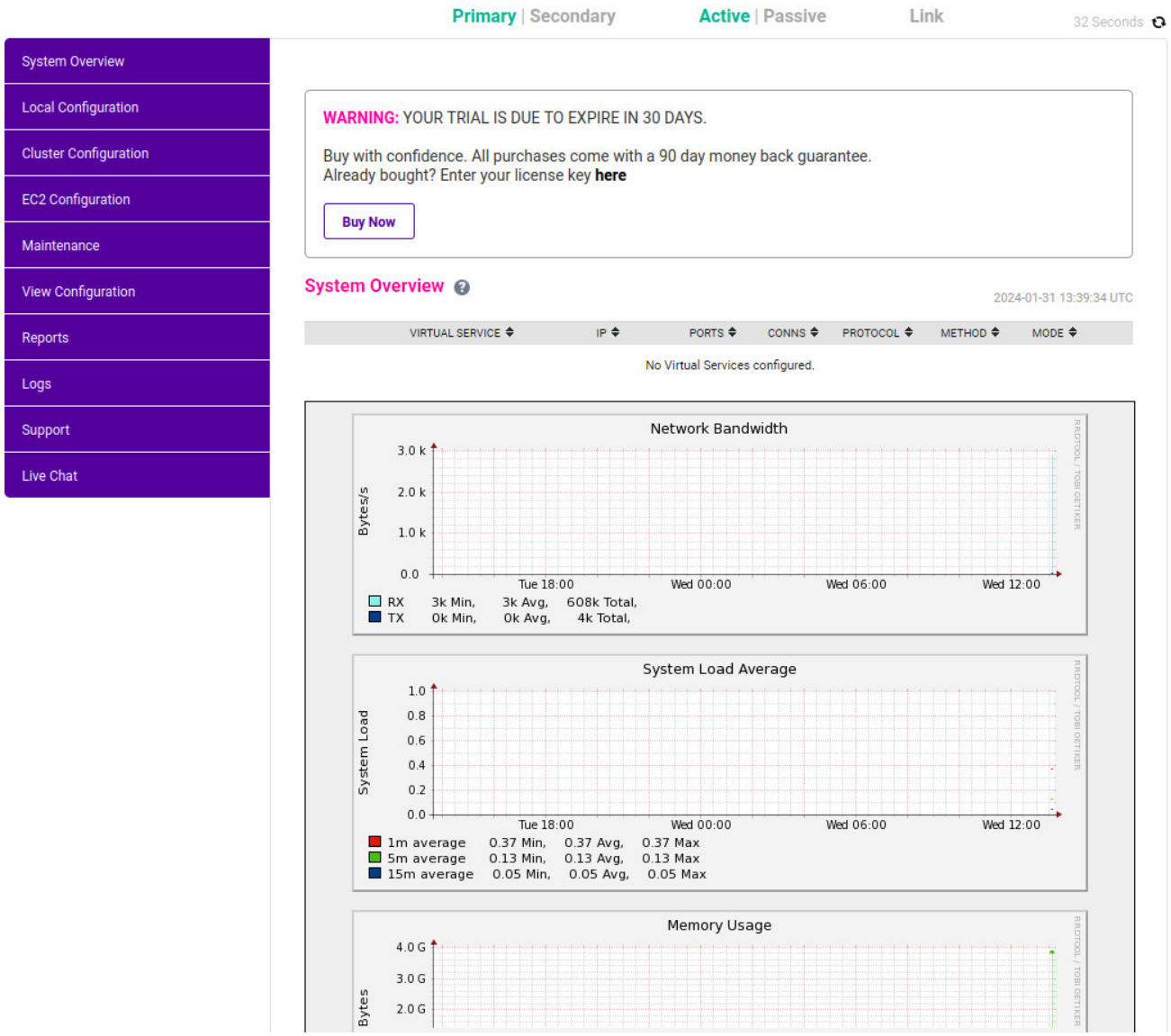
> **Note** If you need to change the port, IP address or protocol that the WebUI listens on, please refer to Service Socket Addresses.

Log in to the WebUI using the following default credentials:

Username: loadbalancer
Password: <EC2 Instance-ID>

> **Note** To change the password, 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
EC2 Configuration - Configure AWS specific settings
Maintenance - Perform maintenance tasks such as service restarts and taking backups
View Configuration - Display the saved appliance configuration settings
Reports - View various appliance reports & graphs
Logs - View various appliance logs
Support - Create a support download, contact the support team & access useful links
Live Chat - Start a Live Chat session with one of our Support Engineers

7.2. Appliance Security

> **Note** For full details of each security mode and all other 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:

- **Custom** - In this mode the security options can be configured to suit your requirements
- **Secure - (Default)** - In this mode:
  - "root" user console access & SSH password access are disabled
  - WebUI connections are forced to use HTTPS
  - Access to the *Local Configuration > Execute shell command* menu option is disabled
  - The Firewall Script & the Firewall Lockdown Wizard Script cannot be edited
- **Secure - Permanent** - This mode is the same as **Secure** but once set it cannot be changed

> **Important** Setting the security mode to **Secure - Permanent** is irreversible.

To configure the Security Mode:

1. Using the WebUI, navigate to: *Local Configuration > Security*.
2. Select the required *Appliance Security Mode*.
3. Click **Update**.

Passwords

The loadbalancer WebUI account

The password for the **loadbalancer** WebUI user account is set to the AWS Instance-ID by default. This can be changed using the WebUI menu option: *Maintenance > Passwords*.

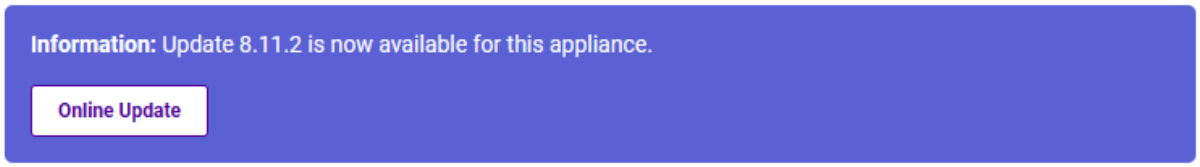
The root Linux account

it's not possible to directly log in 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
```

7.3. Appliance Software Updates

For v8.6.0 and later, the appliance periodically contacts the Loadbalancer.org update server (**update.loadbalancer.org**) and checks for updates. If an update is found, a message will be displayed at the top of the screen as shown in the following example:



To start to the update process click **Online Update**.

> **Note** Auto-check for updates can be disabled if preferred. In this case the update check must be initiated manually. For more details, please refer to Appliance Software Updates.

7.4. 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.

7.5. 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
```

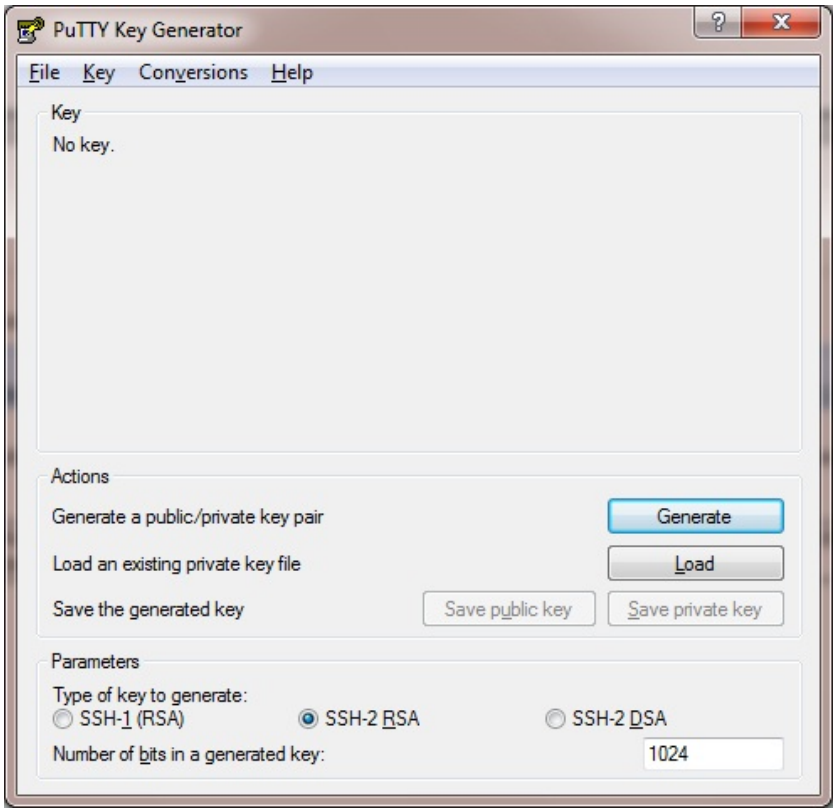
Or using the fqdn:

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

Using Windows

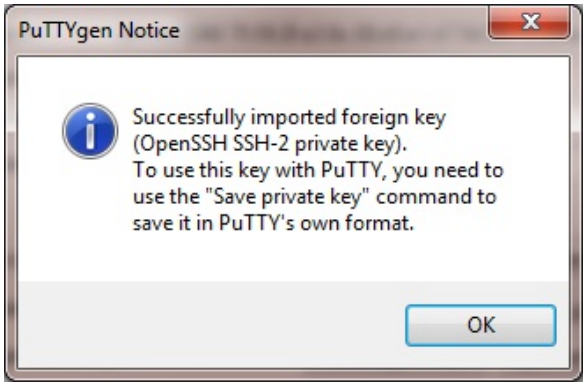
> **Note** If you create the keypair before deploying the load balancer instance using the ***Network & Security > Key Pair*** option in the EC2 console, the output file format to be set to ppk. In this case Puttygen is not needed.

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:

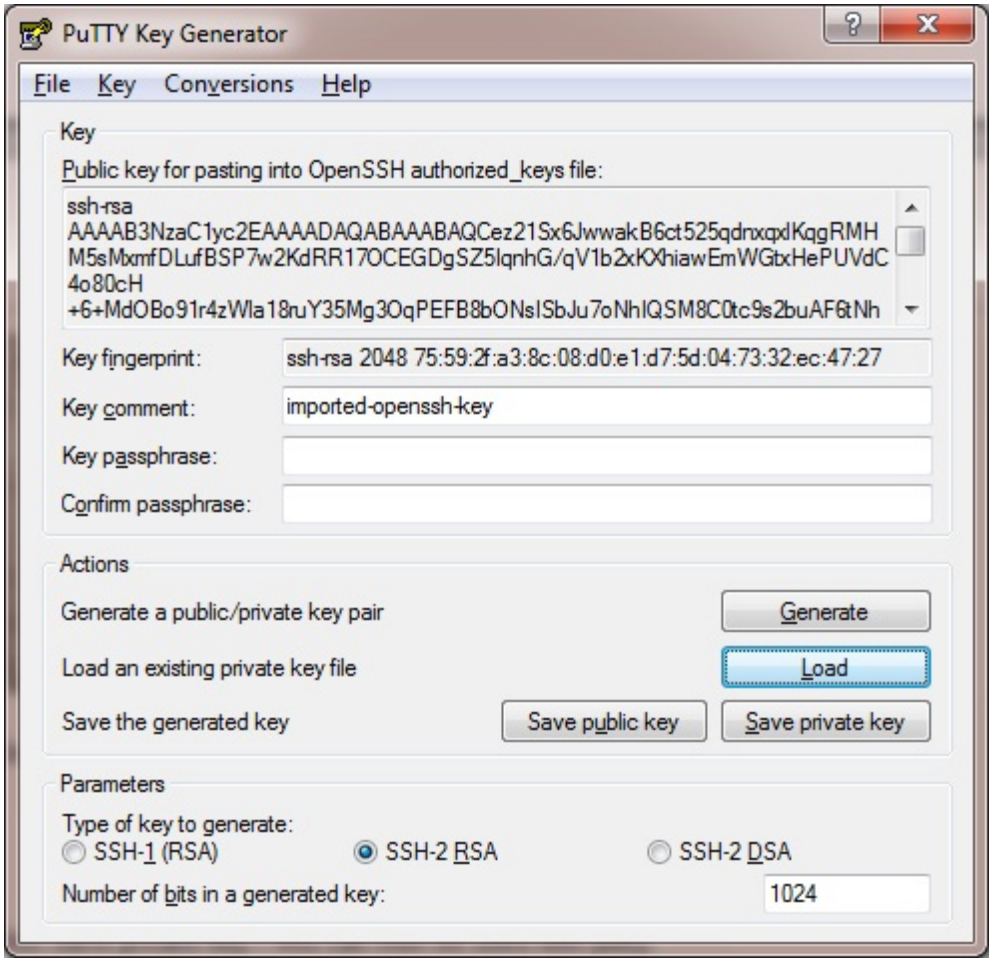


Click **Load**, change the file-type to all files and select the pem file saved earlier when creating your Key Pair.

You should see the following message:

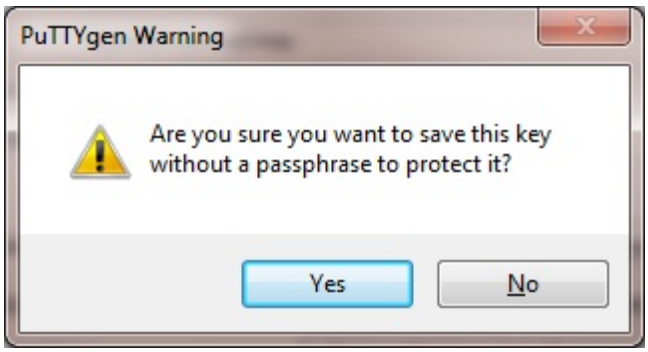


Click **OK**.



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

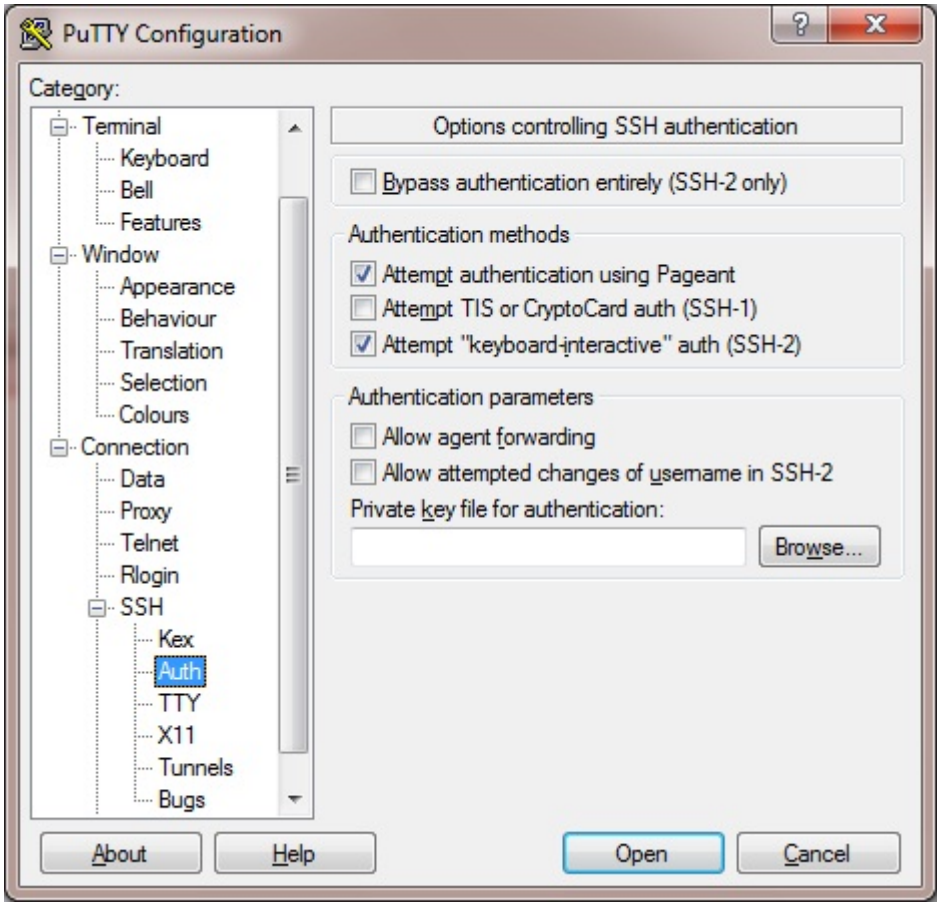
You can also choose to enter an additional pass-phrase for improved security, if you don't, the following message will be displayed:



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
> ```

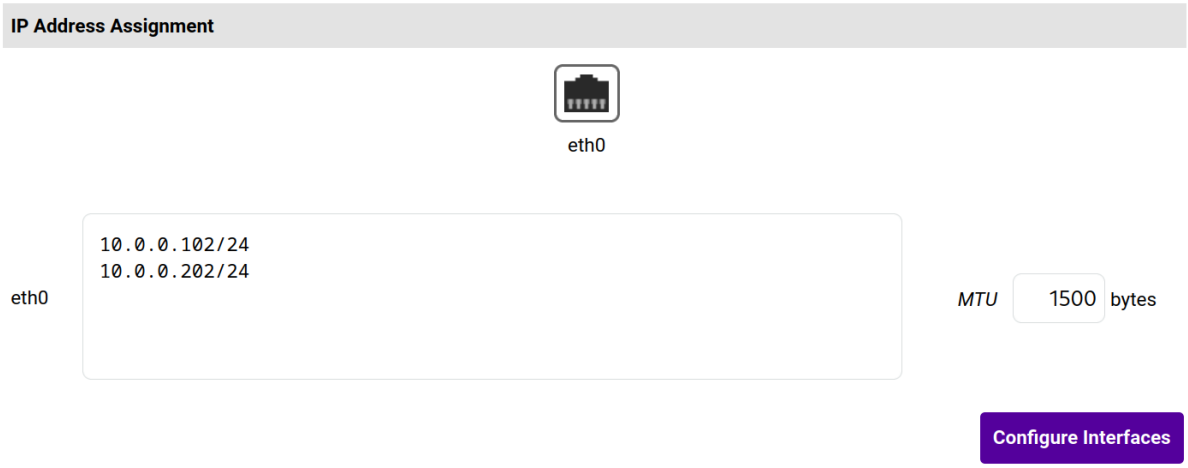
7.6. Enterprise AWS Non-standard WebUI Menu Options

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 the Administration Manual.

1. **Local Configuration > Network Interface Configuration**

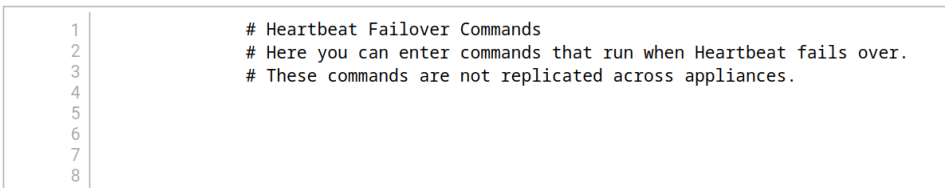
Network Interface Configuration



- Shows the private IP addresses allocated to the instance.
- The first address in the list is auto-allocated when launched - it's not possible to change the auto-allocated IP address.
- Multiple IP addresses can be assigned as shown. IP addresses added here after the first one in the list are shown as *Secondary Private IP Addresses* in the AWS/EC2 Management Console.

2. **Cluster Configuration > Heartbeat Advanced**

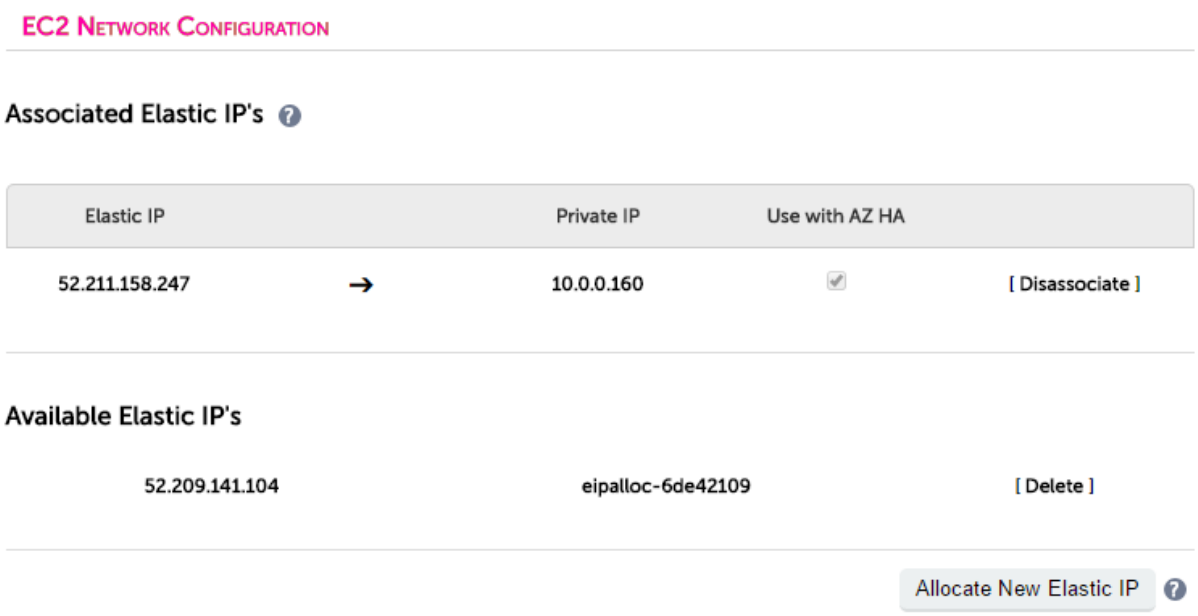
Heartbeat Failover Script



- Enables commands to be run at failover/failback for an HA pair. This includes Amazon CLI commands. For details on all CLI commands available, please refer to the Amazon CLI Command Reference.

> **Note** For an example, please refer to Configuring HA using 2 Instances (Primary & Secondary) in a Single AZ which uses **replace-route** to dynamically modify the routing table.

3. **EC2 Configuration > EC2 Network Configuration**

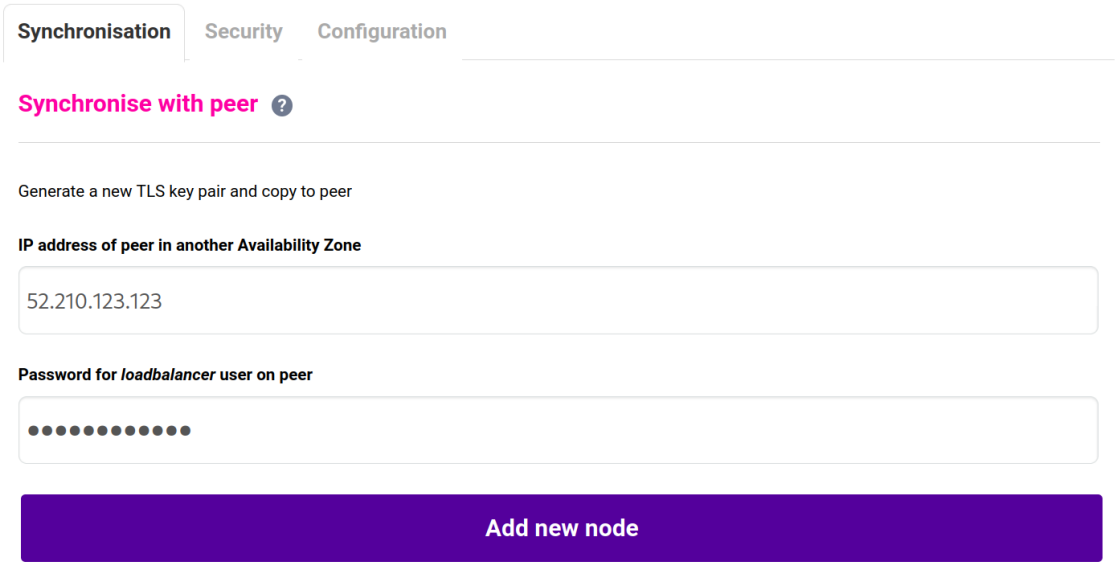


- Used to allocate/delete and associate/disassociate Elastic IPs.
- Row-1 above shows that EIP 52.211.158.247 is associated with private IP 10.0.0.160. If you want to undo the association click **[Disassociate]**.
- Row-2 above shows that EIP 52.209.141.104 is currently not associated with any Private IPs, it can be deleted by clicking **[Delete]**.
- New 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.

4. **EC2 Configuration > EC2 Zone HA Configuration**

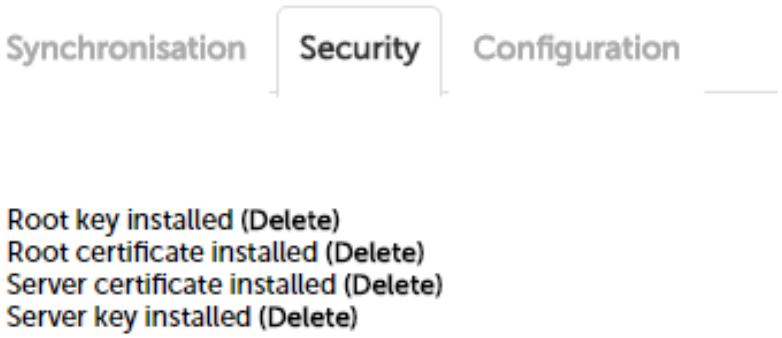
   Used to configure Zone HA, where 2 instances are deployed in different AZs in a Primary 1/Primary 2 configuration.

   - **Synchronization Tab**



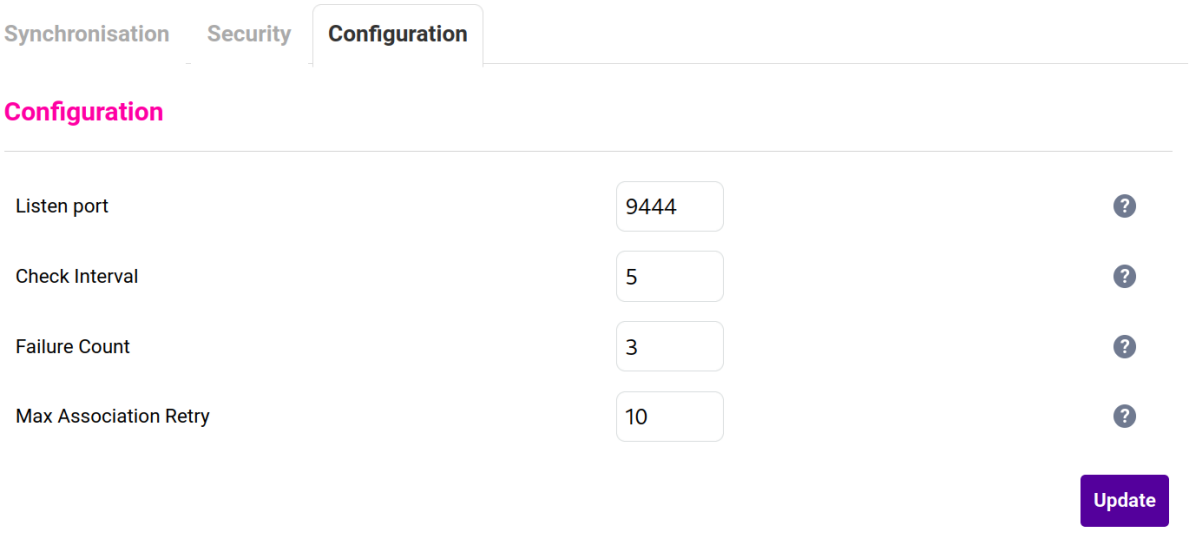
- Used to generate new keys & signed certificates and synchronize with the desired peer. The IP address of the peer instance and the password for the ***loadbalancer*** user must be entered. When **Add new node** is clicked, new keys and signed certificates will be generated and synchronized with the node specified. These keys are used to verify the peer when monitoring an Elastic IP across Availability Zones.

- **Security Tab**



- 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.

- **Configuration Tab**



- *Listen Port* - This is the port the service will listen on and connect to on the peer. The appliances in each Availability Zone should use the same port.
- *Check Interval* - This is the interval between health checks. It also sets the timeout value for when a health check is considered failed.
- *Failure Count* - This sets the desired number of health check failures before moving the Elastic IP address. The recommended value is 3 as this helps rule out temporary issues.

- *Max Association Retry* - This sets the desired number times to retry associating the elastic IP with the private IP address before giving up.

> **Note** As mentioned here, there is a $0.10 charge per Elastic IP address remap for additional remaps over 100 per month.

5. **EC2 Configuration > EC2 Zone HA Status**

Used to monitor Zone HA, where 2 Primary instances are deployed each in a different AZ in a Primary 1/Primary 2 configuration.

Primary 1 instance:

Primary 2 instance:

- In the example above, the VIP (**10.0.0.160**) on the Primary 1 instance is currently associated with the EIP.

8. Autoscaling

8.1. 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

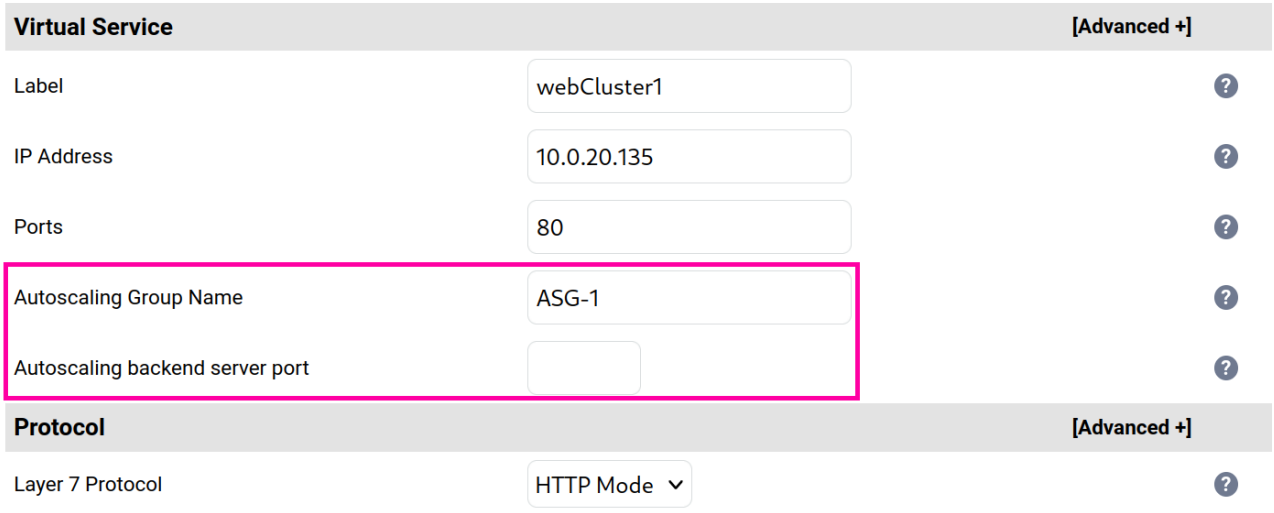
Using the EC2 Management Console, create your launch configuration and auto-scaling group according to your requirements.

Step 2 - Create the Virtual Service on the Load Balancer

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

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

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



> **Note** For Layer 7 VIPs there is an additional field called ***Autoscaling backend server port*** 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, 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 AWS Auto-scaling, please refer to What is Amazon EC2 Auto Scaling?.

9. Configuration Examples

The following sections provide 2 examples to help illustrate how the load balancer is configured:

- Example 1 - Public facing Web Servers - Dual subnet, Layer 7 - The load balancer is located in a public subnet and the Real Servers are located in a private subnet. At layer 7 the load balancer acts as a proxy so in this case there is no need to change routing rules.
- Example 2 - Public facing Web Servers - Dual subnet, Layer 4 NAT mode - The load balancer is located in a public subnet and the Real Servers are located in a private subnet. For NAT mode to operate correctly, routing rules for the Real Server subnet must be changed so that return traffic passes back via the load balancer.

9.1. Deployment Notes

High Availability

To enable HA, a second appliance can added as explained in Configuring HA using 2 Instances (Primary & Secondary) in a Single AZ and Configuring HA using 2 Instances (Primary 1 & Primary 2) Split Between 2 AZs.

> **Note** We recommend that the first appliance is fully configured & tested first and then the second appliance should be added. This applies to both HA methods.

Availability Zones

Load balanced Real Servers can be located in any Availability Zone within the region. For layer 7 VIPs (without transparency enabled), routing rules do not need to be changed as shown in Example 1 - Public facing Web Servers - Dual subnet, Layer 7. For Layer 4 NAT mode and Layer 7 (with transparency enabled), you'll need to modify the routing rules for the Real Server subnet as shown in Example 2 - Public facing Web Servers - Dual subnet, Layer 4.

Real Server Internet access via the Load Balancer Instance

If your Real Servers are located in a private subnet behind the load balancer and need Internet access for software installation, updates etc., this can be achieved by enabling ***Auto-NAT*** on the load balancer. For more details, please refer to NAT Gateway.

9.2. Example 1 - Public facing Web Servers - Dual subnet, Layer 7

The load balancer is located in a public subnet and the Real Servers are located in a private subnet. At layer 7 the load balancer acts as a proxy so in this case there is no need to change routing rules.

Step 1 - AWS Setup/Instance Deployment

1. Configure a VPC with a private and public subnet using the VPC configuration tool.
2. Deploy the load balancer instance in the public subnet as described in Deploying Enterprise AWS. Make sure you specify an appropriate IAM Role.
3. Deploy your required Real Server instances (e.g. web servers) into the private subnet.
4. If you want the Real servers to be able to access the Internet via the load balancer, please refer to NAT Gateway.
5. Ensure that the Security Group(s) allow the required inbound access. The following example uses 2 security groups. **SG-1** is associated with the Real Servers and **SG-2** is associated with the load balancer.

   SG-1

Protocol	Port	Source	Use
TCP	80	10.0.0.0/16	web traffic / health checks
TCP	22	10.0.0.0/16	web server management (using load balancer as jump box)

> **Note** Since layer 7 SNAT mode is not transparent, the source address of traffic reaching the web servers will be the load balancer. Therefore, the web traffic rule must allow traffic from the

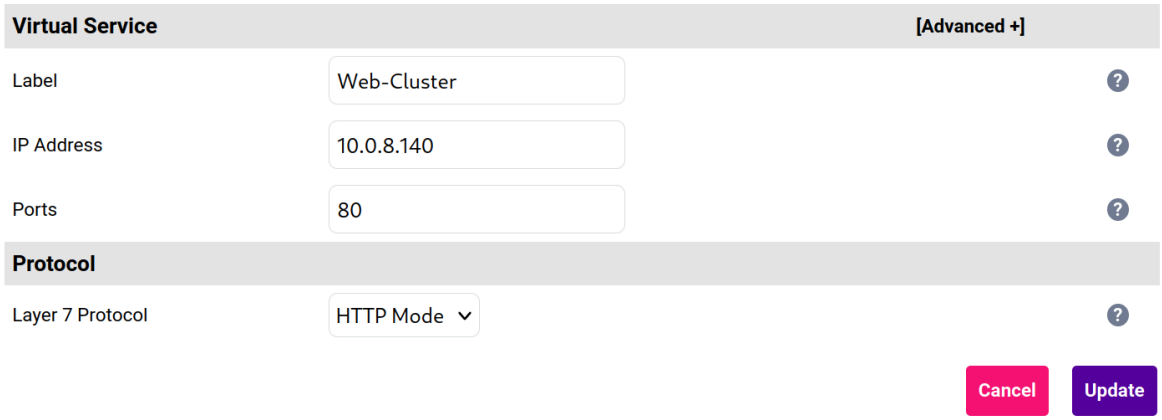
public subnet, in this case the whole VPC has been specified.

SG-2

Protocol	Port	Source	Use
TCP	80	0.0.0.0/0	web traffic
TCP	22	management subnet	appliance management
TCP	9443	management subnet	appliance management

Step 2 - Virtual Service (VIP) Configuration

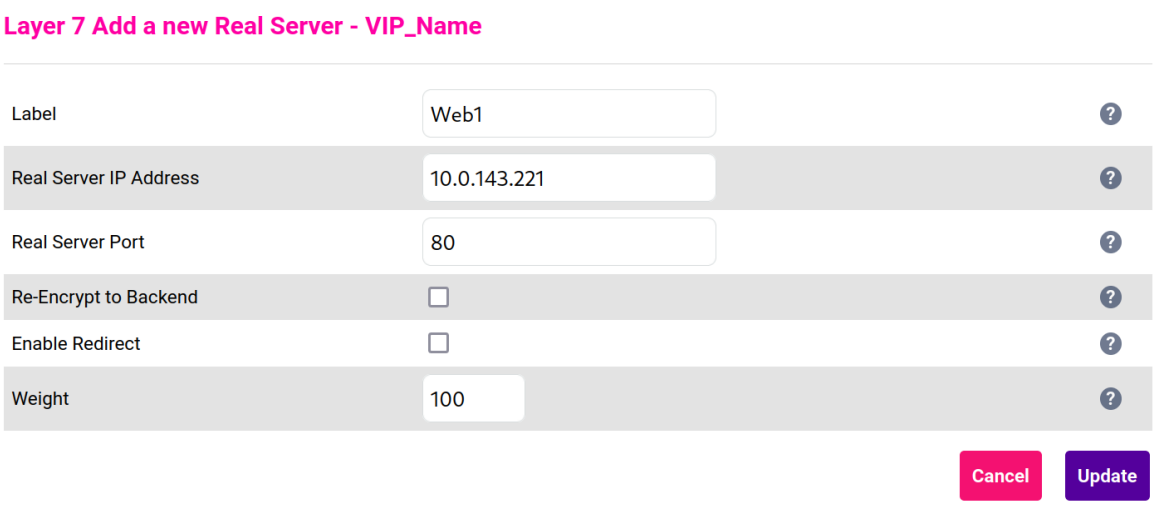
1. Using the WebUI, navigate to: *Cluster Configuration > Layer 7 - Virtual Services* and click **Add a New Virtual Service**.
2. Enter the following details:



   - Enter an appropriate label for the VIP, e.g. **Web-Cluster**.
   - Set the *Virtual Service IP address* field to the required IP address, e.g. **10.0.8.140**.
   - Set the *Virtual Service Ports* field to the required port, e.g. **80**.
   - Leave *Layer 7 Protocol* set to **HTTP Mode**.
3. Click **Update**.

Step 3 - Real Server (RIP) Configuration

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:



- Enter an appropriate label for the RIP, e.g. **Web1**.
- Change the *Real Server IP Address* field to the required IP address, e.g. **10.0.143.221**.
- Set the *Real Server Port* field to the required port, e.g. **80**.

3. Click **Update**.
4. Repeat the above steps to add your other web server(s).

Step 4 - Apply the new Settings

To apply the new settings, HAProxy must be reloaded. This can be done using the reload button in the "Commit changes" box at the top of the screen or by using the *Restart Services* page:

1. Using the WebUI, navigate to: *Maintenance > Restart Services*.
2. Click **Reload HAProxy**.

Step 5 - Associate the VIP with an Elastic IP Address

1. Using the EC2 Management Console, allocate a new Elastic IP address.
2. Now associate this address with the VIP, in this case **10.0.8.140**.

Step 6 - Test & Verification

Browse to the Elastic IP address and verify that you are able to access the load balanced web servers.

9.3. Example 2 - Public facing Web Servers - Dual subnet, Layer 4

The load balancer is located in a public subnet and the Real Servers are located in a private subnet. For NAT mode to operate correctly, routing rules for the Real Server subnet must be changed so that return traffic passes back via the load balancer.

Step 1 - AWS Setup/Instance Deployment

1. Configure a VPC with a private and public subnet using the VPC configuration tool.
2. Deploy the load balancer instance in the public subnet as described in Deploying Enterprise AWS. Make sure you specify an appropriate IAM Role.
3. Deploy your required Real Server instances (e.g. web servers) into the private subnet.
4. If you want the Real servers to be able to access the Internet via the load balancer, please refer to NAT Gateway.
5. Disable the **Source/Destination Check** for the load balancer instance.
6. Add a default route to the private subnet's routing table, set the target to be the load balancer instance.
    - Under the VPC Management Console, select ***Route Tables***.
    - Select the route table that relates to the private subnet.
    - Select the ***Routes*** tab, and click **Edit routes**.
    - Click **Add Route**.
    - set the destination to **0.0.0.0/0**.
    - In the *Target* drop-down select **Instance**, then select the load balancer.
    - Click **Save Changes**.
7. Ensure that the Security Group(s) allow the required inbound access. The following example uses 2 security groups. **SG-1** is associated with the Real Server's ENI and **SG-2** is associated with the load balancer's ENI.

SG-1

Protocol	Port	Source	Use
TCP	80	0.0.0.0/0	web traffic / health checks
TCP	22	10.0.0.0/16	web server management ((using load balancer as jump box))

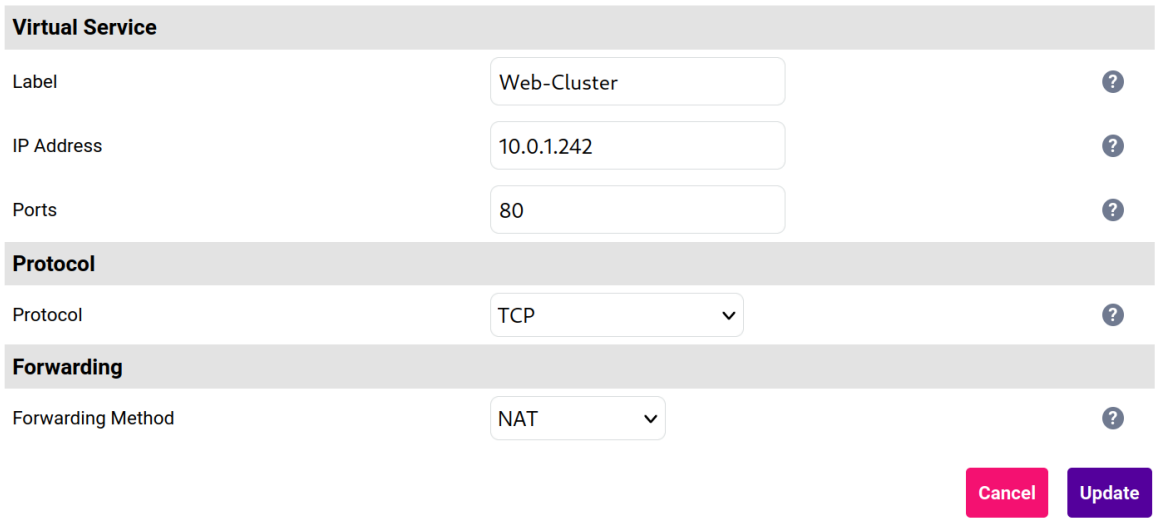
> **Note** Since layer 4 NAT mode is transparent, the source address of traffic reaching the web servers will be the requesting client. Therefore, the web traffic rule must allow traffic from the Internet.

SG-2

Protocol	Port	Source	Use
TCP	80	0.0.0.0/0	web traffic
TCP	22	management subnet	appliance management
TCP	9443	management subnet	appliance management

Step 2 - Virtual Service (VIP) Configuration

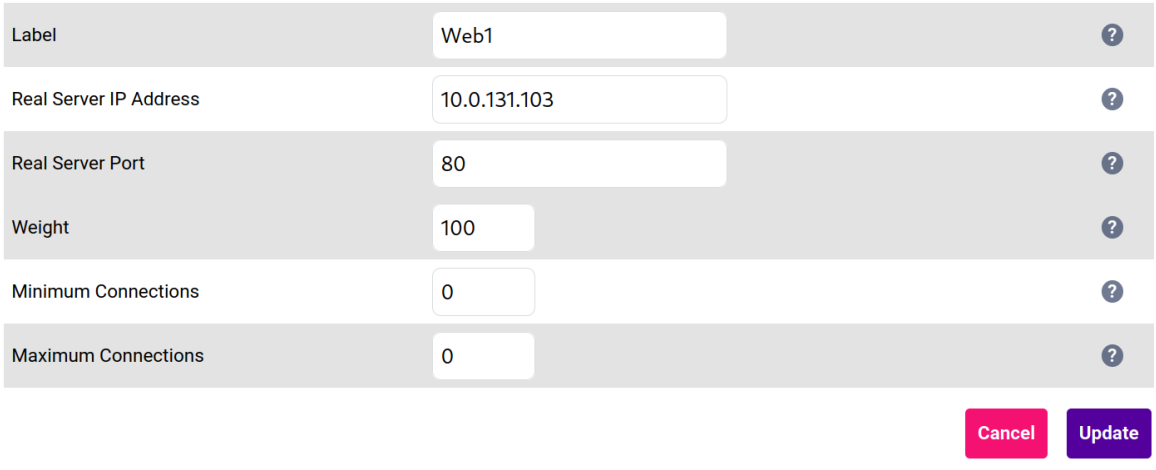
1. Using the WebUI, navigate to: *Cluster Configuration > Layer 4 - Virtual Services* and click **Add a New Virtual Service**.
2. Enter the following details:



3. Enter an appropriate label for the VIP, e.g. **Web-Cluster1**.
4. Set the *Virtual Service IP address* field to the required IP address, e.g. **10.0.1.242**.
5. Set the *Virtual Service Ports* field to the required port, e.g. **80**.
6. Leave *Protocol* set to **TCP**.
7. Click **Update**.

Step 3 - Real Server (RIP) Configuration

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



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

4. Change the *Real Server IP Address* field to the required IP address, e.g. **10.0.131.103**.
5. Set *Real Server Port* to **80**.
6. Click **Update**.
7. Repeat the above steps to add your other web servers(s).

Step 4 - Associate the VIP with an Elastic IP Address

1. Using the EC2 Management Console, allocate a new Elastic IP address.
2. Now associate this address with the VIP, in this case **10.0.8.145**.

Step 5 - Test & Verification

Browse to the Elastic IP address and verify that you are able to access the load balanced web servers.

10. Configuring HA for Enterprise AWS

10.1. Introduction

2 methods are supported:

1. Configuring HA using 2 Instances (Primary & Secondary) in a Single AZ
2. Configuring HA using 2 Instances (Primary 1 & Primary 2) Split Between 2 AZs

10.2. Configuring HA using 2 Instances (Primary & Secondary) in a Single AZ

Enterprise AWS supports our traditional 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 must be deployed in the **same Availability Zone/Subnet** to allow VIP(s) to be brought up on either appliance. For more information on how this mode works, please refer to Deployment Concepts - Single Availabilty Zone.

> **Note** Whilst HA clustering can be configured at any time, we generally recommend that the Primary appliance should be fully configured first and then the Secondary appliance should be added to create the clustered pair. Once this is done, all load balanced services configured on the Primary are automatically replicated to the Secondary using SSH/SCP. Subsequent configuration changes should be made on the Primary instance and the Secondary will then be kept in sync automatically. Pairing must be performed on the unit that is to be the Primary appliance.

> **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 - Deploy a second Load Balancer Instance

1. For more details, please refer to Accessing & Deploying the AMI.

Step 2 - Update Security Group Settings

1. Ensure that the security group(s) allow the following additional Inbound traffic for both load balancer instances. These rules are required to ensure that the HA pair can be configured and can communicate successfully.

Protocol	Port	Source	Use
TCP	9443	peer instance	HA pair configuration
UDP	6694	peer instance	HA pair communication (Heartbeat)
TCP	22	peer instance	HA pair configuration
ICMP		peer instance	HA pair configuration (ICMP Echo Request)

Step 3 - Prepare both instances for pairing

> **Note** Perform the following on **both** instances.

1. Using the WebUI, navigate to: ***Local Configuration > Security*** and change the ***Appliance Security Mode*** to **Custom**, then click **Update**.

> **Note** You may need to refresh the WebUI to ensure that the ***Execute Shell Command*** option is visible in the menu.

2. 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:

```
lb_enable_root enable
```

Step 4 - Configure the Source/Dest. Check

> **Note** This is only required if you plan to use Layer 4 NAT mode or Layer 7 SNAT mode with TProxy.

> **Note** Perform the following on **both** instances.

1. Right-click the instance and select: ***Networking > Change Source/Dest. Check*** and click **Yes, Disable**.

Step 5 - Configure the Heartbeat Failover Script

> **Note** This is only required if you plan to use Layer 4 NAT mode or Layer 7 SNAT mode with TProxy.

For Layer 4 NAT mode, or Layer 7 mode with TProxy (transparency) enabled, AWS routing rules must be

configured so that the load balancer is the default gateway. These routing rules must be changed to route via the Secondary instance when a failover occurs and again to route via the Primary when failback occurs. To set this up:

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

```
aws ec2 replace-route --route-table-id rtb-09bde5ec25c725b78 --destination-cidr-block 0.0.0.0/0
--instance-id i-06c17fefdf0efac0d --region eu-west-2
```

> **Note** Ensure that the command above is entered on a single line.

   - change **rtb-09bde5ec25c725b78** to the RT-ID of the table associated with your Real Server's subnet.
   - change **i-06c17fefdf0efac0d** to the Instance-Id of your Primary instance.
   - change **eu-west-2** to your region.

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

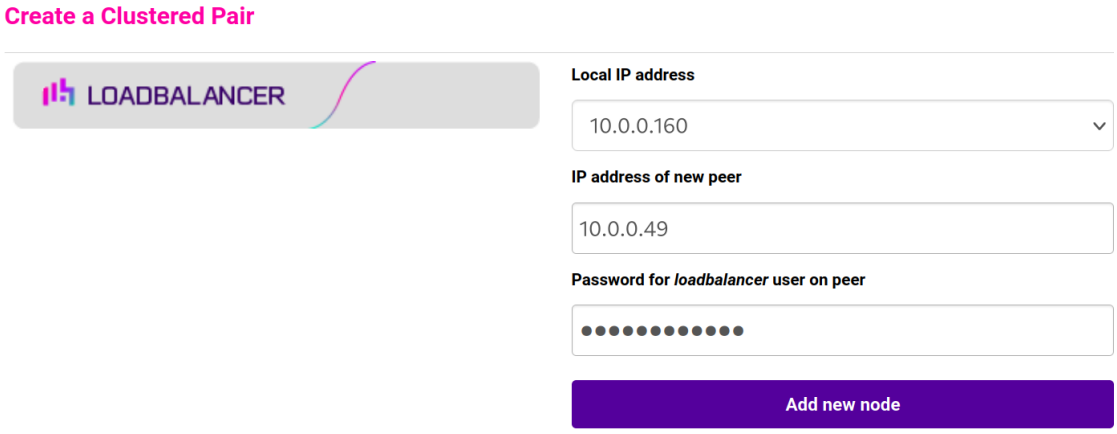
```
aws ec2 replace-route --route-table-id rtb-09bde5ec25c725b78 --destination-cidr-block 0.0.0.0/0
--instance-id i-0499a7fde15eb74f1 --region eu-west-2
```

> **Note** Ensure that the command above is entered on a single line.

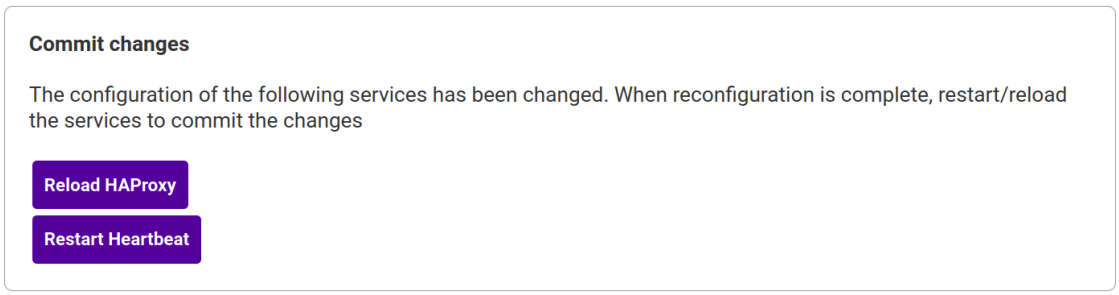
   - change **rtb-09bde5ec25c725b78** to the RT-ID of the table associated with your Real Server's subnet.
   - change **i-0499a7fde15eb74f1** to the Instance-Id of your Secondary instance.
   - change **eu-west-2** to your region.

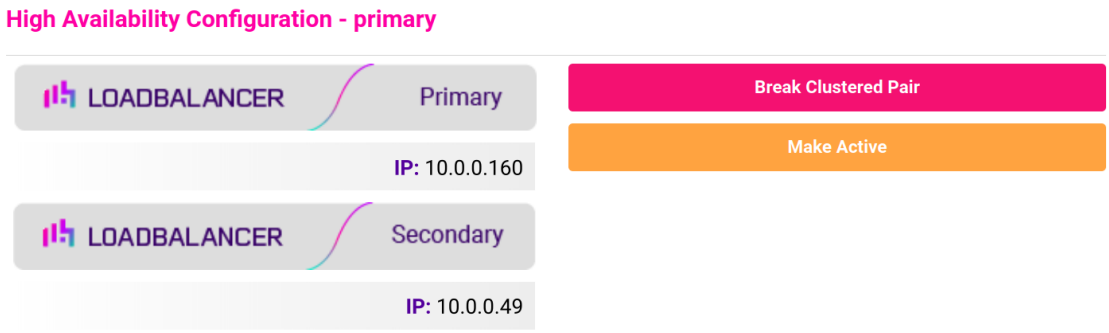
Step 6 - Configure High-Availability

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:





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

Step 7 - Verify Synchronization State

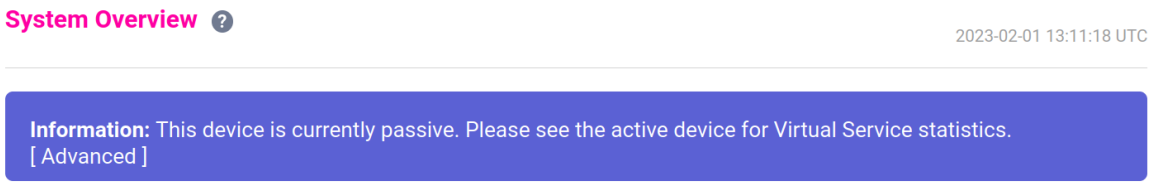
1. Once all services have restarted, the synchronization process will be complete.
2. Verify that the status on the Primary & Secondary is as follows:

Primary Unit:

Secondary Unit:

> **Note** If services have not yet been configured, "Active" and "Passive" will be grayed out on both appliances.

The Secondary can be made active by clicking **[Advanced]** in the information box that states "This device is currently passive", and then clicking the **Take over** button:



> **Note** For other possible states, please refer to Clustered Pair Diagnostics.

10.3. Configuring HA using 2 Instances (Primary 1 & Primary 2) Split Between 2 AZs

Enterprise AWS also supports a second HA method where each instance is deployed in a different AZ. 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 Availabilty Zones.

> **Note** This procedure assumes the first appliance is already up and running and that it will be Primary 1.

Step 1 - Deploy a second Load Balancer Instance

1. Deploy a second load balancer in a different AZ/subnet, this will be the Primary 2 instance. For more information, please refer to Accessing & Deploying the AMI.

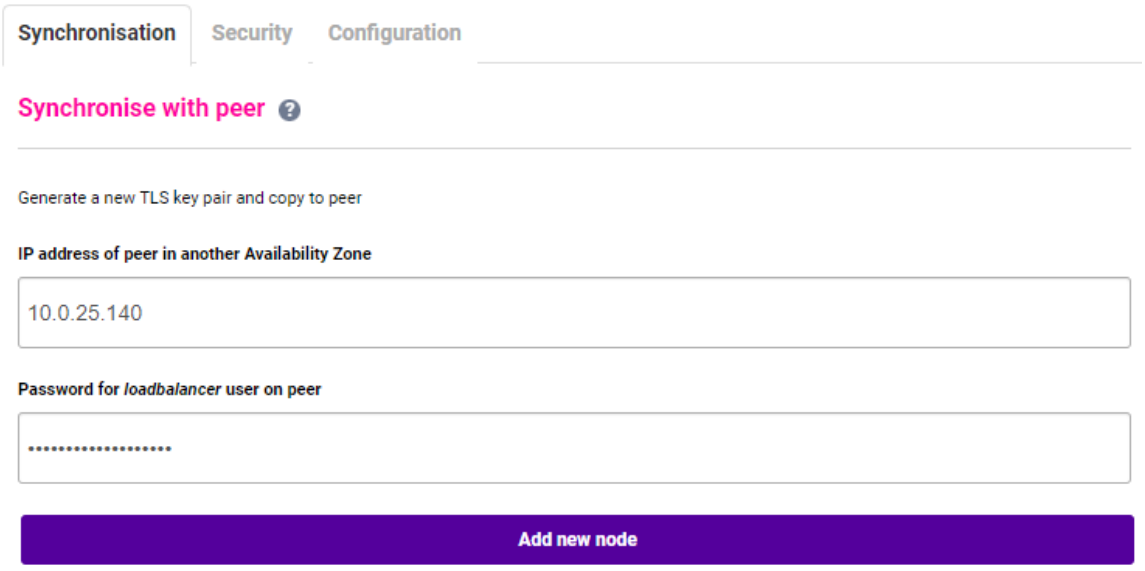
Step 2 - Update Security Group Settings

1. Ensure that the security group(s) allow the following additional Inbound traffic for both load balancer instances. These rules are required to ensure that the HA pair can be configured and can communicate successfully.

Protocol	Port	Source	Use
TCP	9443	peer instance	AZ HA configuration
TCP	9444	peer instance	AZ HA communication

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

1. Using the WebUI of the Primary 1 instance, navigate to: *EC2 Configuration > EC2 Zone HA Configuration* and select the *Synchronization* Tab.



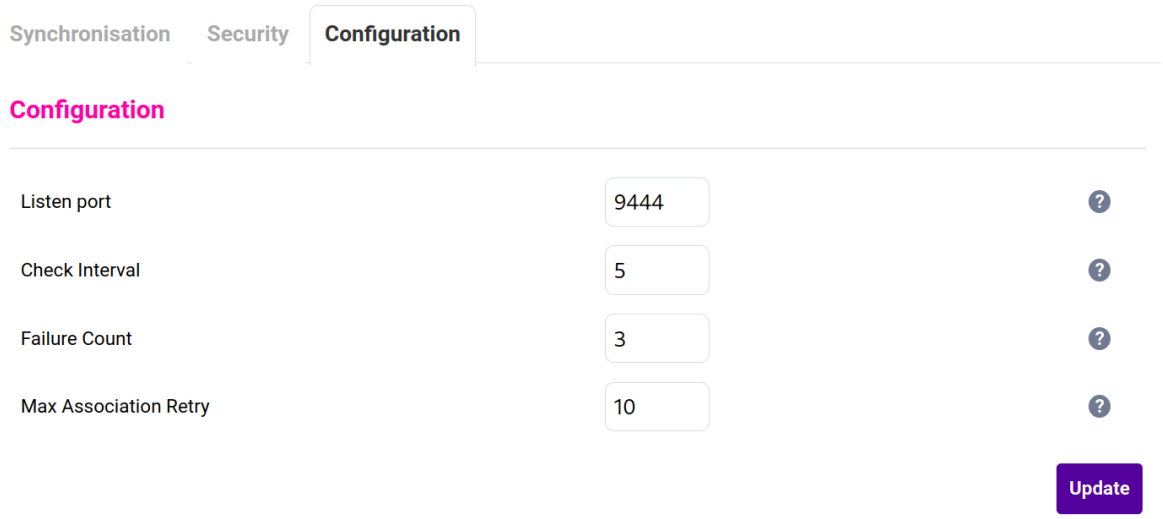
   - Enter the private IP address and **loadbalancer** user password for the second (Primary 2) instance. By default this is the EC2 Instance-ID.
   - Click **Add new node**.
   - A new Keypair & associated certificates will be generated and copied to the second instance.

> **Note** These can be viewed and also deleted if required using the *Security* tab on the *EC2 Zone HA Configuration* WebUI menu option on each appliance.

Step 4 - Review Zone HA Settings

The TCP port, check interval and failure count settings can be changed if needed. To view/update the check parameters:

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



> **Note** The default values work well in most situations. If these need to be changed, make the same changes on both instances.

Step 5 - Configure corresponding VIP(s) on the Second Instance (Primary 2)

1. Corresponding VIP(s) must be configured on the second instance (Primary 2). The EIP(s) will be associated with these VIP(s) if failover from the first instance (Primary 1) occurs. In this example, the following Real Servers are configured for the VIP on both Primary 1 and Primary 2 instances:

Real Server Name	Real Server IP	AZ
Server 1	10.0.128.100/20	eu-west-2a
Server 2	10.0.128.101/20	eu-west-2a
Server 3	10.0.144.110/20	eu-west-2b
Server 4	10.0.144.111/20	eu-west-2b

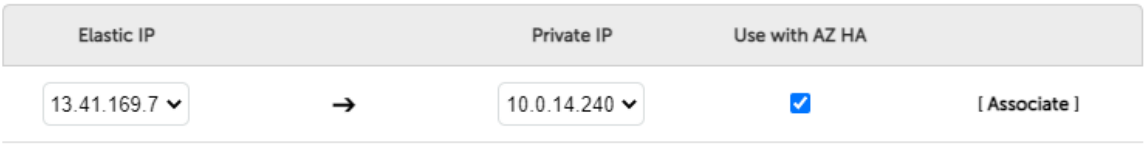
   Under normal circumstances, whether the EIP is associated with the VIP on Primary 1 or the VIP on Primary 2, all four Real Servers will be available.

   If one of the AZs fails, there are still two operational Real Servers in the remaining AZ.

   If the instance with the active EIP association(s) is located in the failed AZ, this will be detected by the peer instance in the remaining AZ and the EIP(s) will be automatically re-associated.

Step 6 - Associate EIPs on the First Instance (Primary 1)

1. On the Primary 1 instance, using the WebUI navigate to: *EC2 Configuration > EC2 Network Configuration*.
2. Select the required EIP in the first drop-down and the relevant VIP address in the second drop-down.

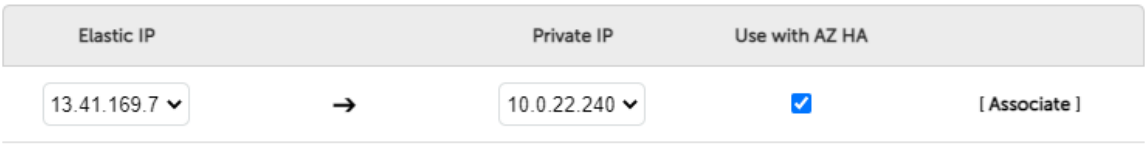


- Enable (check) the **Use with AZ HA** checkbox.
- Click **[Associate]**.
- Repeat for any other EIPs/VIPs.

> **Note** The load balancer instance where the EIP association is first made will be the "active" appliance.

Step 7 - Associate EIPs on the Second Instance (Primary 2)

1. On the Primary 2 instance, using the WebUI navigate to: *EC2 Configuration > EC2 Network Configuration*.
2. Select the same EIP that was selected in **Step 6**.
3. Select the VIP that corresponds to the VIP that was selected in **Step 6**.



- Enable (check) the **Use with AZ HA** checkbox.
- Click **[Associate]**.
- Repeat for all other EIPs/VIPs.

Step 8 - Configure Failover Scripts (If Required)

> **Note** Typically, this is only required if you are using Layer 4 NAT mode or Layer 7 SNAT with with transparency (TProxy) where the routing table for the Real Server subnet(s) need to be modified at failover.

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.

Step 9 - Checking EIP Status

1. On the first instance, using the WebUI, navigate to: *EC2 Configuration > EC2 Zone HA Status*.
2. The current status will be displayed:

- The EIP status is **Local**, i.e. the EIP is associated with the VIP on ***this*** instance.

3. On the second instance, using the WebUI, navigate to: *EC2 Configuration > EC2 Zone HA Status*.
4. The current status will be displayed:

- The EIP status is **Peer**, i.e. the EIP is associated with the VIP on the ***other*** instance.

Step 10 - 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.

Forcing EIP failover

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. The **[Associate]** option forces an immediate EIP re-association.

11. Testing & Verification

For additional guidance on diagnosing and resolving any issues you may have, please also refer to Diagnostics & Troubleshooting.

12. Monitoring

12.1. Using Appliance Reports & Log Files

The appliance includes several logs and reports that are very useful when diagnosing issues.

For more details on accessing and interpreting appliance logs, please refer to Diagnostics & Troubleshooting.

For more details on accessing and interpreting appliance reports & graphs, please refer to Monitoring & Reporting.

12.2. AWS Cloudwatch

Cloudwatch basic monitoring can be used at no charge. More detailed monitoring can be enabled if required. For more details on basic and detailed monitoring, please refer to Basic monitoring and detailed monitoring.

For details of how to enable detailed monitoring, please refer to Enable or turn off detailed monitoring for your instances.

12.3. AWS Service Limits

AWS maintains service quotas (formerly called service limits) for each account to help guarantee the availability of AWS resources and prevent accidental provisioning of more resources than needed. For more details, please refer to Managing AWS Service Quotas.

13. Backup & Restore

13.1. Appliance

The appliance includes comprehensive backup and restore functionality that allows the full configuration to be backed up and restored. For full details, please refer to Backup & Restore.

13.2. AWS

AWS Backup provides comprehensive backup & restore features that allows the complete load balancer instance to be backed up and restored. For full details, please refer to What is AWS Backup?.

14. More Information

Please refer to our website for all the latest Manuals and Deployment Guides.

15. Loadbalancer.org Technical Support

Our highly experienced Support Engineers are on hand to help 24 hours a day, 365 days a year.

15.1. Contacting Support

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

LOADBALANCER

Visit us: www.loadbalancer.org

Phone us: +44 (0)330 380 1064

Phone us: +1 833 274 2566

Email us: info@loadbalancer.org

Follow us: @loadbalancer.org

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.