CumuLogic Load Balancer Overview Guide

March 2013

Table of Contents

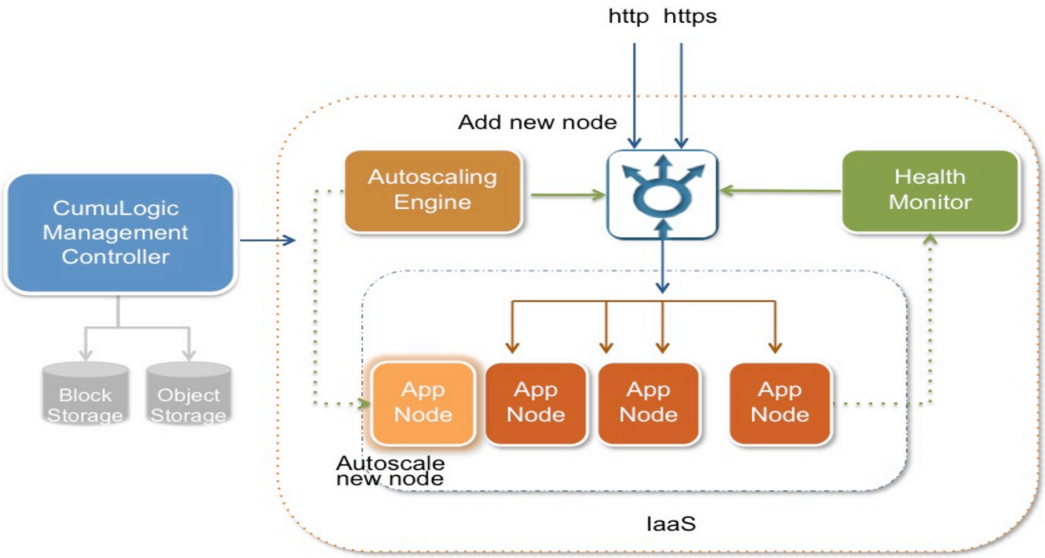
CumuLogic Load Balancer

CumuLogic Load Balancer provides fully managed instances of load balancer on any Infrastructure-as-a-Service (IaaS) cloud. CumuLogic Load Balancer gives you full access to the functionality and performance of Nginx servers. CumuLogic Load Balancer instances are fully managed and monitored, eliminating the need for manual configuration. Each load balancer instance can automatically detect the cluster nodes it load balances and adjusts its configuration for optimal performance. Users can provision load balancer instances on any IaaS cloud using HTTP API, or the easy-to-use CumuLogic User Console.

CumuLogic Load Balancer is based on the open source Ngnix load balancer and supports Layer-7 HTTP and HTTPS protocols.

Architectural Overview of CumuLogic Load Balancer

The following diagram shows how the Load Balancer works with various components of the CumuLogic Platform.



1. The client sends an HTTP/HTTPS request to the load balancer to access your application.
2. The load balancer determines where the request should be rerouted depending on the configuration. The default routing algorithm used is round-robin.
3. The load balancer keeps track of the status of all the registered application instances and the nodes are marked inactive if they do not meet the threshold defined in the health check configuration.
4. The load balancer routes the client request to the identified healthy application instance. At this point, the client is communicating with one of the application instances through the load balancer. The load balancer listeners can be configured to use either HTTP, HTTPS for both front-end connection (client to load balancer) and back-end connection (load balancer to back-end instance).

How to Use CumuLogic Load Balancer

You can use a load balancer instance to configure and load balance application instances deployed on any cloud. CumuLogic Load Balancer is fully managed and is application node-aware, automatically detecting node failures and reconfiguring routing to avoid routing traffic to failed nodes.

By default, a load balancer instance is configured to load balance applications deployed on CumuLogic PaaS.

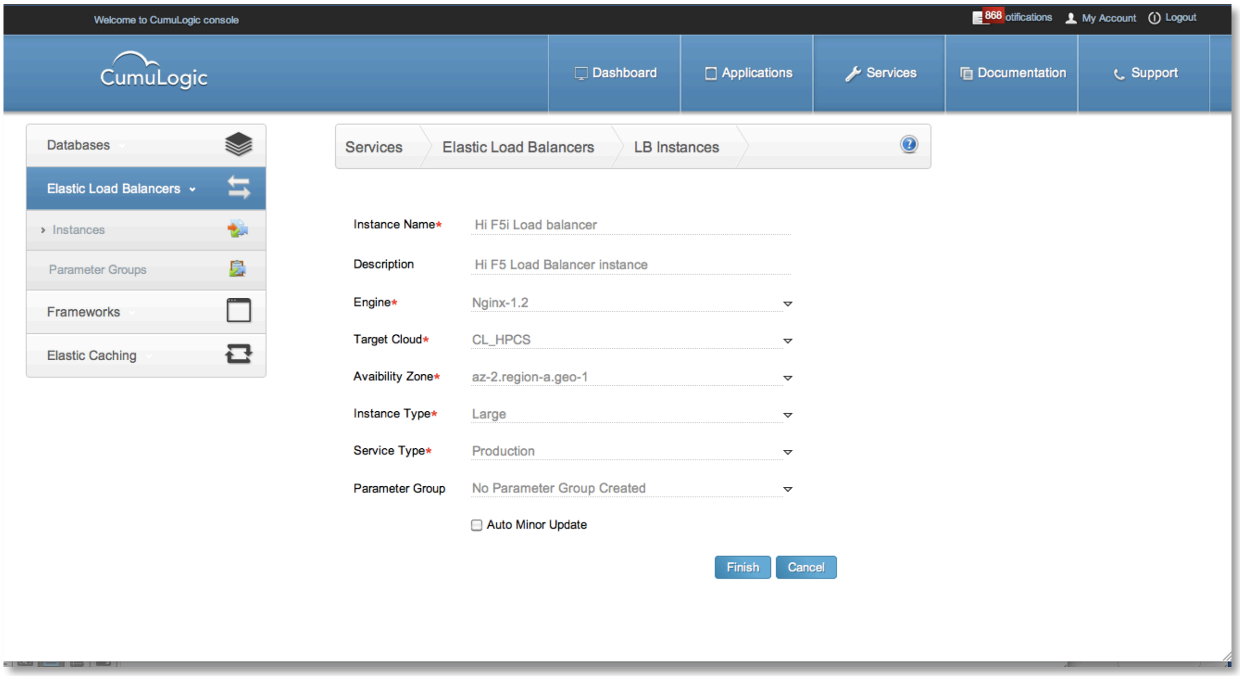
CumuLogic Load Balancer can also be configured to route traffic to additional new nodes added to the application. When an application deployed on CumuLogic PaaS autoscales the application server or framework nodes, CumuLogic PaaS automatically reconfigures the load balancer to route traffic to new nodes as well.

CumuLogic Load Balancer is also monitored and has self-healing capabilities to recover failures.

Creating a Load Balancer Instance

CumuLogic Load Balancer can be launched using the User Console or a simple set of API.

To create a new instance of the load balancer, select **Load Balancer** from menu bar on the left and choose the size of the load balancer instance as shown below.



Instance Name* Hi F5i Load balancer

Description Hi F5 Load Balancer instance

Engine* Nginx-1.2

Target Cloud* CL_HPCS

Avaibility Zone* az-2.region-a.geo-1

Instance Type* Large

Service Type* Production

Parameter Group No Parameter Group Created

□ Auto Minor Update

Figure 1: Creating a Load Balancer Instance

Select the target cloud, availability zone, size of the instance and launch your load balancer. You can also configure the load balancer parameters (see Parameter Groups for details), customize access control (see Access Groups) and customize the preferences for maintenance time windows.

Once you have launched a load balancer instance, you can configure it to load balance existing applications/nodes. You need to provide application context, HostName/IP and port numbers of the nodes to load balance.

Load Balancer Listener Configurations

The following table summarizes the listener settings that you can use to configure CumuLogic Load Balancer.

Use Case	Protocol	Optional config	Notes
Basic HTTP Load Balancer	HTTP	- Backend Port - Application context	
Secure Load balancing using SSL	HTTPS	- Backend port - Custom Cipher setting (coming soon)	- Default ciphers HIGH:!aNULL:!MD5; - SSLv2, SSLv3, TLSv1 protocols

X-Forwarded-For Support

The X-Forwarded-For request header helps you identify the IP address of a client. Because load balancers intercept traffic between clients and servers, your server access logs contain only the IP address of the load balancer. To see the IP address of the client, use the X-Forwarded-For request header.

Elastic load balancer stores the IP address of the client in the X-Forwarded-For request header and passes the header along to your server.

The load balancer comes with following default X-Forwarded-For configuration.

```
proxy_set_header   Host $host;
proxy_set_header   X-Real-IP         $remote_addr;
proxy_set_header   X-Forwarded-For   $proxy_add_x_forwarded_for;
proxy_set_header   X-Forwarded-For $remote_addr;
```

HTTPS Support

CumuLogic Load Balancer provides HTTPS support allowing you to use the SSL/TLS protocol for encrypted connections. This enables traffic encryption between the clients that initiate HTTPS request and your load balancer.

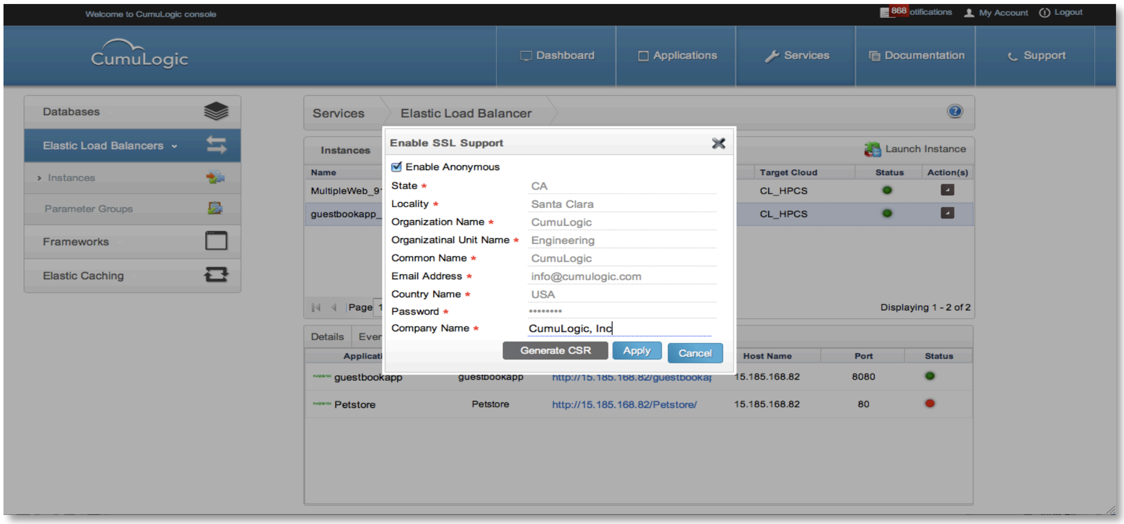
To enable HTTPS support for your load balancer, you'll have to install an SSL server certificate on your load balancer. The load balancer uses the certificate to terminate and then decrypt requests before sending them to the application instances.

Enabling SSL on the Load Balancer

You can enable SSL on the load balancer instance by selecting **Enable SSL** from the action menu on the running load balancer instance.

You can enable SSL on CumuLogic Load Balancer by selecting the "Enable SSL" action. You can enable SSL with self-signed certificate or CA certificates.

To enable self-signed certificate, just click on "Enable Anonymous," otherwise you can generate and download a CSR will help you create a load balancer using the CumuLogic Service Dashboard



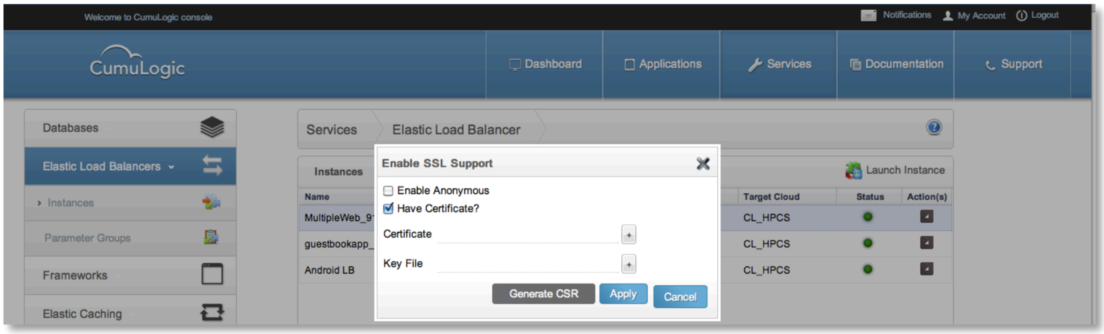
Enable SSL Support

☑ Enable Anonymous

State * CA
Locality * Santa Clara
Organization Name * CumuLogic
Organizatinal Unit Name * Engineering
Common Name * CumuLogic
Email Address * info@cumulogic.com
Country Name * USA
Password * ••••••••
Company Name * CumuLogic, Inc

Figure 2: Enabling SSL with self-signed certificate

To enable SSL with a Root CA certificate, you can select "have CA Certificate" from the Enable SSL screen and upload the certificate files.



☐ Enable Anonymous
☑ Have Certificate?

Certificate

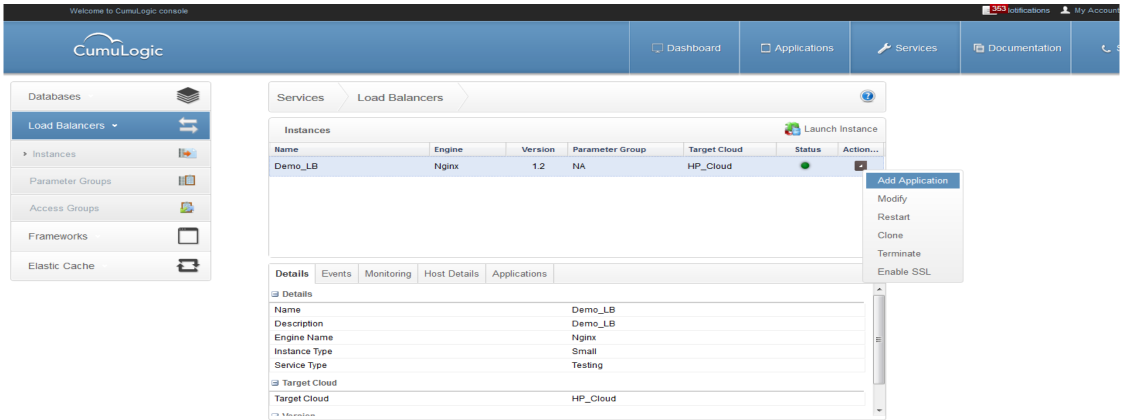
Key File

Name	Target Cloud	Status	Action(s)
MultipleWeb_9	CL_HPCS	●	
guestbookapp_	CL_HPCS	●	
Android LB	CL_HPCS	●	

Figure 3: Enabling SSL with CA certificate

Configuring the Load Balancer for Applications

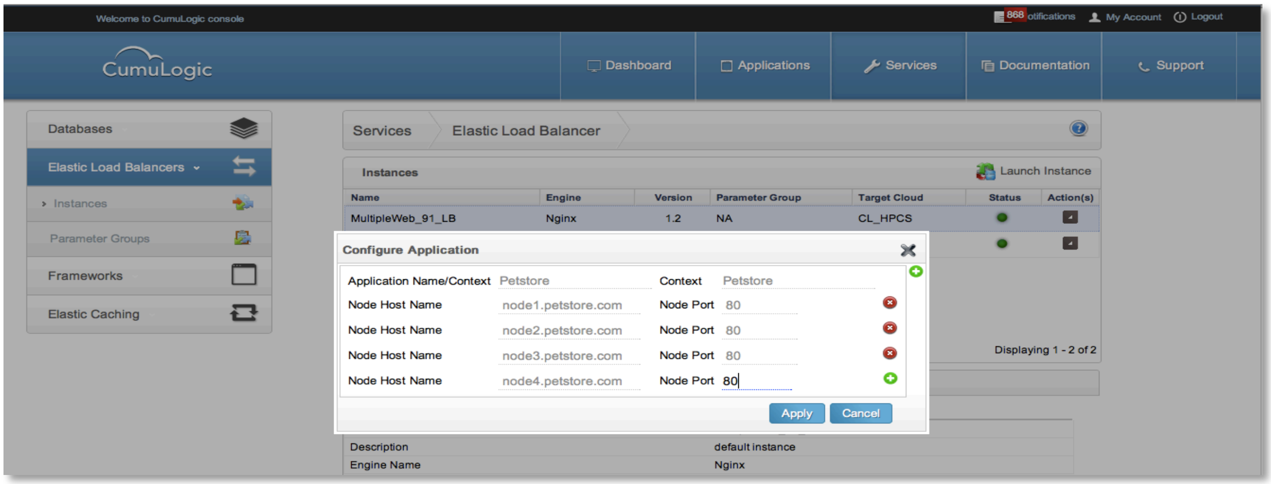
To configure the load balancer instance to load balance between multiple nodes of an application, select the **Add Application** option from the action menu.



Name	Engine	Version	Parameter Group	Target Cloud	Status	Action...
Demo_LB	Nginx	1.2	NA	HP_Cloud	●	

Details	
Name	Demo_LB
Description	Demo_LB
Engine Name	Nginx
Instance Type	Small
Service Type	Testing
Target Cloud	
Target Cloud	HP_Cloud

Figure 4 Configuring the Load Balancer for Applications



Instances

Name	Engine	Version	Parameter Group	Target Cloud	Status	Action(s)
MultipleWeb_91_LB	Nginx	1.2	NA	CL_HPCS	●	
					●	

Displaying 1 - 2 of 2

Configure Application

Application Name/Context	Petstore	Context	Petstore
Node Host Name	node1.petstore.com	Node Port	80
Node Host Name	node2.petstore.com	Node Port	80
Node Host Name	node3.petstore.com	Node Port	80
Node Host Name	node4.petstore.com	Node Port	80

Description	default instance
Engine Name	Nginx

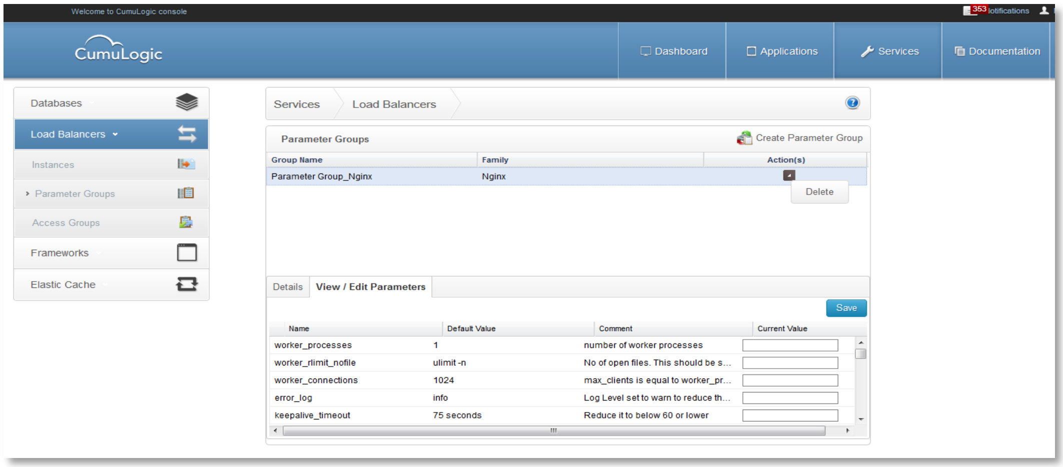
Figure 5: Configuring a load balancer instance

As shown in the screen above, you can add host names or IP addresses of nodes used for running the application. In this example, the load balancer is configured to route traffic to the petstore sample application, which runs on four nodes in this demo.

Load Balancer Parameter Groups

Parameter groups allow users to optimize the performance of the load balancer for specific workloads. You can easily create a new parameter group from a parameter family group of the engine and modify the selected parameters.

You can apply a parameter group to running instances or start a new load balancer instance with new parameter group.



Group Name	Family	Action(s)
Parameter Group_Nginx	Nginx	

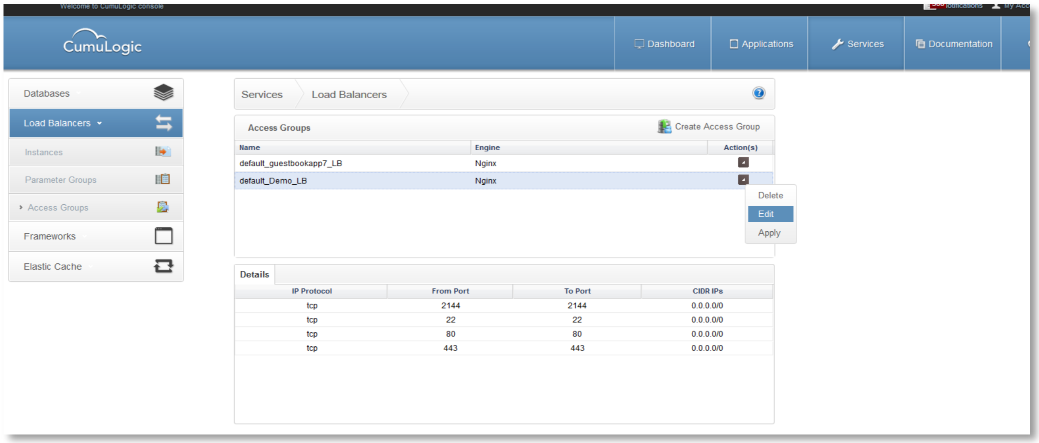
Name	Default Value	Comment	Current Value
worker_processes	1	number of worker processes	
worker_rlimit_nofile	ulimit -n	No of open files. This should be s...	
worker_connections	1024	max_clients is equal to worker_pr...	
error_log	info	Log Level set to warn to reduce th...	
keepalive_timeout	75 seconds	Reduce it to below 60 or lower	

Figure 6: View/Edit Parameter Group

Load Balancer Access Groups

You can restrict access to your load balancer by configuring the Access Group. You can use an existing Access Group or create a new Access Group specific to a Load Balancer.

From the Access Group screen click on "Access Group" to see current configuration details. You can select desired action from the menu. You can Edit/Delete Access Group.

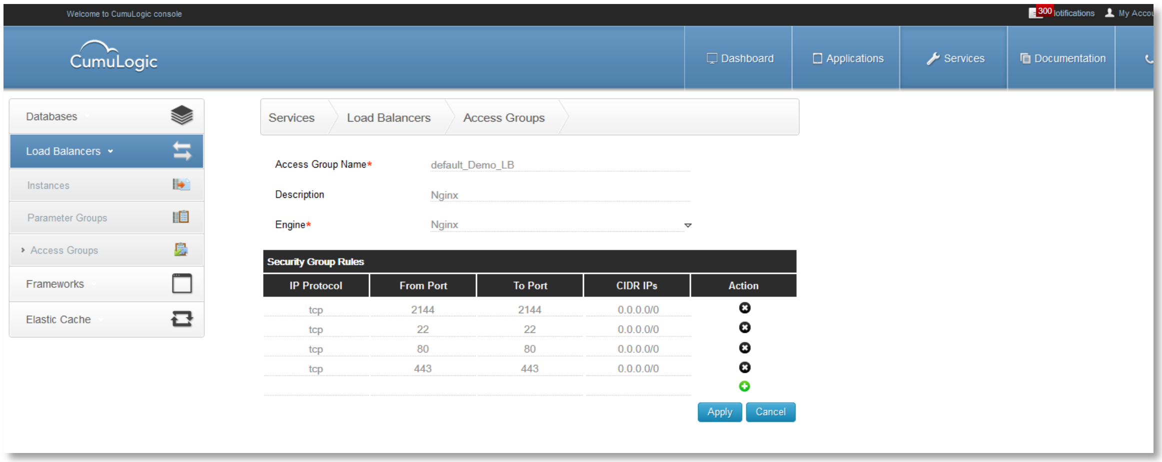


Name	Engine	Action(s)
default_guestbookapp7_LB	Nginx	
default_Demo_LB	Nginx	

IP Protocol	From Port	To Port	CIDR IPs
tcp	2144	2144	0.0.0.0/0
tcp	22	22	0.0.0.0/0
tcp	80	80	0.0.0.0/0
tcp	443	443	0.0.0.0/0

Figure 7: Viewing Access Group

To edit Access Group select "Edit" action. You can define the port range and cidr details to open access to specific ports for specific cidr range.



Access Group Name* default_Demo_LB

Description Nginx

Engine* Nginx

Security Group Rules

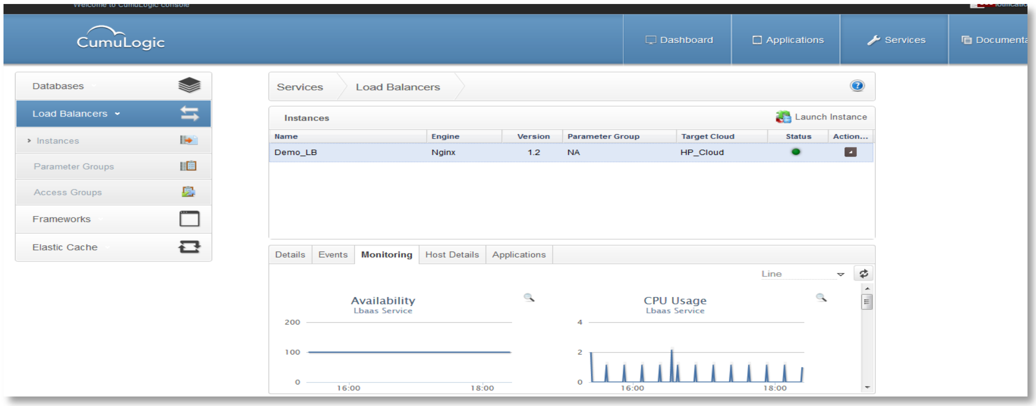
IP Protocol	From Port	To Port	CIDR IPs	Action
tcp	2144	2144	0.0.0.0/0	⊗
tcp	22	22	0.0.0.0/0	⊗
tcp	80	80	0.0.0.0/0	⊗
tcp	443	443	0.0.0.0/0	⊗
				⊕

Figure 8: Editing Access Group

After Editing the Access Group, you can select Apply to apply the changes to Load Balancer.

Monitoring Load Balancer

Performance metrics are available under the "Monitoring" tab.

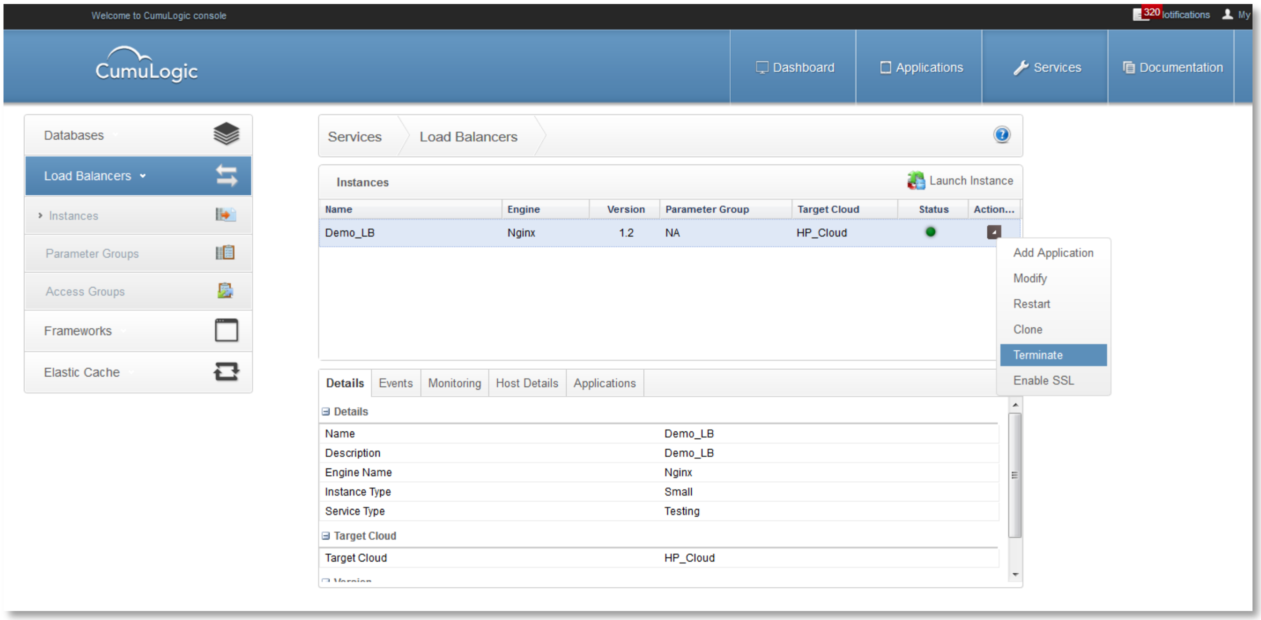


Name	Engine	Version	Parameter Group	Target Cloud	Status	Action...
Demo_LB	Nginx	1.2	NA	HP_Cloud	●	

Figure 9: Performance metrics

Terminating the Load Balancer

To terminate the load balancer instance, select "Terminate" from the available actions, which will prompt you for confirmation. Once confirmed, the load balancer instance will be terminated along with all configurations.



Name	Engine	Version	Parameter Group	Target Cloud	Status	Action...
Demo_LB	Nginx	1.2	NA	HP_Cloud	●	

⊟ Details

Name	Demo_LB
Description	Demo_LB
Engine Name	Nginx
Instance Type	Small
Service Type	Testing

⊟ Target Cloud

Target Cloud	HP_Cloud

Figure 10: Terminating a Load Balancer Instance

Load Balancer APIs

APIs allow users to control and optimize a load balancer instance for their specific workloads. Below is a list of currently supported APIs.

Request	URI	Description
GET	/services/loadbalancers	List all Load balancers. Response returns the list of load balancers including loadbalancerId, load balancer Name, Status and Creation date
GET	/services/loadbalancers/loadbalancerId	List details of selected Load Balancer. The response includes Id, name, Created Date, Status, protocols supported, port number, algorithms, list of current nodes and VIP.
DELETE	/services/loadbalancers/loadbalancerId	Deletes the load balancer instance
POST	/services/loadbalancers/create	Creates a new load balancer instance with options provided. Returns details of the load balancer instance, load balancer Id and all options
POST	/services/loadbalancers/loadbalancerId/nodes	Adds a node to load balancer. Request provides node IP address, port number, node name
DELETE	/services/loadbalancers/loadbalancerId/nodes	Deletes node load balanced by the load balancer. Request provides node IP address or node name
PUT	/services/loadbalancers/loadbalancerId/nodes	Updates a node to load balancer. Request provides node IP address, port number, node name

Table 1