



**Curso: Sistemas Operativos II**  
**Plataforma: Linux - OpenSuse**

# CLUSTER

---

Quando se requiere de un sistema en donde la infraestructura es construida haciendo uso de un conjunto de computadoras mediante la utilización de componentes de hardware comunes con la principal característica de que se comportan como si fuesen una sola computadora, se hace evidente la utilización de conceptos asociados como lo es un Cluster, por ende este es un grupo de múltiples computadoras unidos mediante una red de alta velocidad, de tal forma que el conjunto es visto como una única computadora.



# INDICE

---

TEMA	No. PAGINA
Introducción .....	3
Objetivos .....	4
Marco Teórico .....	5
Configuración .....	7
Conclusiones .....	21
Bibliografía .....	22
Recomendaciones .....	23



# INTRODUCCION

---

En la actualidad con el surgimiento de nuevas tecnologías y nuevas necesidades de los usuarios de sistemas informáticos, surge la necesidad de poder implementar un concepto el cual garantice y asegure que los recursos y las aplicaciones de importancia decisiva permanecerán disponibles así como que también la carga de trabajo sea balanceada para garantizar el alto rendimiento.

Por ende un clúster es un grupo de sistemas independientes, conocidos como nodos, que trabajan juntos como un sistema único para garantizar la premisa anterior mencionada en donde hacíamos referencia a garantizarle al usuario la disponibilidad.

La organización en clústeres permite a los usuarios y administradores tener acceso a los nodos y administrarlos como un sistema único en lugar de como equipos independientes.

Un clúster de servidores puede comprender hasta ocho nodos y se puede configurar de tres formas: como clúster de servidores de nodo único, como clúster de servidores de dispositivo de quórum único o como clúster de servidores de conjunto de nodos mayoritario.

En este documento podremos adentrarnos en conceptos fundamentales asociados a los clusters así como un ejemplo de configuración en modo activo/activo el cual hace referencia a que si un servidor o nodo posee mucha carga esta se distribuye entre los nodos para obtener un alto rendimiento en la ejecución de los procesos.



# OBJETIVOS

---

## General:

- Lograr una correcta implementación de un sistema de CLUSTER, derivado de una comprensión a cabalidad de los conceptos que de el se derivan.

## Específicos:

- Estudiar la historia de los sistemas de CLUSTER en general.
- Comprender el funcionamiento de un CLUSTER montado en un sistema Linux - OpenSuse.
- Estudiar los requerimientos a nivel de hardware para su correcta implementación.
- Ejemplificar la configuración mediante un caso práctico en modo activo/activa.



# CLUSTER:

---

## Historia de CLUSTER:

El comienzo del término y del uso de este tipo de tecnología es desconocido pero se puede considerar que comenzó a finales de los años 50 y principios de los años 60.

La base formal de la ingeniería informática de la categoría como un medio de hacer trabajos paralelos de cualquier tipo fue posiblemente inventado por Gene Amdahl de IBM, que en 1967 publicó lo que ha llegado a ser considerado como el papel inicial de procesamiento paralelo: la Ley de Amdahl que describe matemáticamente lo que se puede esperar paralelizando cualquier otra serie de tareas realizadas en una arquitectura paralela.

La historia de los primeros grupos de computadoras es más o menos directamente ligado a la historia de principios de las redes, como una de las principales motivaciones para el desarrollo de una red para enlazar los recursos de computación, de hecho la creación de un cluster de computadoras. Las redes de conmutación de paquetes fueron conceptualmente inventados por la corporación RAND en 1962.

Utilizando el concepto de una red de conmutación de paquetes, el proyecto ARPANET logró crear en 1969 lo que fue posiblemente la primera red de computadoras básico basadas en el cluster de computadoras por cuatro tipos de centros informáticos (cada una de las cuales fue algo similar a un "cluster" pero no un "comodity cluster" como hoy en día lo entendemos.

El primer producto comercial de tipo cluster fue ARCnet, desarrollada en 1977 por Datapoint pero no obtuvo un éxito comercial y los clusteres no consiguieron tener éxito hasta que en 1984 VAXcluster produjeran el sistema operativo VAX/VMS.

[ 1 ]

## Definición de CLUSTER:

Es un grupo de computadoras que están interconectadas y funcionan como una sola unidad de proceso de información.

[ 2 ]



## Funcionamiento:

Un cluster funciona como un sistema único o monolítico, es decir, para un usuario o para un problema dado todos los nodos (o sistemas miembros) del cluster son vistos como una sola computadora. Esta es la particularidad más representativa de un cluster.

[ 2 ]

## Clasificación:

Esta clasificación se basa en relación al uso que se les da a los clusters y los servicios que ofrecen, por lo que existen las siguientes categorías:

**High Performance:** Son clusters en los cuales se ejecutan tareas que requieren de gran capacidad computacional, grandes cantidades de memoria, o ambos a la vez. El llevar a cabo estas tareas puede comprometer los recursos del cluster por largos periodos de tiempo.

**High Availability:** Son clusters cuyo objetivo de diseño es el de proveer disponibilidad y confiabilidad. Estos clusters tratan de brindar la máxima disponibilidad de los servicios que ofrecen. La confiabilidad se provee mediante software que detecta fallos y permite recuperarse frente a los mismos, mientras que en hardware se evita tener un único punto de fallos.

**High Throughput:** Son clusters cuyo objetivo de diseño es el ejecutar la mayor cantidad de tareas en el menor tiempo posible. Existe independencia de datos entre las tareas individuales. El retardo entre los nodos del cluster no es considerado un gran problema.



[ 1 ]



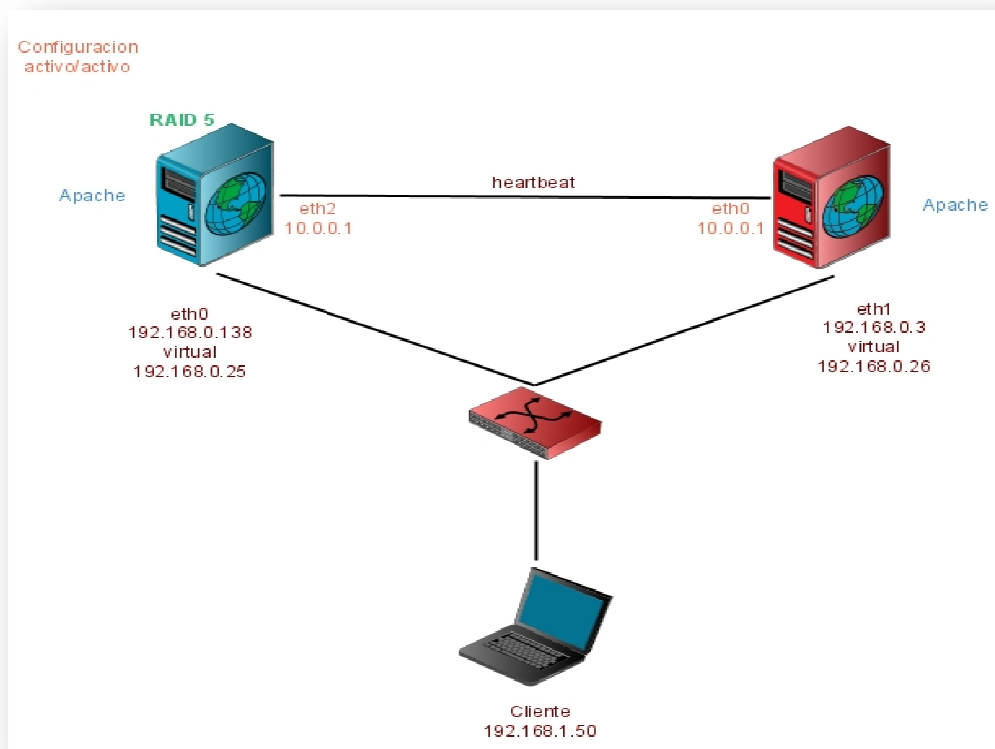
# CONFIGURACION

## ASPECTOS TECNICOS.

La configuración que se ejemplificará a continuación se realizó bajo las siguientes condiciones:

	CLUSTER	
	MAQUINA 1	MAQUINA 2
		
<b>Sistema Operativo</b>	Linux	Linux
<b>Distribución</b>	OpenSuse 10.3 Kernel 2.6	OpenSuse 10.3 Kernel 2.6
<b>Disco Duro</b>	10 GB y un RAID 5 de 74 GB	40 GB
<b>Procesador</b>	Pentium 4 de 3.0 Ghz	AMD 1 GB
<b>Memoria</b>	2 Gb de RAM	512 MB de RAM

## TOPOLOGIA A IMPLEMENTAR:

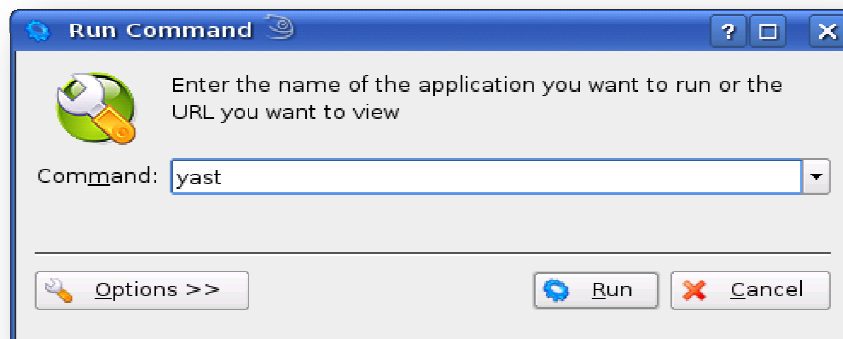




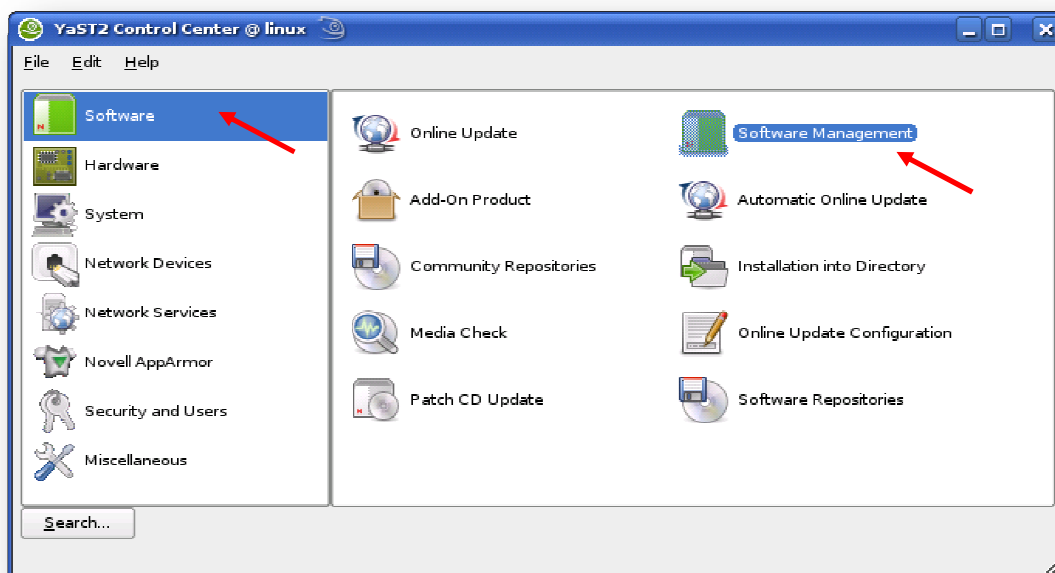
## PASOS PARA CONFIGURAR CLUSTER:

### INSTALACION DE HEARTBEAT Y APACHE:

1. Presionamos **Alt+F2**, con ello se nos abrirá una ventana en la cual escribimos **yast** en modo **superusuario** para poder ejecutar el centro de administración de OpenSuse en el cual podremos configurar el sistema de CLUSTER.



2. Se nos mostrará el centro de control **YaST2** de OpenSuse en el cual debemos dirigirnos a la parte de **Software** dentro de las categorías de configuraciones, luego debemos seleccionar la opción de **Software Management**, la cual nos brinda la funcionalidad de administrar los paquetes, en este caso nos brinda la posibilidad de poder realizar.

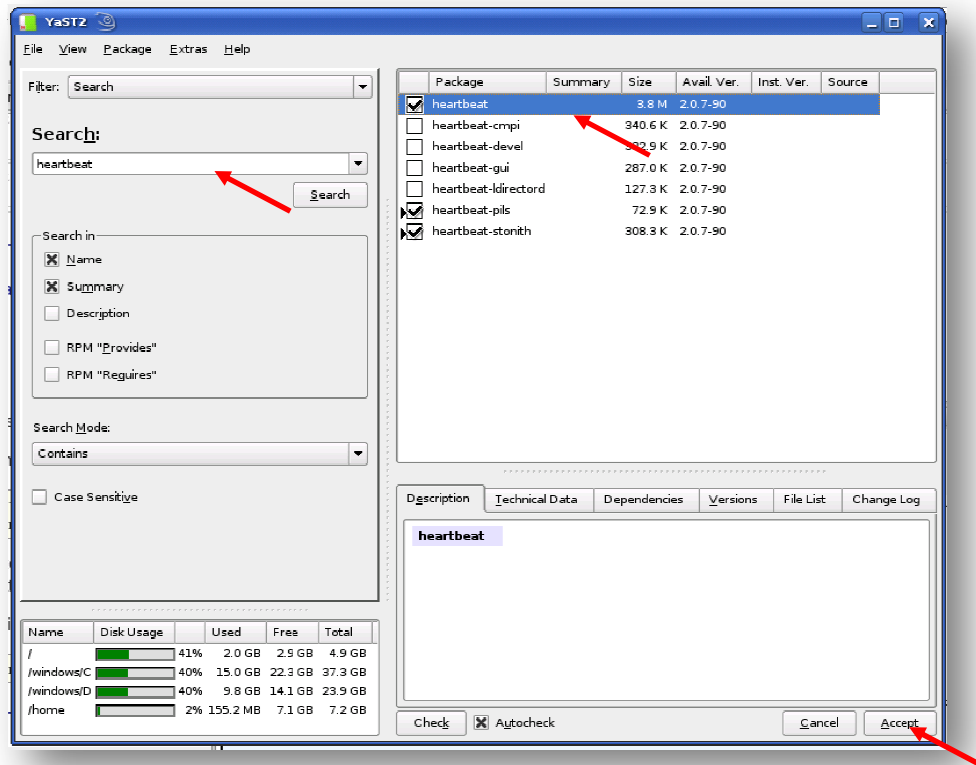




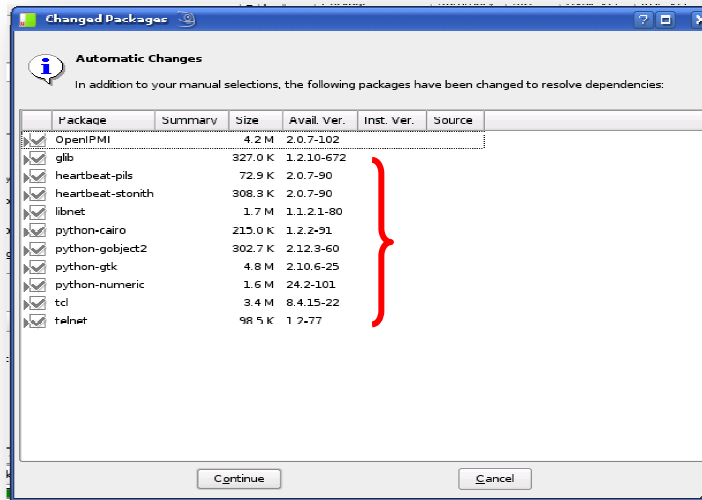


3. Luego se nos mostrará la siguiente pantalla en donde podremos buscar y seleccionar los paquetes a instalar, en este caso instalaremos **Heartbeat**.

En el campo **SEARCH** debemos escribir **Heartbeat** para que automáticamente se realice la búsqueda de paquetes, luego solo los seleccionamos del listado resultante de la búsqueda, luego de seleccionar los paquetes a instalar solo le damos clic en **accept** para dirigirnos al siguiente paso.

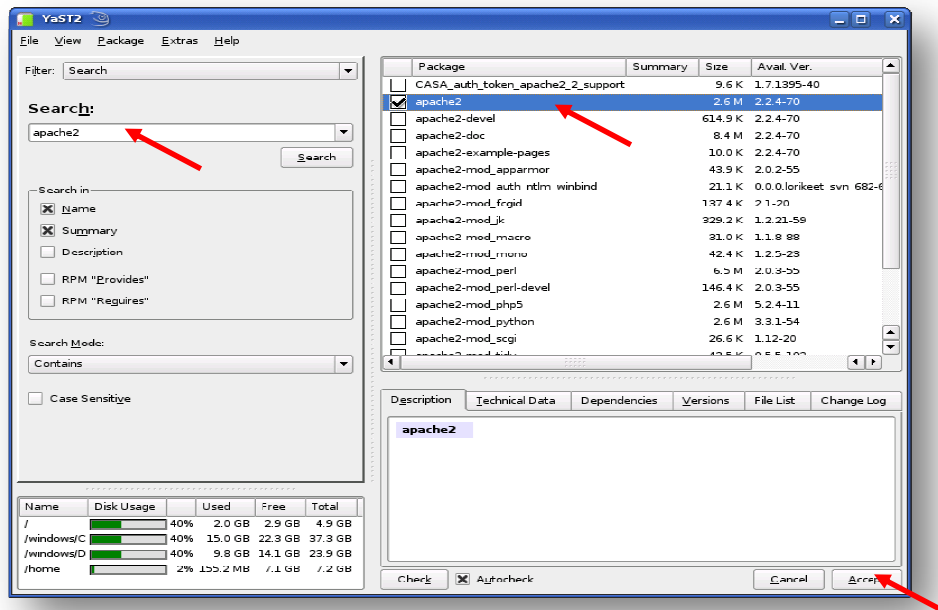


4. Luego se nos mostrará una ventana, la cual nos indica las dependencias que automáticamente se instalarán, esta es una ventaja de los gestores de paquetes gráficos dado que no tenemos que preocuparnos por instalar manualmente dichas dependencias, en la distribución OpenSuse, estos se instalan automáticamente, únicamente debemos darle clic en el botón **continue** para proseguir con la instalación.



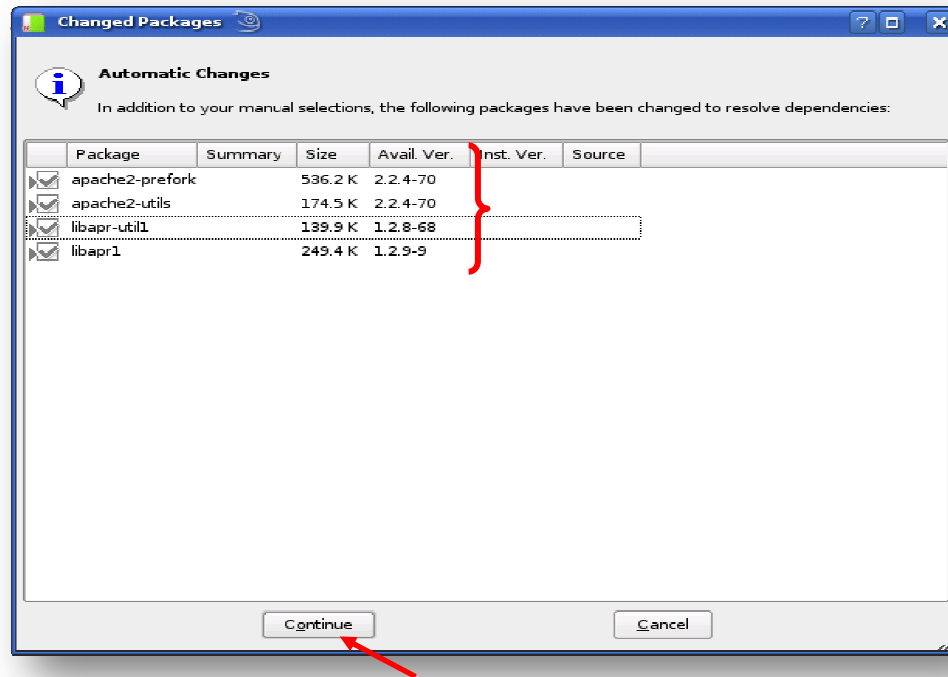
5. Dada por finalizada la instalación de **Heartbeat**, procedemos a instalar **Apache**, de manera homologa al proceso que anteriormente hemos realizado con la instalación de **Heartbeat**, pero ahora debemos instalar **Apache** en cada uno de los nodos que conforman nuestra topología, en este caso **Nodo1** y **Nodo2**.

En el campo **SEARCH** debemos escribir **Apache** para que automáticamente se realice la búsqueda de paquetes, luego solo los seleccionamos del listado resultante de la búsqueda, luego de seleccionar los paquetes a instalar solo le damos clic en **accept** para dirigirnos al siguiente paso.





6. Luego se nos mostrará una ventana, la cual nos indica las dependencias que automáticamente se instalarán, únicamente debemos darle clic en el botón **continue** para proseguir con la instalación.



Y con ello se da por concluido el proceso de instalación de los paquetes necesarios para realizar el cluster en la distribución OpenSuse.

### Configuración de las tarjetas de Red:

La siguiente tabla nos muestra como hemos configurados las interfaces de red dentro de nuestra topología.

Nodo 1	Nodo 2	Descripción
# ifconfig eth0 192.168.0.138	# ifconfig eth0 192.168.0.3	red publica
# ifconfig eth1 10.0.0.1	# ifconfig eth1 10.0.0.2	red privada para Heartbeat



Luego procedemos a modificar el archivo “*/etc/hosts*” el cual debe de tener el siguiente contenido:

```
127.0.0.1          nodo1 localhost.localdomain localhost
192.168.0.3       nodo2
192.168.0.25      servicio1
192.168.0.26      servicio2
```

## Configuración de Heartbeat:

En la siguiente tabla se muestran los archivos de configuración y de log que utiliza *heartbeat*, se debe de realizar la correcta configuración de estos archivos antes de iniciar *heartbeat*.

Directorio	Descripción
<i>/etc/ha.d/authkeys</i>	Autenticación de los nodos
<i>/etc/ha.d/ha.cf</i>	Configuración general de Heartbeat
<i>/etc/ha.d/haresources</i>	Configuración de los recursos
<i>/var/lib/heartbeat/crm/cib.xml</i>	Se generan con <i>haresources2cib.py</i>
<i>/var/log/ha-log</i>   <i>/var/log/message</i>	Bitácora

## Configurando el archivo ha.cf

Este archivo le dice a *Heartbeat* qué tipos de interfaces se van a utilizar para comunicarse con los otros nodos del cluster. También define los nodos que van a formar el cluster y los archivos de log donde se registran las acciones de la aplicación. A continuación se muestra el contenido que debería de tener este archivo:

```
# Facility to use for syslog()/logger
logfacility    local0
```

```
# Intervalo de tiempo en el cual se generan los latidos
keepalive 2
```



**# Tiempo que transcurre antes de considerar a un nodo muerto**  
deadtime 30

**#**  
warntime 3

**#**  
initdead 100

**# Puerto que se utilizara para la comunicacion bcast/ucast**  
udpport 694

**# Velocidad en baudios para el puerto serial**  
baud 19200

**# Este permite que solo se produzca balanceo al momento del fallo**  
auto\_failback on

**# Permite la generacion de archivos de log**  
use\_logd yes

**# Interfaz donde escuha heartbeat**  
bcast eth2

**# Definicion de los nodos del cluster, estos nombres deben de ser los mismos**  
**#que los nombre de host de las maquinas, de lo contrario heartbeat no inicia**  
node nodo1  
node nodo2

**# Se habilita el Cluster Resource Manager CRM**  
crm yes

**# Actualiza el CRM al ocurrir un fallo**  
respawn hacluster /usr/lib/heartbeat/ipfail



## Configurando el archivo haresources

En este archivo definen los recursos que son gestionados por **Heartbeat**. Los recursos son script Linux Standar Base (LSB) como los que se usan para arrancar o parar servicios al arrancar el sistema en los diferentes runlevels.

**Heartbeat** buscará estos scripts en estas dos rutas: `/etc/rc.d` y `/etc/ha.d/resource.d`, por lo que se debe alojar el script al menos en una de las dos.

A continuación se muestra el contenido de este archivo

```
nodo1 192.168.0.25 apache2
```

```
nodo2 192.168.0.26 apache2
```

En donde las direcciones IP que se muestran son las direcciones virtuales en las que se levantarán los servicios. El siguiente campo que le sigue es el nombre del servicio que se desea levantar en el cluster, en este caso es un **servidor Apache** el cual tiene como nombre de servicio **"apache2"**.

**Nota:** Este archivo debe ser el mismo en ambos nodos.

## Configurando el archivo cib.xml

Para generar el archivo **cib.xml** se debe cambiar el usuario y grupos propietarios de los directorios de **HeartBeat**, para ello se debe escribir:

```
#chown -R hacluster:haclient /var/run/heartbeat
```

```
#chown -R hacluster:haclient /var/lib/heartbeat
```

```
#chown -R hacluster:haclient /usr/lib/heartbeat
```

Si al ejecutar los comandos anteriores nos da error es posiblemente porque el usuario y grupo no existen, de ser este el caso se deben de crear, para esto se utilizan los siguientes comandos:



```
#useradd hacluster
```

```
#groupadd haclient
```

Para crear el archivo de configuración del CRM en `/var/lib/heartbeat/crm/cib.xml` se utiliza el siguiente comando:

```
#!/usr/lib/heartbeat/haresources2cib.py /etc/ha.d/haresources
```

Puede ser que al ejecutar el comando anterior nos de un error diciendo que el archivo ya existe, esto es porque **heartbeat** al momento de su instalación crea este archivo, de ocurrir esto se debe de eliminar este archivo antes de ejecutar el comando anterior. Para eliminar este archivo se utiliza los siguientes comando:

```
# rm /var/lib/heartbeat/crm/cib.xml
```

```
# rm /var/lib/heartbeat/crm/cib.xml.sig
```

Una vez genera el archivo `cib.xml` se debe de renombrar el archivo `haresources` por `haresources_old`

Como ultimo paso en la configuración de **heartbeat** se debe de agregar la siguiente restricción en la sección de `<constraints>` del archivo `cib.xml`.

```
<rsc_location id="group_1:connected" rsc="group_1">  
  <rule id="group_1:connected:rule" score_attribute="pingd">  
    <expression id="group_1:connected:expr:defined" attribute="pingd" operation="defined"/>  
  </rule>  
</rsc_location>
```



## Sincronización Horaria entre los nodos:

Para el correcto funcionamiento de **Heartbeat** se recomienda que todos los nodos del cluster tengan la fecha y hora sincronizadas.

## Arrancar Heartbeat

Para arrancar el paquete es necesario ejecutar el siguiente script logeado como **root**:

```
/etc/init.d/heartbeat start
```

```
daniel@:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
nod01:~ # /etc/init.d/heartbeat start
Starting High-Availability services2008/05/16_23:26:56 INFO: Resource is stopped
2008/05/16_23:26:57 INFO: Resource is stopped
nod01:~ # done
```

Para comprobar que **Heartbeat** ha arrancado correctamente se debe ejecutar el siguiente comando:

```
#!/etc/init.d/heartbeat status
```

```
daniel@:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
nod01:~ # /etc/init.d/heartbeat status
Checking for High-Availability services
nod01:~ # running
```





Para parar Heartbeat habrá que ejecutar este script logueado como root

```
#!/etc/init.d/heartbeat stop
```

```
daniel@:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
nodol:~ # /etc/init.d/heartbeat stop
Stopping High-Availability services
logd[9382]: 2008/05/16_23:25:09 debug: Stopping ha_logd with pid 5067
logd[9382]: 2008/05/16_23:25:09 info: Waiting for pid=5067 to exit
logd[9382]: 2008/05/16_23:25:10 info: Pid 5067 exited
nodol:~ #
```

Para comprobar que **Heartbeat** ha parado correctamente se debe ejecutar el siguiente comando:

```
#ps -ef | grep heartbeat | grep -v grep
```

Si este comando devuelve que aún se está ejecutando algún proceso volver a ejecutar el script de parada y si aún así persistiera algún proceso, habrá que proceder a matarlo con un **kill -9**

### Comprobar qué servicios hay en un cluster:

Para realizar dicha acción hay que loguearse como **root** en cualquiera de las máquinas que formen el cluster y ejecutar el siguiente comando:

```
# crm_resource -L | grep Group
```

```
daniel@:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
nodol:~ # crm_resource -L | grep Group
Resource Group: group_1
Resource Group: group_11
nodol:~ #
```



### Comprobar en qué nodo se encuentra el servicio:

Para realizar dicha acción se debe logarse como **root** en cualquiera de las máquinas que formen el cluster y ejecutar el siguiente comando:

```
# crm_resource -W -r group_1 -t group
```

```
daniel@:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
nodo1:~ # crm_resource -W -r group_1 -t group
resource group_1 is running on: nodo1
nodo1:~ #
```

### Arrancar un servicio

Para arrancar un servicio se debe loguearse como **root** en cualquiera de las máquinas que formen el cluster y ejecutar el siguiente comando:

```
# crm_resource -r group_1 -t group -p target_role -v started
```

```
daniel@:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
nodo1:~ # crm_resource -r group_1 -t group -p target_role -v started
nodo1:~ #
```

### Parar un servicio

Para detener un servicio se debe loguearse como **root** en cualquiera de las máquinas que formen el cluster y ejecutar el siguiente comando:

```
# crm_resource -r group_1 -t group -p target_role -v stopped
```



```
daniel@:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
nodo1:~ # crm_resource -r group_1 -t group -p target_role -v stopped
nodo1:~ #
```

### Comprobar que el servicio ya no está corriendo en ningún nodo.

Migrar el servicio a otro nodo

Se debe loguear como **root** en cualquiera de las máquinas que formen el cluster y ejecutar el siguiente comando:

```
# crm_resource -M -r group_1 -t group -H nodo1
```

### Comprobar que el servicio se ha realocado al nodo indicado

Migrar el servicio al nodo primario

Se debe loguear como **root** en cualquiera de las máquinas que formen el cluster y ejecutar el siguiente comando:

```
# crm_resource -U -r group_1 -t group
```

### Dejar un Nodo en Standby

Se debe loguearse como **root** en cualquiera de las máquinas que formen el cluster y ejecutar el siguiente comando:

```
# crm_resource -H nodo1 -v off
```

### Poner un Nodo OnLine

Se debe loguear como **root** en cualquiera de las máquinas que formen el cluster y ejecutar el siguiente comando:

```
# crm_resource -H nodo1 -v on
```



## Monitorizar el estado del Cluster

Se debe loguear como **root** en cualquiera de las máquinas que formen el cluster y ejecutar:

```
# crm_mon -i2
```

```
daniel@:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
Refresh in 4s...
=====
Last updated: Fri May 16 23:37:39 2008
Current DC: nodo2 (5f068bdc-2ccc-4e74-a454-2e4529bfb738)
2 Nodes configured.
2 Resources configured.
=====
Node: nodo2 (5f068bdc-2ccc-4e74-a454-2e4529bfb738): online
Node: nodo1 (c7d1746a-3339-4639-9a24-917cd7e12a69): online
Resource Group: group_1
IPaddr_192_168_0_26 (heartbeat::ocf:IPaddr): Started nodo2
apache2_12 (lsb:apache2): Started nodo2
```

Y con ello damos por concluida la configuración de nuestro cluster en un sistema operativo Linux, bajo la distribución OpenSuse.



# CONCLUSIONES

---

En base a lo anteriormente expuesto se concluye lo siguiente:

- Que un cluster puede ser utilizado para muchas aplicaciones.
- Que un cluster es una solución económica para resolver problemas que necesitan muchos recursos de computo.
- Que un cluster es una forma económica tanto en software como en hardware para realizar tareas de forma distribuido y paralelo.
- Que un cluster se define como un grupo de computadoras que están interconectadas y funcionan como una sola unidad de proceso de información..



# TUTORIAL DESARROLLADO POR

---

	<b>Carlos Enrique Rodas Gálvez</b>	<b>2002-12383</b>
	<b>Álvaro Daniel Castillo Carrera</b>	<b>2003-12531</b>
	<b>Miguel Enrique Guerra Connor</b>	<b>2002-17739</b>
	<b>Vinicio Rodolfo Miranda Orozco</b>	<b>2002-12355</b>

## BIBLIOGRAFIA

---

La sección conceptual de este documento fue investigada bajo las siguientes fuentes bibliográficas.

### Motor de Búsqueda:

[www.google.com.gt](http://www.google.com.gt)

### Sitios Virtuales Consultados:

[1] [http://es.wikipedia.org/wiki/Cluster\\_de\\_computadores](http://es.wikipedia.org/wiki/Cluster_de_computadores)

[2] [http://www.dei.uc.edu.py/tai2003-2/clustering/html/concepto\\_de\\_cluster.html](http://www.dei.uc.edu.py/tai2003-2/clustering/html/concepto_de_cluster.html)



## RECOMENDACIONES

---

En base a la experiencia que se deriva del equipo de implementación se plantean unas recomendaciones básicas que deben ser tomadas en cuenta para tener una óptima configuración e implementación de un sistema Cluster.

- 1) Realizar las acciones tales como instalación y configuración de paquetes logueado como usuario **root**.
- 2) Hacer copias de seguridad de los archivos de configuración antes de proceder a modificarlos.
- 3) Verificar los archivos logs en caso ocurriera un error para ir escribir en una consola `#tail /var/log/message/`.
- 4) Monitorear constantemente el estado del cluster, esto lo podemos realizar con las instrucciones proporcionadas en este documento.