IMPLEMENTACIÓN Y MONITORIZACIÓN DE UN CLÚSTER DE SERVIDORES WEB LINUX CON BALANCEO DE CARGA

Autor: José Cristian Cañaveras Vélez

OBJETIVOS

- Implementar una infraestructura Web de Alta disponibilidad tolerante a fallos en los servidores.
- Disponer de la información replicada obteniendo velocidad en la respuesta de las peticiones y seguridad.
- Reducir la carga de los servidores repartiendola entre los clústers disponibles.
- Monitorización del clúster de manera sencilla y notificación ante fallos.

RECURSOS USADOS

Equipo Portátil (real):

- Intel Core I7-4710HQ
- Memoria RAM: 16GB
- Disco Duro m-SATA SSD 250GB
- Disco Duro SATA 1TB
- Disco Duro SATA 750GB
- S.O: Windows 7 Professional x64 (VirtualBOX y GNS3)

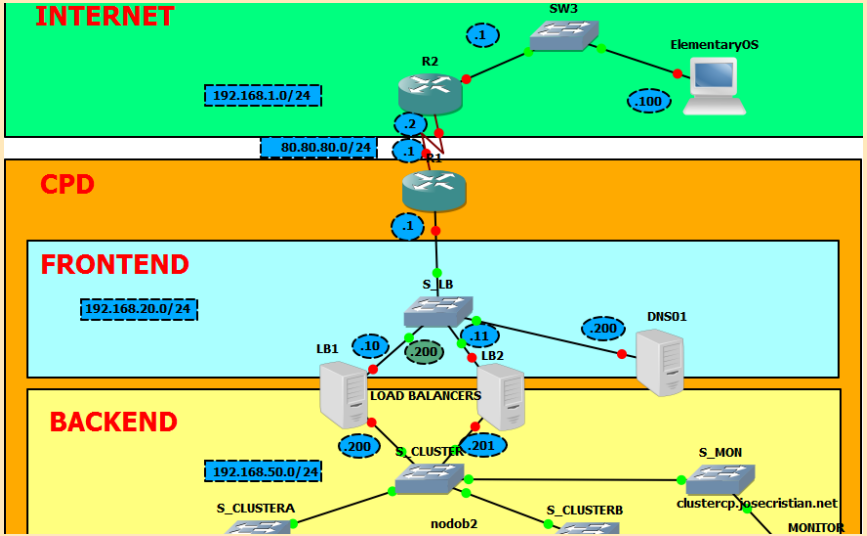
Sistemas Operativos (virtualizados con VirtualBOX):

- CentOS 7 x64
- Ubuntu Server 14.04.2 x64
- Debian 8.0 x64
- ElemetaryOS freya x64

Balanceadores de carga, ¿Qué es?

Balanceo de Carga

- El balanceo de carga consiste repartir las peticiones de los usuarios entre diferentes servidores mediante un algoritmo cuya función es controlar como se balancean dichas peticiones.

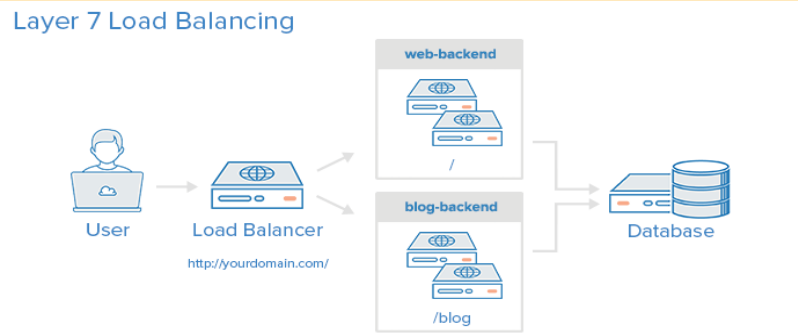


Balanceadores de carga. Tipos (I)

En función de la capa de red del modelo OSI que se use:

- Capa 7 (Aplicación)

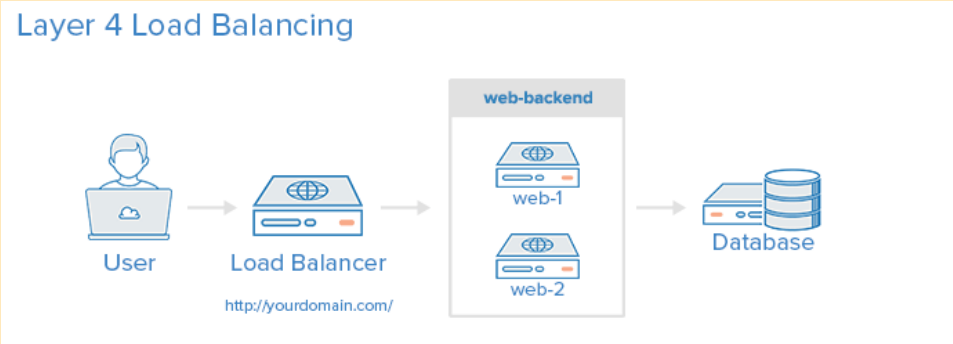
La función de este tipo de balanceo es la de reenviar las peticiones de los usuarios en función del contenido de la solicitud del usuario observando la cabecera.



Balanceadores de carga. Tipos (II)

- Capa 4 (Transporte)

La función de este tipo de balanceo es la de reenviar las peticiones de los usuarios en función del dominio, subdominio o puerto que los usuarios soliciten, **este tipo de balanceo será el que usaré**.



Balanceadores de carga. Algoritmos

- En función del modo en que repartan la carga existen diferentes algoritmos:

- **RoundRobin.** Reparte las peticiones a medida que van llegando entre los servidores disponibles.
- **LeastConn.** Reparte las conexiones al servidor con menos conexiones en el momento de realizar al balanceo.
- **Source.** Reparte las peticiones en función del origen, es decir, la dirección IP de origen, para asegurar que un mismo usuario siempre se conecta al mismo servidor o clúster.

Balanceadores de carga. HAproxy (I)

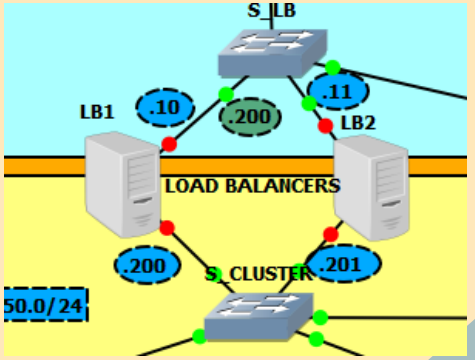


- HAproxy es un software balanceador de carga TCP/HTTP que reparte la conexiones entre los servidores disponibles actuando como proxy inverso.

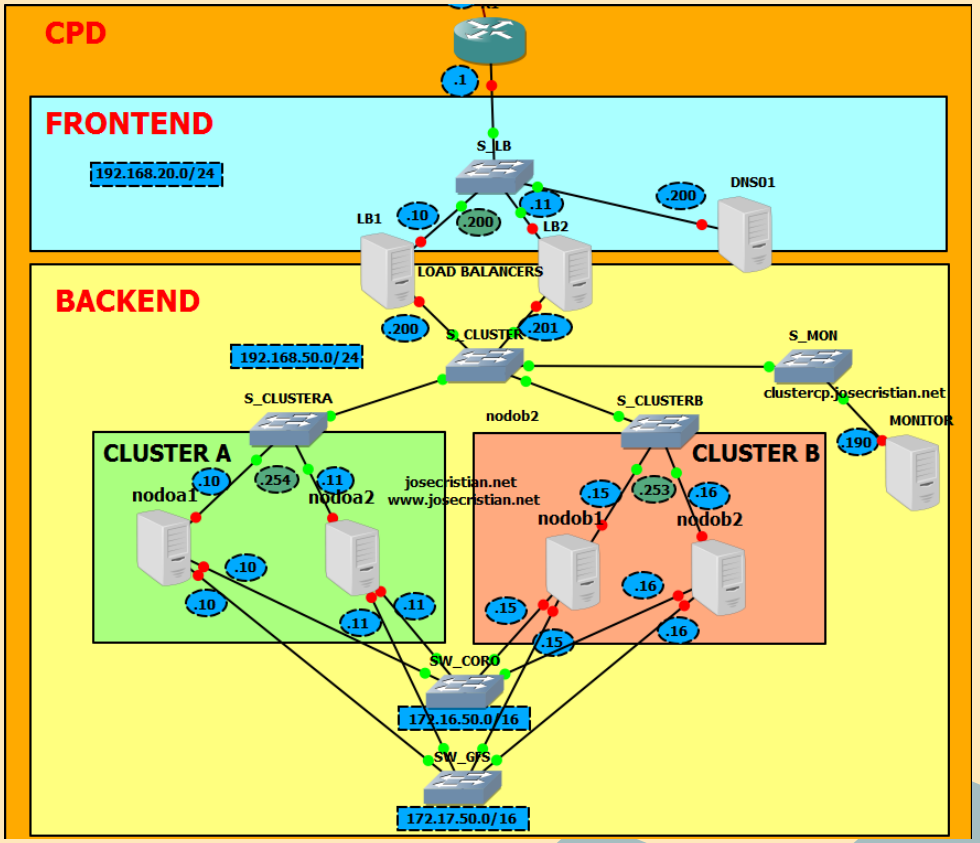
Balanceadores de carga. HAproxy (II)

KeepAlived

- **KeepAlived** es un software que nos permite ofrecer una dirección IP virtual, en el caso de los balanceadores de carga permite ofrecer una dirección IP compartida en ambos balanceadores, de tal manera que cuando el usuario accede a la dirección IP accede realmente a uno de los servidores, si por algún motivo ese servidor fallase automáticamente la dirección IP compartida se pasaría al otro servidor de manera transparente para el usuario.

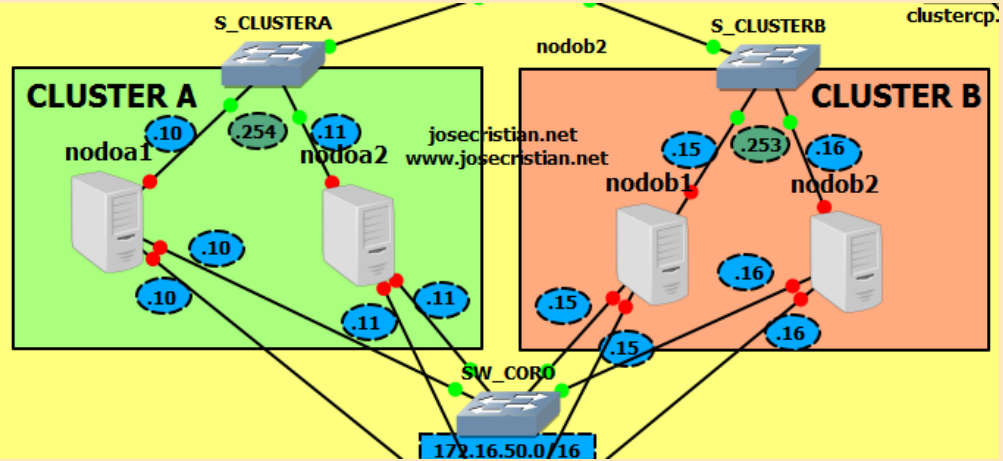


Balanceadores de carga. HAproxy (III)



CLÚSTER. ¿Qué es?

- Un clúster es un conjunto de 2 o mas máquinas cuya función es mantener uno o varios servicios de forma compartida operando de forma conjunta como un único recurso y estar monitorizándose constantemente.



CLÚSTER. Componentes (I)

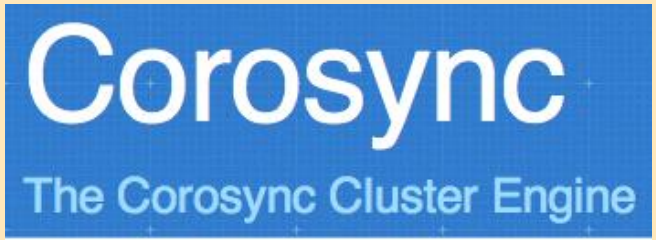
- Un clúster se compone de 3 componentes software, en mi caso he optado por los siguientes:

- Corosync
- Pacemaker
- PCS

CLÚSTER. Componentes (II)

Corosync

- La función de corosync es la de proporcionar la comunicación entre los nodos e informar de errores, de tal manera que si uno de los nodos falla es el servicio encargado de informar al otro nodo/s del error.



CLÚSTER. Componentes (III)

Pacemaker

- La función de Pacemaker es la de proporcionar el control de recursos en cada nodo del clúster para en caso de notificación de corosync se asigne el recurso a otro nodo disponible en ese momento.



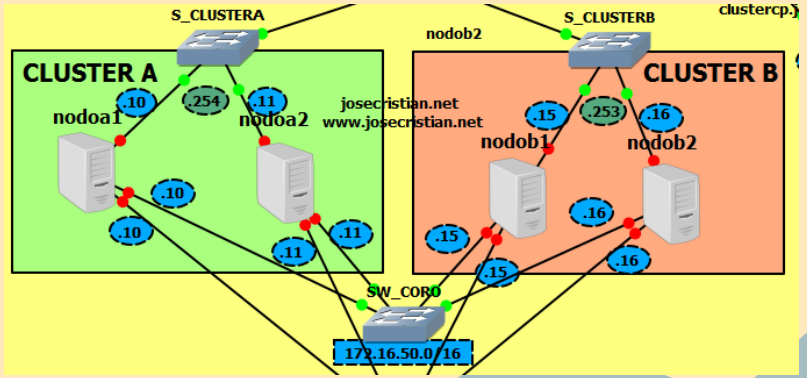
CLÚSTER. Componentes (IIII)

PCS

La función de PCS es proporcionar la autenticación entre los nodos y el mecanismo de control, también permite el arranque, parada, etc. del clúster

CLÚSTER. Componentes (IIII)

- En mi caso dispongo de 2 clústers, cada uno de ellos con 2 nodos, cada clúster es independiente del otro, por lo que pueden fallar hasta 3 nodos para que el sistema siga funcionando, **cada clúster** tiene una dirección IP flotante (compartida) entre los nodos del mismo cluster, por otro lado el cambio de recurso cuando un nodo falla se realiza a traves de otra red (independiente a la de la direccion IP) de tal manera que no consumimos el ancho de banda disponible y conseguimos mayor seguridad.

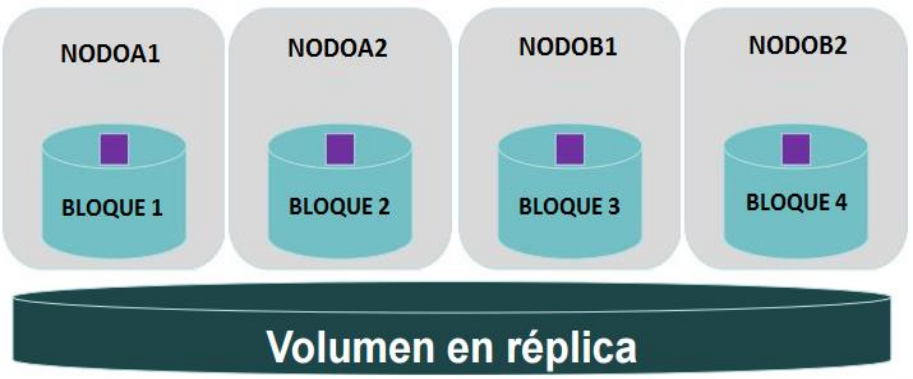


Replicación de datos.

Para asegurar una alta disponibilidad y rapidez en la presentación de los datos al usuario se ha implementado un sistema de replicación de datos, es decir, como si se tratara de un RAID-1 a nivel de Red, para ello se ha usado la solucion **GlusterFS.**

Replicación de datos. GlusterFS (I)

GlusterFS es un sistema de archivos que ofrece almacenamiento a gran escala de forma replicada ofreciendo alta disponibilidad y escalabilidad.



Replicación de datos. GlusterFS (II)

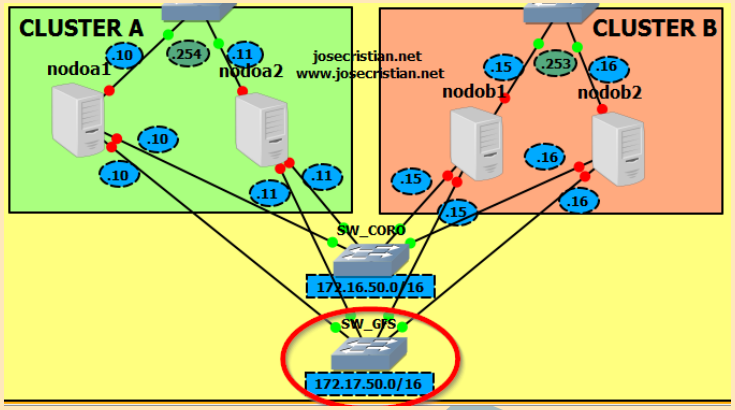
En mi caso he replicado un disco duro en cada servidor, de tal manera que el segundo disco duro de cada nodo tendrá el mismo contenido, si hay algún cambio en el disco un nodo automáticamente se replicará a los demás bloques o discos.

Replicación de datos. GlusterFS (III)

- GlusterFS permite, a diferencia de DRBD, mas escalabilidad, replicación a nivel de bloques, compatible con NFS y FUSE (cliente nativo de GlusterFS), permite la replicación de mas de 2 nodos.
- Por el contrario DRBD únicamente permite un máximo de 2 nodos para la replicación, si añadimos un tercero este se comportara como «backup» de los anteriores y no como una réplica

Replicación de datos. GlusterFS (IIII)

Para la implementación de GlusterFS hemos creado una red exclusivamente para la replicación de los datos y así disponer de todo el ancho de banda disponible para esta tarea y por seguridad.



DNS

Un servidor DNS permite traducir nombres de dominio en direcciones IP y viceversa, por lo que cuando un usuario introduzca el dominio http://josecristian.net **(o www)** el servidor DNS le redirigirá a la **dirección IP virtual** del balanceador de carga y este será el encargado de redirigirla a uno de los clústers en función del algoritmo configurado (RoundRobin) y la ACL.
Sin embargo si el usuario introduce la dirección **http://clustercp.josecristian.net** el servidor DNS le redireccionará al servidor MONITOR que es el encargado del panel de monitorización del clúster.

APLICACIÓN DE MONITORIZACION

El sistema de monitorización **clusterCP** permite:

- Controlar el estado de los clústers.
- Generar estadísticas de los fallos de los nodos (caídas).
- Permite modificar el algoritmo de conexiones del balanceador de carga.
- Acceder a las estadísticas de los balanceadores de carga.
- Añadir clústers.

APLICACIÓN DE MONITORIZACION - MVC (I)

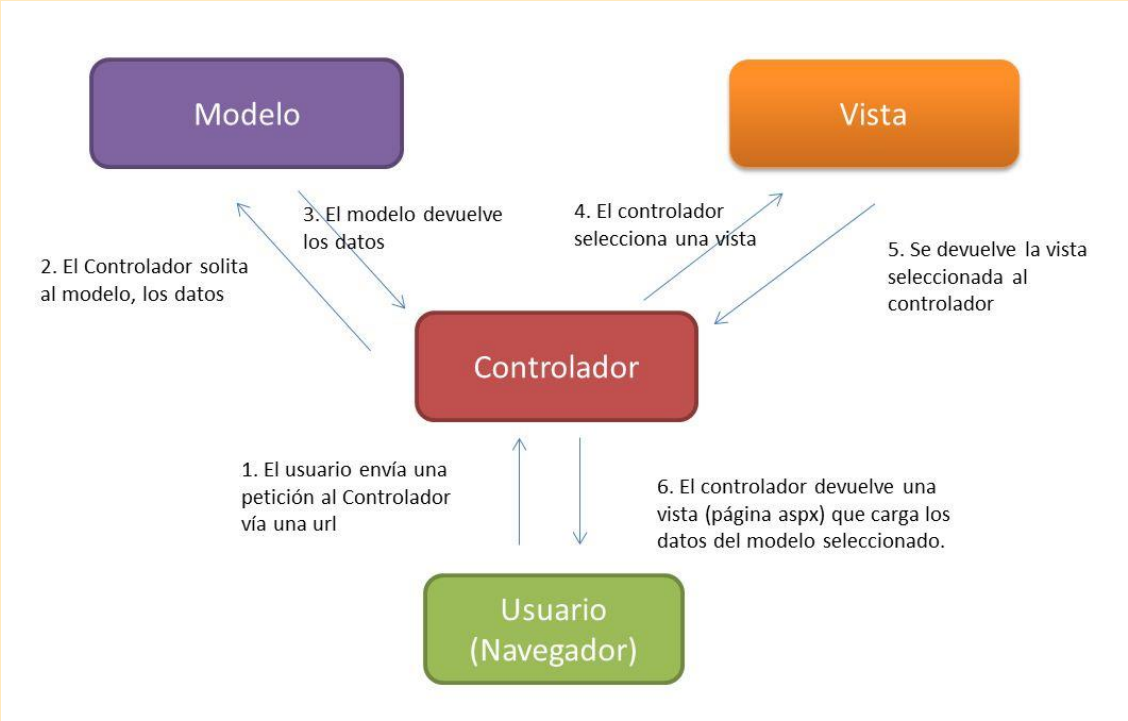
La aplicación se ha realizado siguiendo el patrón de diseño MVC (Modelo-Vista-Controlador).

Permite separar los datos que usuario visualiza (estilo) de la aplicación que gestiona los datos y accede a la base de datos:

Consta de 3 componentes:

- Modelo
- Vista
- Controlador

APLICACIÓN DE MONITORIZACION – MVC (II)



APLICACIÓN DE MONITORIZACION – MongoDB (I)



MongoDB es un sistema de Base de Datos NoSQL (No solo SQL – Not Only SQL). A diferencia de MySQL, PostgreSQL, etc. los datos no se guardan en tablas ni filas ni columna, si no en **colecciones, documentos y nodo** respectivamente lo que permite no tener un esquema fijo, rapidez y escalabilidad.

Tendremos 3 colecciones con los siguientes documentos:

- Clusters: nombre, ip, nodos
- Caidas: nodo, fecha, hora
- Balanceadores: nombre, ip

APLICACIÓN DE MONITORIZACION – MongoDB (II)

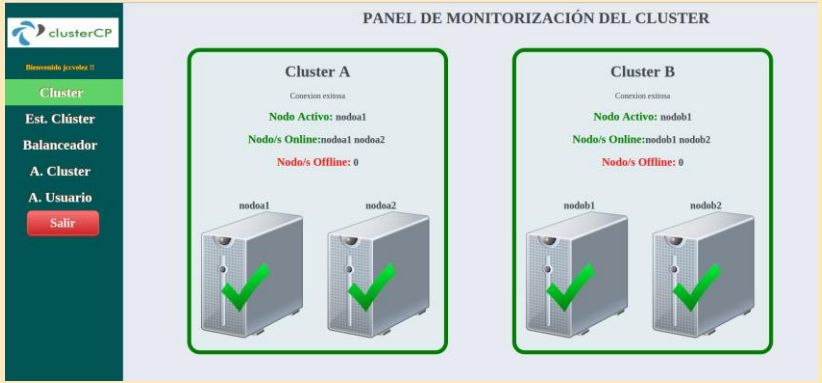
- MongoDB usa JavaScript para realizar las consultas a través de la consola.

- **Create**
  db.coleccion.insert(documento)
- **Read**
  db.coleccion.find/findOne(expresion)
- **Update**
  db.coleccion.update(documento_original, documento_modificado)
- **Delete**
  db.coleccion.remove(expresion)

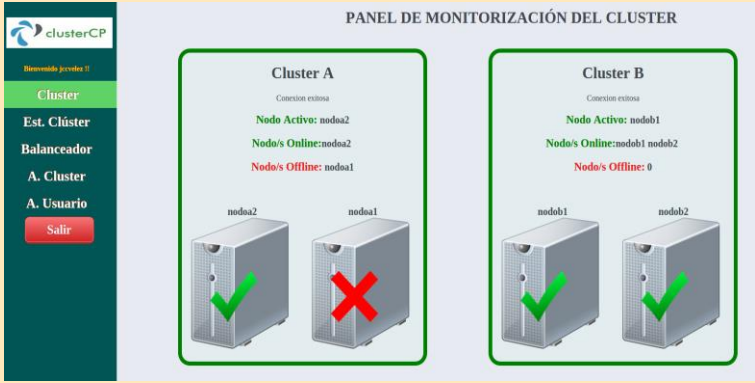
- PHP permite conectar a mongoDB a traves de driver o extension, PHP se conecta a MongoDB usando POO (Programacion Orientada a Objetos)

APLICACIÓN DE MONITORIZACION - CLUSTER (I)

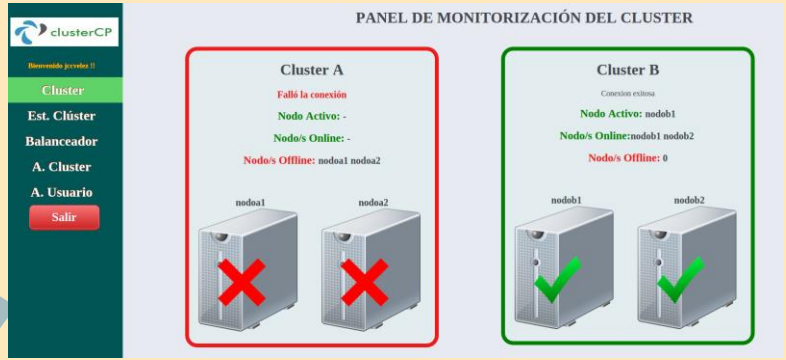
CLUSTER CORRECTO



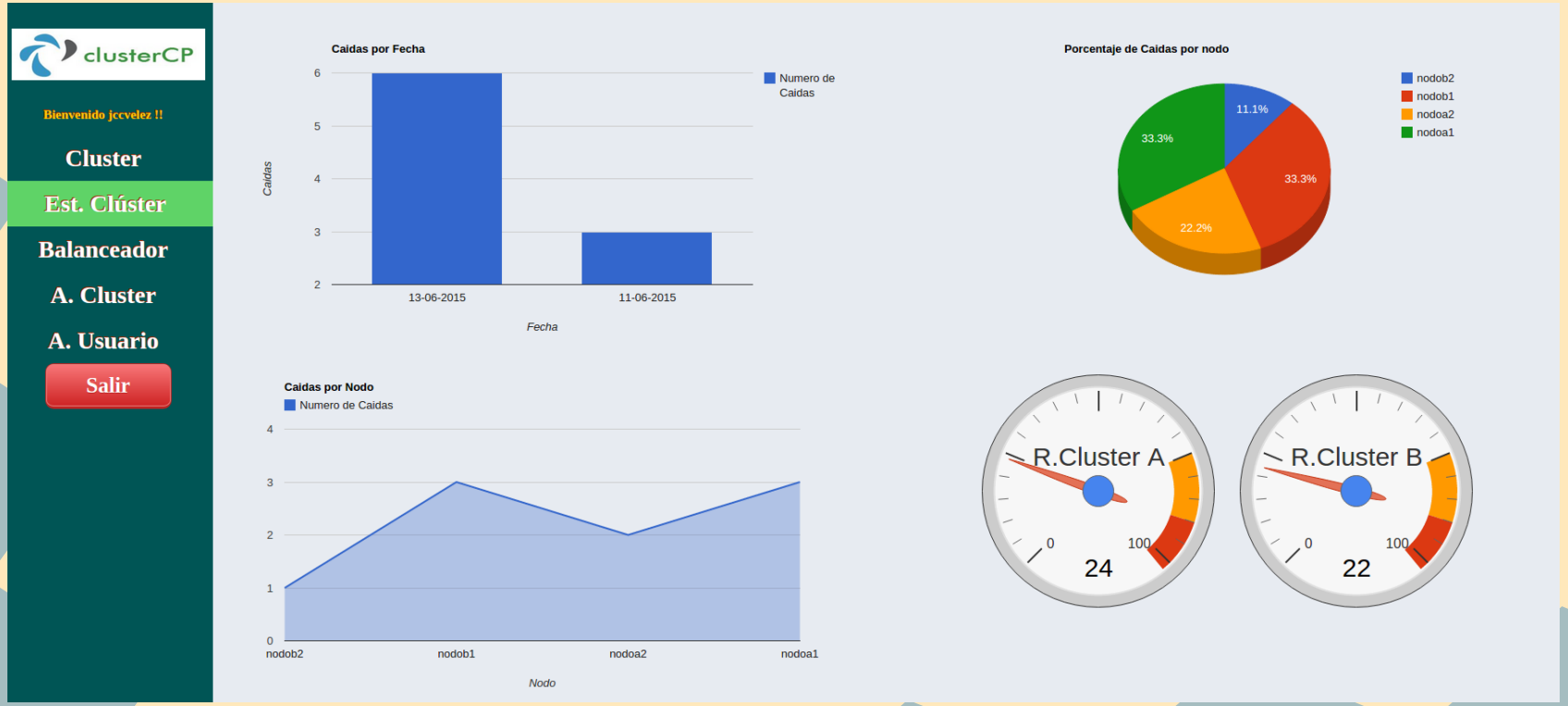
NODO FALLIDO



CLUSTER FALLIDO



APLICACIÓN DE MONITORIZACION – CLUSTER (II)



APLICACIÓN DE MONITORIZACION - BALANCEADOR (I)



APLICACIÓN DE MONITORIZACION - BALANCEADOR (II)

CLUSTER CORRECTO

HAProxy version 1.5.12, released 2015/05/02

Statistics Report for pid 5467

> General process information

pid = 5467 (process #1, nbproc = 1)
uptime = 0d 0h42m15s
system limits: memmax = unlimited; ulimit-n = 4034
maxsock = 4034; **maxconn** = 2000; **maxpipes** = 0
current conns = 1; current pipes = 0/0; conn rate = 0/sec
Running tasks: 1/9; idle = 100 %

active UP
active UP, going down
active DOWN, going up
active or backup DOWN
active or backup DOWN for maintenance (MAINT)
active or backup SOFT STOPPED for maintenance
backup UP
backup UP, going down
backup DOWN, going up
not checked
Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

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

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

clusterfrontend

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				0	8	-	1	6	2 000	259			151 237	368 076	0	0	0	0				OPEN								

backcluster

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
clustera	0	0	-	0	1		0	1	-	2	2	42m7s	732	822		0		0	0	0	0	11m22s UP	L4OK in 0ms	1	Y	-	6	1	4m27s	-
clusterb	0	0	-	0	1		0	1	-	2	2	42m6s	732	822		0		0	0	0	0	42m15s UP	L4OK in 0ms	1	Y	-	0	0	0s	-
Backend	0	0		0	1		0	1	200	4	4	42m6s	1 464	1 644	0	0		0	0	0	0	42m15s UP		2	2	0		0	0s	

backclustercp

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
monitor	0	0	-	0	8		0	5	-	253	253	1m24s	145 608	236 799		0		0	0	0	0	42m15s UP	L4OK in 1ms	1	Y	-	0	0	0s	-
Backend	0	0		0	8		0	5	200	253	253	1m24s	145 608	236 799	0	0		0	0	0	0	42m15s UP		1	1	0		0	0s	

CLUSTER FALLIDO

HAProxy version 1.5.12, released 2015/05/02

Statistics Report for pid 5467

> General process information

pid = 5467 (process #1, nbproc = 1)
uptime = 0d 0h43m22s
system limits: memmax = unlimited; ulimit-n = 4034
maxsock = 4034; **maxconn** = 2000; **maxpipes** = 0
current conns = 1; current pipes = 0/0; conn rate = 0/sec
Running tasks: 1/9; idle = 100 %

active UP
active UP, going down
active DOWN, going up
active or backup DOWN
active or backup DOWN for maintenance (MAINT)
active or backup SOFT STOPPED for maintenance
backup UP
backup UP, going down
backup DOWN, going up
not checked
Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

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

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

clusterfrontend

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				0	8	-	1	6	2 000	259			168 323	941 733	0	0	0					OPEN								

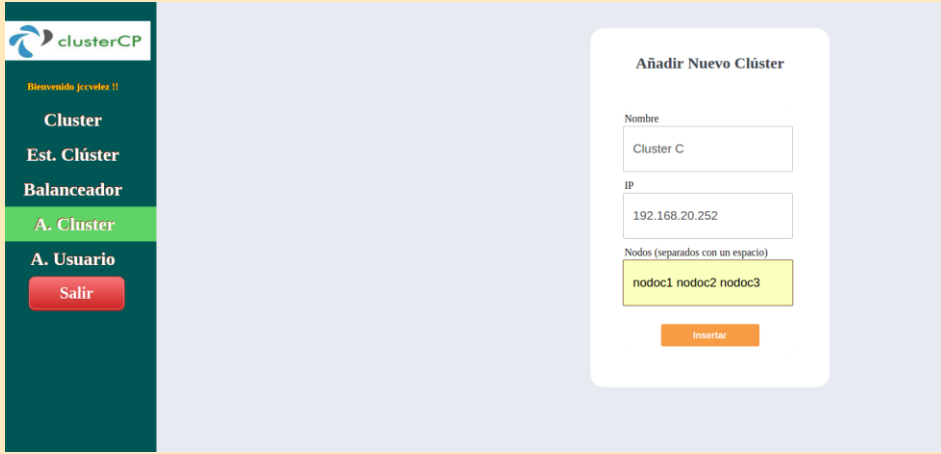
backcluster

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
clustera	0	0	-	0	1		0	1	-	2	2	43m14s	732	822		0		0	0	0	0	1s DOWN	L4TOUT in 2001ms	1	Y	-	10	2	4m28s	-
clusterb	0	0	-	0	1		0	1	-	2	2	43m13s	732	822		0		0	0	0	0	43m22s UP	L4OK in 1ms	1	Y	-	0	0	0s	-
Backend	0	0		0	1		0	1	200	4	4	43m13s	1 464	1 644	0	0		0	0	0	0	43m22s UP		1	1	0		0	0s	

backclustercp

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
monitor	0	0	-	0	8		0	5	-	253	253	2m31s	145 608	236 799		0		0	0	0	0	43m22s UP	L4OK in 0ms	1	Y	-	0	0	0s	-
Backend	0	0		0	8		0	5	200	253	253	2m31s	145 608	236 799	0	0		0	0	0	0	43m22s UP		1	1	0		0	0s	

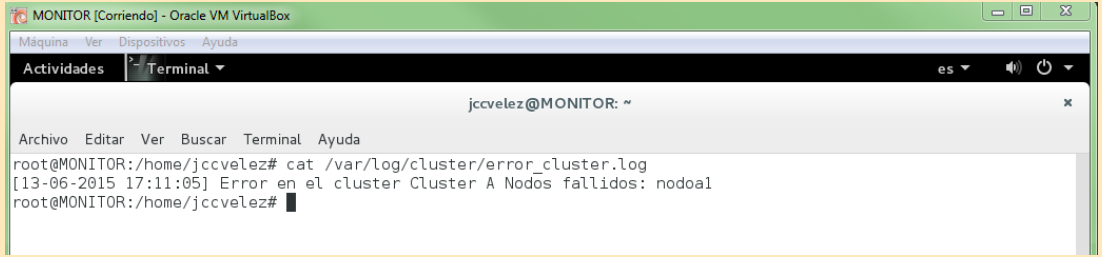
APLICACIÓN DE MONITORIZACION – AÑADIR CLUSTER

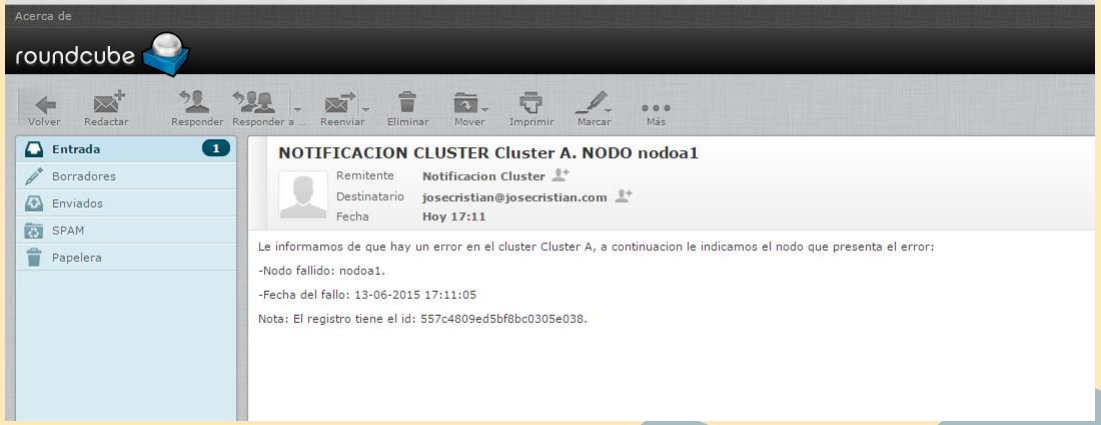




NOTIFICACION POR CORREO

- Cuando algún nodo falla automáticamente se guarda en la colección caídas el error, se guarda un log en el servidor y se envía un correo.





PREGUNTAS



FIN