

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA

UNAN – León

Facultad de Ciencias y Tecnología

Departamento de Computación



Elaboración de prácticas de laboratorio para el componente Computación en la nube del plan académico 2011 de la carrera de Ingeniería en Telemática de la UNAN-León.

Tesis para optar al título de

INGENIERO EN TELEMÁTICA

Presentado por:

Br. Angel Evelio Maradiaga Leytón
Br. Ervin Ismael Montes Téllez

Tutor: MSc. Denis Espinoza Hernández.

León, 5 de Agosto de 2016

Agradecimiento

Primeramente al Eterno Dios por darme la sabiduría y fuerza para concluir las metas propuestas.

A mis padres Ervin Montes y Jacinta Téllez, por la confianza que tuvieron en cada momento de mi vida, por su amor y todo el esfuerzo y sacrificio que pasaron para que yo pudiera culminar mis estudios.

A Raquel Esquivel y mis hermanas que ha sido un gran apoyo para mi vida.

Al master Dennis Espinoza que con su conocimiento, sus orientaciones, su manera de trabajar, su persistencia, su paciencia, su amistad y motivación ha sido fundamental para mi formación.

A mis amigos y compañeros que forman parte del grupo Telematixwar por compartir conmigo cada momento especial, y por la amistad incondicional que siempre me brindaron.

Finalmente agradezco a quien lee este apartado y más de mi tesis, por permitir a mis experiencias, investigaciones y conocimiento incurrir dentro de su repertorio de información mental.

Ervin Montes.

Primeramente al Eterno Dios por darme la sabiduría y fuerza para concluir las metas propuestas.

A mis padres Evelio Maradiaga y Isabel Leytón, por la confianza que tuvieron en cada momento de mi vida, por su amor y todo el esfuerzo y sacrificio que pasaron para que yo pudiera culminar mis estudios.

Al master Dennis Espinoza que con su conocimiento, sus orientaciones, su manera de trabajar, su persistencia, su paciencia, su amistad y motivación ha sido fundamental para mi formación.

A mis amigos y compañeros que forman parte del grupo Telematixwar por compartir conmigo cada momento especial, y por la amistad incondicional que siempre me brindaron.

Finalmente agradezco a quien lee este apartado y más de mi tesis, por permitir a mis experiencias, investigaciones y conocimiento incurrir dentro de su repertorio de información mental.

Angel Maradiaga.

Índice general

1	Introducción	1
1.1	ANTECEDENTES	2
1.2	PLANTEAMIENTO DEL PROBLEMA	4
1.3	JUSTIFICACIÓN	6
1.3.1	Originalidad	6
1.3.2	Alcance	6
1.3.3	Producto	6
1.3.4	Impacto	7
1.4	OBJETIVOS	8
1.4.1	Objetivo general	8
1.4.2	Objetivos específicos	8
2	Diseño Metodológico	9
2.1	RECOLECCIÓN DE INFORMACIÓN	9
2.2	SELECCIÓN DE LA HERRAMIENTA A IMPLEMENTAR	9
2.3	ELABORACIÓN Y DESARROLLO DE LOS LABORATORIOS	10
3	Desarrollo Teórico	12
3.1	COMPUTACIÓN EN LA NUBE	12
3.1.1	Características Principales	12
3.1.2	Ventajas	14
3.1.3	Desventajas	14
3.1.4	Virtualización	15
3.1.5	Retos de la virtualización	16
3.1.6	¿Por qué Virtualizar?	17
3.1.7	Tipos de virtualización	17
3.1.8	Virtualización con VMWare	19
3.1.8.1	Virtualización	19
3.1.8.2	Máquina Virtual	19
3.1.8.3	Guest y Host	19
3.1.8.4	Ventajas de Virtualización	20
3.1.8.5	Snapshot	20
3.1.8.6	Clonación de VMs con VMware	21
3.1.8.7	Tipos de interfaces de red en VMWare	21

3.1.9	OpenVZ	23
3.1.9.1	Virtualización a nivel de SO (virtualización con contenedores)	23
3.1.9.2	OpenVZ.....	24
3.1.9.3	Virtualización y el aislamiento.....	25
3.1.9.4	Manejo de recursos	25
3.1.9.5	Punto restauración	25
3.1.10	Web Panel OpenVZ.....	26
3.1.11	Alta disponibilidad y balanceo de carga	26
3.1.11.1	Alta disponibilidad.....	26
3.1.11.2	Sistemas de archivos de Alta disponibilidad	27
3.1.11.3	Clúster de alta disponibilidad	27
3.1.11.4	Balanceo de carga.....	27
3.1.12	RAID	28
3.1.12.1	Particularidades.....	29
3.1.12.2	Niveles de RAID	30
3.1.12.3	Tolerancia a fallas	36
3.1.12.4	Confiabilidad y Disponibilidad	38
3.1.13	GlusterFS.....	40
3.1.13.1	Ventajas de GlusterFS.....	40
3.1.13.2	Método de acceso a la Data	40
3.1.13.3	Distribución y Replicación de la Data.....	41
3.1.13.4	Componentes de Gluster.....	44
3.1.14	Heartbeat	45
3.1.14.1	Proyectos de alta disponibilidad	46
3.1.14.2	Clúster.....	47
3.1.14.3	Clúster de alta disponibilidad.....	47
3.1.15	LVS (Linux Virtual Server).....	48
3.1.15.1	Técnicas de balanceador de carga con LVS.....	49
3.1.15.2	Técnicas de programación de conexiones.....	52
4	Desarrollo Práctico.....	54
	ORGANIZACIÓN DE LAS PRÁCTICAS	54
	PRÁCTICA 1: REDES DE ORDENADORES EN VMWARE	55
	PRÁCTICA 2: VIRTUALIZACIÓN POR CONTENEDORES OPENVZ.....	59
	PRÁCTICA 3: WEB PANEL OPENVZ	70

PRÁCTICA 4: RAID.....	78
PRÁCTICA 5: GLUSTERFS.....	85
PRÁCTICA 6: HEARTBEAT	89
PRÁCTICA 7: LVS	94
PRÁCTICA 8: CREACIÓN DE UNA PEQUEÑA NUBE	98
5 Conclusiones	102
6 Recomendaciones	103
7 Bibliografía.....	104
8 Anexos	105

Índice de ilustraciones

Figura 1 Etapas del trabajo.....	9
Figura 2 Arquitectura virtualizada.	17
Figura 3 Host ejecutando múltiples aplicaciones.	20
Figura 4 Ilustración de Snapshot	21
Figura 5 Ilustración de maquina en modo Bridge.....	22
Figura 6 Ilustración de maquina en modo Host-only.	23
Figura 7 Ilustración de maquina en modo NAT.....	23
Figura 8 Múltiples VPS con OpenVZ.	24
Figura 9 Aislamiento de contenedores OpenVZ.	25
Figura 10 Ilustración de RAID.....	28
Figura 11 RAID 0 (120GB).	31
Figura 12 RAID 0 (80GB).	31
Figura 13 RAID 1 (60GB).	32
Figura 14 RAID 1 (40GB).	33
Figura 15 RAID 4 (120GB).	33
Figura 16 RAID 5 (120GB).	34
Figura 17 RAID 6 (120GB).	34
Figura 18 RAID 10 (120GB).	35
Figura 19 RAID 50 (240GB).	36
Figura 20 RAID 60 (240GB).	36
Figura 21 Modo de degradación.	37
Figura 22 Host spares.	38
Figura 23 Hot swap.	38
Figura 24 Ilustración de GlusterFS.	40
Figura 25 Distribución y Replicación de la data con: Volumen distribuido.....	41
Figura 26 Distribución y Replicación de la data con: Volumen en réplica.	42
Figura 27 Distribución y Replicación de la data con: Distribuido en Réplica.	42
Figura 28 Distribución y Replicación de la data con: Réplica Distribuida.	43
Figura 29 Distribución y Replicación de la data con: Geo-Replicación.....	44
Figura 30 Distribución y Replicación de la data con: Volumen Stripped.....	44
Figura 31 Funcionamiento de Heartbeat.....	46
Figura 32 Funcionamiento de LVS.	49

Figura 33 Estructura del modo LVS-NAT.....	50
Figura 34 Estructura del modo LVS-TUN.	51
Figura 35 Estructura del modo LVS-DR.....	52
Figura 36 Topología a seguir para la práctica numero 1	56
Figura 37 Topología a seguir para la práctica numero 2.....	60
Figura 38 Topología a seguir para la práctica numero 3.....	71
Figura 39 Topología a seguir para la práctica numero 4.....	79
Figura 40 Topología a seguir para la práctica numero 5.....	86
Figura 41 Topología a seguir para la práctica numero 6.....	90
Figura 42 Topología a seguir para la práctica numero 7.....	95
Figura 43 Topología a seguir para la práctica numero 8.....	99

1 Introducción

El presente documento tiene como propósito desarrollar prácticas de computación en la nube para la carrera de Ingeniería en Telemática (IT), donde se desarrollarán enunciados que abordarán temáticas teórica-práctica y a la vez guiara al estudiante a dar solución de la misma, permitiendo que puedan adquirir los conocimientos y sean puestos en práctica durante el transcurso del componente.

En el documento se expondrán temas, como:

Unidad 1: Virtualización

- Full-Virtualización: virtualización más usada.
 - Tipos de interfaces de red.
 - Crear redes entre los equipos.
 - Clonado.
 - Snapshot.
- Virtualización por contenedores OpenVZ.
 - Conceptos.
 - Red.
 - Creación.
- Virtualización por plantillas.
 - `openvz-web-panel-managing`.

Unidad 2: Alta disponibilidad y balanceo de carga

- RAID
- GlusterFS
- Heartbeat
- LVS

El trabajo monográfico está compuesto por dos partes. La primer parte es donde se desarrollan los contenidos teóricos que serán necesarios para poder dar solución a las prácticas, al igual que el enunciado de cada una de ellas. La segunda parte contiene el desarrollo y solución de cada una de las prácticas propuestas.

Se ha buscado que cada uno de los temas que se abordan en la parte teórica tenga su correspondencia directa en la práctica para que el alumno al finalizar esta asignatura, este capacitado para instalar y configurar.

1.1 Antecedentes

En la actualidad la Universidad, se han elaborado temas monográficos relacionados con las propuestas de prácticas de laboratorio para los componentes relacionados con Redes de Computadoras. A continuación se presentan algunos de estos trabajos:

- **“Propuesta de prácticas de laboratorios de Switches y Routers para la carrera de Ingeniería en Telemática UNAN-León”**, elaborado por el Br Rudy Otoniel Quiróz Vázquez, el Br. Franklin Ernesto Ramírez Medina y el Br. Yoel Francisco Rivera González, en Septiembre del 2013. Este documento propone la elaboración de 15 prácticas de laboratorio actualizadas, con la finalidad de abordar temáticas teóricas y prácticas más avanzadas y más ajustadas a la realidad. Dichas prácticas han sido de gran apoyo sobre todo en los componentes curriculares de Comunicación de Datos y Redes Computadores.
- **“Elaboración de prácticas de Switching, Routing y Servicios de Red con IPv6 para el componente Electiva X correspondiente al plan académico 2011 de la carrera de Ingeniería en Telemática del Departamento de Computación de la UNAN-León”**, elaborado por el Br. Delia María Jaime Toruño, Br. Hugo Mariano García Machado y Br. William Francisco Aguilar Zapata, en Mayo 2015. Este documento tiene como finalidad exponer las características más importantes del protocolo IPv6, a través del desarrollo de contenido teórico y de propuestas de prácticas de laboratorio guiadas, dirigidas a estudiantes del último año de la carrera Ingeniería en Telemática de la UNAN-León.
- **“Plan Docente para el componente Aplicaciones Telemáticas”**, elaborado por el MSc. Denis Leopoldo Espinoza Hernández, Julio 2007. Este documento se aborda el desarrollo del Plan de Docente del componente Aplicaciones Telemáticas que se imparte en el cuarto año de la carrera de Ingeniería en Telemática de la UNAN-León. Además se presenta la situación actual de dicho componente, su relación con otros componentes, la metodología y material didáctico de apoyo para impartirla, el sistema de evaluación, la planificación temporal, la presentación del contenido teórico, el desarrollo de las prácticas y la bibliografía necesaria para las mismas.

Es válido mencionar que existen documentos y sitios web con algunos ejercicios prácticos relacionado con este tema; sin embargo, no existe uno completo por parte de la UNAN-León con amplios contenidos teóricos-prácticos como los que se desarrollan en el presente.

1.2 Planteamiento del problema

El plan de estudio 2011 de la carrera de IT posee una serie de componentes Electivos para ampliar la formación de sus estudiantes. Este componente computación en la nube se ubica en el semestre IX del plan académico antes mencionado. Explica los fundamentos de la computación en la nube abordando la infraestructura que da soporte a todo lo que en ella exista, los servicios que se ofrecen y otros conceptos propios de este tipo de implementaciones.

La micro programación nos indica dentro del componente se guía al alumno para que pueda configurar una pequeña nube en la cual los usuarios creen máquinas virtuales Linux basadas en plantillas haciendo uso de técnicas de virtualización, alta disponibilidad y balanceo de carga.

Este componente se relaciona con los componentes de:

- Redes de Computadores, para realizar correctamente las configuraciones IP de las redes que se crean.
- Gestión de Sistemas UNIX, pues todas las configuraciones se realizan en estos sistemas.
- Administración de Servicios de Red, al realizar la gestión de los servicios que se abordan en esa asignatura.
- Administración de servidores, a la que aporta los conceptos principales de mecanismos para mejorar la disponibilidad de los servidores.

El desarrollo de prácticas de laboratorios, es esencial para la formación académica de los estudiantes de la carrera de IT de la UNAN-León (Universidad). Sin embargo no existe un documento estándar con una secuencia de los contenidos abordados para el componente computación en la nube así como un conjunto de prácticas claramente definidas lo que dificulta su impartición a los estudiantes.

Los aspectos antes mencionados provocan que los docentes, en este momento, no posean un documento base de referencia para la asignación de prácticas de laboratorios relacionadas con computación en la nube, y que los estudiantes no posean una documentación y/o bibliografía para dar una correcta solución de las mismas.

Estos problemas provocan las siguientes interrogantes:

- ¿Cómo se debe desarrollar un plan de prácticas de laboratorio relacionadas con Electiva VIII: Computación en la Nube para la carrera de IT de la Universidad?
- ¿Qué secuencia debe tener el documento de manera que permita a los estudiantes poner en prácticas los conocimientos adquiridos en el transcurso del desarrollo de los laboratorios relacionado con Electiva VIII: Computación en la Nube?
- ¿Qué temas deben ser abordados, donde estos a su vez, sean de importancia en el área de administración de servidores?

1.3 Justificación

Tomando como referencia los problemas antes expuestos, nace la idea de crear un documento, en el cual se plasmarán conocimientos teóricos y se desarrollaran prácticas de laboratorios referentes a computación en la nube. Esto permitirá a los estudiantes de la carrera de Ingeniería en Telemática de la Universidad, adquirir conocimientos que les permita enfrentar desafíos reales.

Por este motivo hemos decidido:

- Proponer prácticas de laboratorio que aumenten gradualmente su complejidad y a fines a algunos escenarios de la vida real.

1.3.1 Originalidad

Existen trabajos y documentos que contienen prácticas a fines a estas áreas en internet; sin embargo el presente trabajo pretende aportar a la universidad una guía propia y completa para el plan académico 2011 del componente computación en la nube de la carrera de Ingeniería en Telemática, por lo que este trabajo ayudará de manera significativa a mejorar la calidad de enseñanza-aprendizaje de este componente.

1.3.2 Alcance

- **Para los docentes:** Tendrán un documento que les permita corroborar aspectos teóricos, a su vez brindara prácticas guiadas a los estudiantes.
- **Para los estudiantes:** Comprender los temas y resolver las prácticas propuestas en el documento, obtener un nivel intermedio acerca de los conceptos y temáticas sobre Computación en la nube en el avance del curso.

1.3.3 Producto

El documento presenta las características siguientes:

- **Guiado:** Describirá paso a paso la forma en la que se deberá realizar cada práctica.
- **Sencillo:** Porque el desarrollo de las prácticas y la documentación, serán elaboradas de forma que se comprenda fácilmente.
- **Secuencial:** Las prácticas se proponen de menor a mayor grado de complejidad, de tal forma que el estudiante pueda lograr obtener el aprendizaje necesario.

1.3.4 Impacto

El documento permitirá contar con un material de apoyo eficaz tanto para el personal docente, como para estudiante y público en general, sobre Computación en la Nube. Con esto, estamos contribuyendo a que el estudiante tenga las herramientas necesarias para enfrentar los retos que se presentan en Empresas, Industrias, Organizaciones en la actualidad.

1.4 Objetivos

1.4.1 Objetivo general

Crear propuestas de prácticas de laboratorio de Computación en la Nube, con la finalidad de usarlo en el componente Electiva VIII: Computación en la nube, correspondiente al Plan Académico 2011 de la carrera Ingeniería en Telemática del Departamento de Computación de la UNAN-León.

1.4.2 Objetivos específicos

- Desarrollar los temas seleccionados que servirán a los estudiantes para adquirir conocimientos teóricos-prácticos, que sean útiles para la elaboración de las prácticas.
- Definir el formato que regirán el enunciado de las prácticas de laboratorios a desarrollar, en base a formatos antes usados en diferentes componentes de la carrera.
- Ordenar las prácticas de acuerdo con la complejidad que presentan cada una ellas, abordando temas de nivel básico, medio y avanzado, con contenidos como virtualización, balanceo de carga y alta disponibilidad.

2 Diseño Metodológico

La realización de este tema monográfico se realizó en diversas etapas, en las cuales se muestra los pasos para la concepción de este estudio en el que definimos que se trata de un tipo de investigación aplicada pues lo que se proponga y desarrolle en el presente documento se aplicará al curso de computación en la nube.



Figura 1 Etapas del trabajo.

2.1 Recolección de Información

En la primera etapa de la investigación, se realizó un estudio sobre computación en la nube, con la finalidad de determinar los aspectos más importantes a desarrollar en el tema monográfico, organizando la información según el nivel de complejidad que tiene cada uno de los temas a desarrollar.

2.2 Selección de la herramienta a implementar

En esta etapa se seleccionó la plataforma a usar para el desarrollo de las prácticas de computación en la nube. Después de analizar las herramientas tomando en cuenta las tecnologías soportadas por esta, se seleccionó las siguientes:

- VMWare
- Ubuntu
- CentOS
- OpenVZ
- OpenVZ Web Panel
- GlusterFS
- Heartbeat
- LVS
- RAID

2.3 Elaboración y desarrollo de los laboratorios

- **Organización de las prácticas:** Es el punto donde la información es organizada según el nivel de complejidad que tienen cada uno de los temas a desarrollar tanto teóricos como prácticos.
- **Formato del enunciado de prácticas:**

Título: Nombre de la práctica.

Introducción: Contiene en aspectos generales de lo que posee cada práctica, y en algunos casos, aspectos claves que los estudiantes deben tomar en cuenta para facilitar la solución de las mismas.

Objetivos

- Presenta una visión general de lo que se espera lograr con el desarrollo de la práctica.
- Expondrá aspectos específicos, en los cuales los estudiantes deberán de enfocar su trabajo de laboratorio.

Requerimientos

- **Hardware:** Describe las características de la computadora que se usará en la realización de las prácticas.
- **Software:** Detalla el simulador o emulador en que se desarrollara la práctica.

Duración de la práctica: Tiempo estimado en horas presenciales y no presenciales para dar solución a cada práctica, con la salvedad que una vez que sea usada por diferentes maestros puede variar de acuerdo al criterio personal de evaluación de cada maestro.

Topología: Se expondrá una imagen donde se represente la topología correspondiente a la práctica.

Comandos de ayuda: Cuadro donde se lista los comandos que ayudara a la realización eficaz de la práctica.

Enunciado: Se describe los pasos a seguir para la realización de la práctica.

Preguntas de control: Se evalúa grado de asimilación y comprensión de los conceptos básicos y configuraciones realizadas después de resolver cada práctica.

3 Marco Teórico

3.1 Computación en la nube

La computación en la nube (*cloud computing*) consiste en la posibilidad de ofrecer servicios a través de Internet. La computación en nube es una tecnología nueva que busca tener todos nuestros archivos e información en Internet.

La característica básica de la computación en la nube es que los recursos y servicios informáticos, tales como infraestructura, plataforma y aplicaciones, son ofrecidos y consumidos como servicios a través de la Internet sin que los usuarios tengan que tener ningún conocimiento de lo que sucede detrás. Esto debido a que los usuarios no tienen idea alguna sobre la infraestructura que opera para ofrecer los servicios es que se llama Computación en las Nubes.

3.1.1 Características Principales

- **Auto-Servicio bajo demanda:** El consumidor podrá aprovisionar recursos computacionales en forma unilateral, según lo requiera, y sin requerimiento de interacción humana con el proveedor del servicio.
- **Permitir el acceso desde la red (pública, privada, híbrida, comunitaria):** Todos los recursos que ofrece la nube están disponibles en la red, y el consumidor no sólo puede acceder a ellos a través de mecanismos estándar, sino que también mediante plataformas heterogéneas como teléfonos móviles, laptops, PDAs, etc.
- **Asignación de recursos en modo multiusuario:** A diferencia de las aplicaciones de software tradicionales, en el cloud computing el proveedor tiene una única aplicación que abre a todos los usuarios que desean utilizarla, estableciendo unos recursos de acceso y prestaciones distintos para cada usuario. Al ser aplicaciones multiusuario, puede hacer miles de internautas utilizando la misma herramienta a la vez, cada uno con las mismas o distintas prestaciones.
- **Capacidad de rápido crecimiento:** las unidades de capacidad pueden ser rápidas y fácilmente aprovisionadas (en algunos casos en forma automática), escaladas (crecimiento) o liberadas. Para el consumidor, estos recursos suelen parecer ilimitados, y pueden ser adquiridos en cualquier cantidad y momento.

- **Servicio medido:** Los sistemas de la nube controlan de forma automática y optimizada la utilización de los recursos. Este uso de los recursos puede ser monitoreado y controlado, además, es posible realizar reportes para ambas partes, a fin de establecer la facturación del servicio. El internauta puede en todo momento decidir qué aplicaciones usar y elegir entre aquellas que son gratuitas y las que no lo son. En el caso de las aplicaciones de pago, el coste irá en función de diversas variables, como el servicio contratado, el tiempo que se ha usado ese servicio, el volumen de tráfico de datos utilizado, etc.
- **Elasticidad y escalabilidad:** Las aplicaciones en cloud son totalmente elásticas en cuanto a su rapidez de implementación y adaptabilidad. Además, son totalmente escalables, es decir, hoy podemos estar utilizando solo un 10% del total de la aplicación y mañana podemos acceder al 80% de la misma con total normalidad y rapidez, con tan solo comunicarlo a nuestro proveedor y modificar nuestra tarifa de suscripción.
- **Seguridad:** Cuando se habla de “aplicaciones en Internet”, no se debe entender que nuestros datos están sueltos en la red. Quizás este es el mayor miedo que tienen las empresas y por eso creo que es conveniente explicarlo.

Los datos, cuando están en aplicaciones en cloud, se alojan en DATA CENTERS, empresas específicamente dedicadas a la custodia y salvaguarda de datos de empresas de todo tipo: bancos, entidades financieras, gobierno, multinacionales, pymes, personas como vosotros o como yo.... Son empresas que cuentan con todas las medidas de seguridad necesarias, tanto físicas como de software, de forma que no haya jamás una pérdida de información ni de integridad de los datos.

La única precaución que hay que tener, es encontrar un DATA CENTER o proveedor que nos dé garantías y prestaciones adecuadas al “valor” que damos a nuestros datos. Y eso, es algo que ya hacemos a día de hoy con otros servicios como Internet, teléfono, etc.

3.1.2 Ventajas:

- Integración probada de servicios Red. Por su naturaleza, la tecnología de cloud computing se puede integrar con mucha mayor facilidad y rapidez con el resto de sus aplicaciones empresariales ya sean desarrolladas de manera interna o externa.
- Prestación de servicios a nivel mundial. Las infraestructuras de cloud computing proporcionan mayor capacidad de adaptación, recuperación de desastres completa y reducción al mínimo de los tiempos de inactividad.
- Una infraestructura 100% de cloud computing permite al proveedor de contenidos o servicios en la nube prescindir de instalar cualquier tipo de hardware, ya que éste es provisto por el proveedor de la infraestructura o la plataforma en la nube. El hecho de que requiera mucha menor inversión para empezar a trabajar.
- Implementación más rápida y con menos riesgos. Podrá empezar a trabajar muy rápidamente gracias a una infraestructura de cloud computing. Sus aplicaciones en tecnología de cloud computing estarán disponibles en cuestión de días u horas en lugar de semanas o meses, incluso con un nivel considerable de personalización o integración.
- Actualizaciones automáticas que no afectan negativamente a los recursos de TI. Si actualizamos a la última versión de la aplicación, nos veremos obligados a dedicar tiempo y recursos (que no tenemos) a volver a crear nuestras personalizaciones e integraciones. La tecnología de cloud computing no le obliga a decidir entre actualizar y conservar su trabajo, porque esas personalizaciones e integraciones se conservan automáticamente durante la actualización.
- Contribuye al uso eficiente de la energía. En este caso, a la energía requerida para el funcionamiento de la infraestructura. En las nubes, la energía consumida es sólo la necesaria, reduciendo notablemente el desperdicio.

3.1.3 Desventajas:

- La centralización de las aplicaciones y el almacenamiento de los datos origina una interdependencia de los proveedores de servicios.
- La disponibilidad de las aplicaciones está ligada a la disponibilidad de acceso a Internet.

- Los datos "sensibles" del negocio no residen en las instalaciones de las empresas, lo que podría generar un contexto de alta vulnerabilidad para la sustracción o robo de información.
- La confiabilidad de los servicios depende de la "salud" tecnológica y financiera de los proveedores de servicios en nube. Empresas emergentes o alianzas entre empresas podrían crear un ambiente propicio para el monopolio y el crecimiento exagerado en los servicios.
- La disponibilidad de servicios altamente especializados podría tardar meses o incluso años para que sean factibles de ser desplegados en la red.
- La madurez funcional de las aplicaciones hace que continuamente estén modificando sus interfaces, por lo cual la curva de aprendizaje en empresas de orientación no tecnológica tenga unas pendientes significativas, así como su consumo automático por aplicaciones.
- Seguridad. La información de la empresa debe recorrer diferentes nodos para llegar a su destino, cada uno de ellos (y sus canales) son un foco de inseguridad. Si se utilizan protocolos seguros, HTTPS por ejemplo, la velocidad total disminuye debido a la sobrecarga que estos requieren.
- Escalabilidad a largo plazo. A medida que más usuarios empiecen a compartir la infraestructura de la nube, la sobrecarga en los servidores de los proveedores aumentará, si la empresa no posee un esquema de crecimiento óptimo puede llevar a degradaciones en el servicio o jitter altos.
- Velocidad de respuesta, sobre todo en sistemas críticos o con grandes volúmenes de información. Lo normal es de momento, no utilizar la nube para sistemas críticos.
- En caso de fallo de proveedor, existen dificultades para restablecer el servicio, migrarlo a otro proveedor o establecerlo en local.

3.2 Virtualización

En términos generales consiste en crear/establecer/simular un recurso que no existe en el mundo real/actual. Dentro de un escenario virtualizado este recurso creado es indistinguible de un recurso real. En términos de computación no es un concepto nuevo, como ejemplos:

- La memoria virtual pretende que desde el punto de vista de la aplicación la memoria es ilimitada.
- Un ambiente multi-tarea da la impresión de múltiples procesos ejecutándose simultáneamente.

En nuestro caso nos compete la virtualización de sistemas en el cual todo el computador es simulado:

- Sobre una infraestructura existente se crea una o múltiples máquinas virtuales (en inglés virtual machines – VM).
- Las VM's tienen hardware del mismo tipo (en ocasiones en cantidades diferentes).
- Que el hardware virtual sea similar al hardware físico es una diferencia importante con respecto a la emulación.

Mientras más similar el hardware virtual al físico mayor rendimiento del equipo físico. Para el despliegue de los nodos que prestan un servicio dado, se presentan varios retos entre ellos tenemos:

- Respaldo de Configuración.
- Manejo de los nodos.
- Sistemas de actualización.
- Respaldo de los datos.
- Migración de hardware (por falla o por actualización).
- Consumo de energía.
- Altos costos del hardware.
- Heterogeneidad.
- Tiempo de despliegue.
- Espacio físico para el almacenamiento de los nodos.
- Aislamiento.
- Flexibilidad.
- Escalabilidad.
- Baja utilización de los recursos desplegados.
- Disponibilidad, integridad y autenticidad.

3.2.1 Retos de la virtualización

- Alta utilización de los recursos de hardware.
- Necesidad de los sistemas de archivos distribuidos para albergar las imágenes de las VMs.

3.2.2 ¿Por qué Virtualizar?

Históricamente: Administradores de servicios, vendedores de software, otros, recomiendan e incluso exigen el uso exclusivo de un servidor por cada aplicación.

- Una aplicación por cada servidor virtual (VM's)
- Múltiples VM's por cada anfitrión (en inglés *hosts*)



Figura 2 Arquitectura virtualizada.

Entre las bondades de virtualizar encontramos:

- Correr múltiples VM en un solo servidor físico.
- En ocasiones la suma de los recursos asignados a las VMs puede ser mayor que los recursos del anfitrión (en inglés *overcommitment*).
- Facilidad en el despliegue de nuevos servidores.
- Facilidad para copiar o básicamente migrar VMs.
- En frío cuando la VM no está corriendo.
- En caliente cuando la VM se está ejecutando.

3.2.3 Tipos de virtualización

Desde el punto de vista del OS invitado (en inglés *guest*) tiene completo control del hardware sobre el que se está ejecutando.

Virtualizar es básicamente abstraer al *guest de* los recursos físicos. Según como se realiza esta abstracción encontramos:

- Virtualización completa (full virtualization).
- Paravirtualización (paravirtualization).
- Contenedores (OS virtualization).

Virtualización completa (full virtualization)

- Una de las más utilizadas.
- El OS no tiene idea alguna de estar corriendo en una plataforma virtual (no es necesario cambios en el software).
- Existe un hypervisor entre el OS guest y el hardware del hosts.
- El hypervisor se encarga de emular el hardware que necesita el guest.
- Si el guest requiere ejecutar una instrucción privilegiada el hypervisor se encarga de gestionarla adecuadamente.
- El guest puede ser para arquitecturas diferentes a la arquitectura del hosts (...este cambio de un tipo de instrucciones a otras genera un alto impacto en el rendimiento...)
- Cada VM tiene entonces su propio OS.
- Algunos ejemplos: QUEMU, XEN, Vmware ESX, Virtualbox...

Paravirtualización (paravirtualization)

- También el guest ejecuta su propio OS pero es necesario modificarlo.
- También existe un hypervisor aunque un poco más ligero.
- A diferencia que full virtualization solo las instrucciones privilegiadas son enviadas al hypervisor. El resto van directas al hardware.
- Lo anterior crea un menor impacto en el rendimiento y mayor aprovechamiento de recursos.
- XEN es un ejemplo de virtualización paravirtualizada.

Contenedores (OS virtualization)

- Ambientes de ejecución aislados entre sí que comparten un mismo kernel.

- Cada contenedor se puede configurar para asignar los recursos que usara en la ejecución de sus procesos (estos recursos los comparte con los demás contenedores)
- Los recursos de hardware no son virtualizados (el proceso en ejecución accede directo al hardware)
- No permite el uso de OS diferentes entre el guest y el host.
- La Virtualización a nivel de SO ha sido diseñada para proveer el aislamiento que se requiere sin perder rendimiento.
- OpenVZ es un ejemplo de OS virtualization.

3.3 Virtualización con VMWare

3.3.1 Virtualización

Combinación de hardware y software que permite a un recurso físico funcionar como múltiples recursos lógicos. Se encarga de crear una interfaz externa que encapsula una implementación subyacente mediante la combinación de recursos en localizaciones físicas diferentes.

3.3.2 Máquina Virtual

Simula una plataforma de hardware autónoma incluyendo un sistema operativo completo que se ejecuta como si estuviera instalado.

Es posible virtualizar el hardware de servidor, el software de servidor, virtualizar sesiones de usuario, virtualizar aplicaciones y también se pueden crear máquinas virtuales en una computadora de escritorio.

3.3.3 Guest y Host

El sistema huésped o guest es el sistema operativo virtualizado por el software, y pueden haber varios sistemas huésped en un solo sistema anfitrión. Los sistemas huésped no tienen por qué interferir entre ellos ni con el sistema anfitrión.

El Host o anfitrión es el sistema operativo de la maquina real.



Figura 3 Host ejecutando múltiples aplicaciones.

- La máquina virtual se comporta como una aplicación más en el host.
- El guest percibe la máquina virtual como si fuera hardware real.

3.3.4 Ventajas de Virtualización

- Permite el aislamiento de aplicaciones/servicios y usuarios sobre la misma máquina para que no interfieran entre sí.
- Se aprovecha más la infraestructura hardware existente.
- Se pueden ejecutar varios servidores/servicios en una misma máquina (email, DNS, web, BBDD).
- Respaldo (backup) de máquinas completas, datos y ante diversos problemas.
- Independencia del hardware.

3.3.5 Snapshot

Un Snapshot (o imagen instantánea) de un disco es una copia del archivo de disco de la máquina virtual (VMDK) en un momento concreto. Conserva el sistema de archivos del disco y la memoria del sistema de nuestra VM, permitiéndonos volver a esa imagen guardada (similar a un punto de restauración). Al crear un Snapshot, lo que anteriormente era grabable se convierte en archivo de solo lectura desde ese momento. Se crean nuevos archivos que contienen los cambios efectuados al archivo original (disco virtual, vmdk).

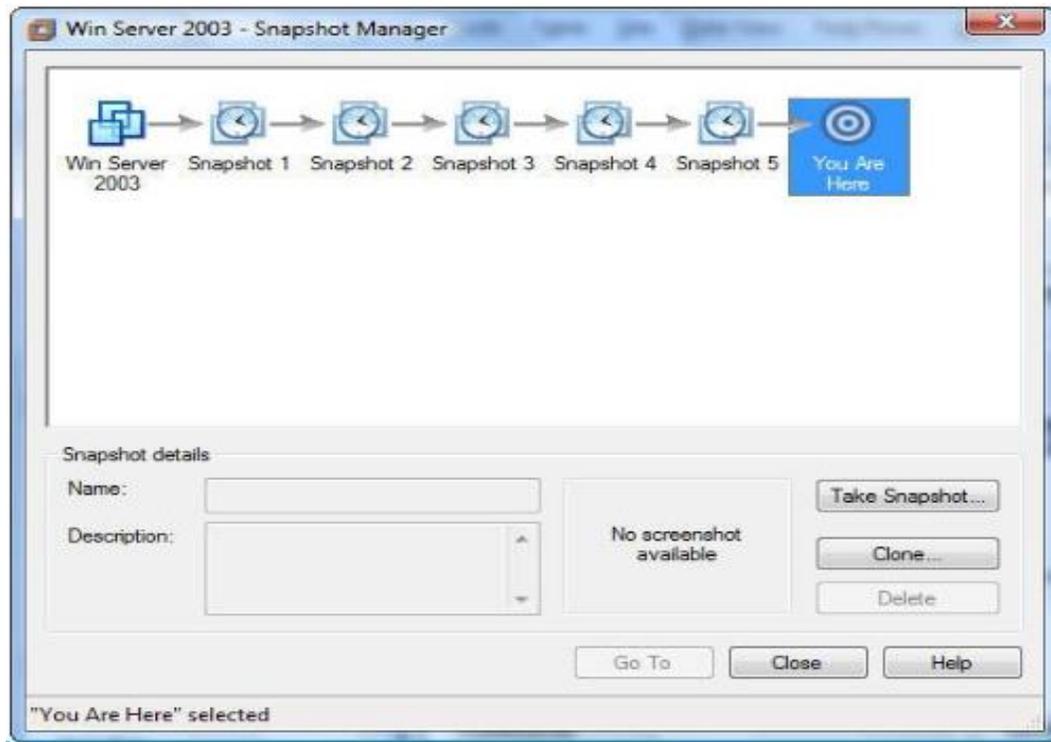


Figura 4 Ilustración de Snapshot .

El tamaño de un archivo snapshot no puede rebasar nunca el tamaño del archivo del disco original. Hay un overhead adicional (espacio extra) en el disco que contiene la información utilizada para gestionar los snapshots.

3.3.6 Clonación de VMs con VMware

Linked Clone: Es una copia de una máquina virtual que comparte los discos virtuales con la máquina virtual padre (VM Base) de manera permanente. Esto ahorra espacio en disco, y permite que múltiples máquinas virtuales utilicen la misma instalación del software. Un Linked Clone debe tener acceso continuo a las VM padres. Sin acceso a la VM padre, un Linked Clone está desactivado.

Full Clone: Es una copia independiente de una máquina virtual, no comparte nada con la máquina virtual padre después de la operación de clonación. Operación en curso de un Full Clone es totalmente independiente de la máquina virtual principal.

3.3.7 Tipos de interfaces de red en VMWare

Bridge: La máquina virtual se muestra como si se tratase de otro equipo físico más en la red. Es el tipo de interfaz de red puesto por defecto cuando creamos una máquina virtual. La máquina virtual será totalmente independiente en la red. Será como un equipo más. Si la máquina real está configurada para recibir una ip por DHCP, la máquina virtual recibirá ip del mismo servidor DHCP.



Figura 5 Ilustración de maquina en modo Bridge.

Host-only: La máquina virtual sólo puede acceder al equipo y a otras máquinas virtuales de la red VMWare. En el modo host-only la máquina virtual no solo está protegido de tu red de área local, sino que está totalmente aislada de ella. La red de la máquina virtual está totalmente dentro de la máquina real. Al igual que en el modo NAT (el cual veremos a continuación), la máquina virtual recibirá una ip del servidor DHCP que funciona dentro de la aplicación de virtualización (tanto virtual box como VMWare).

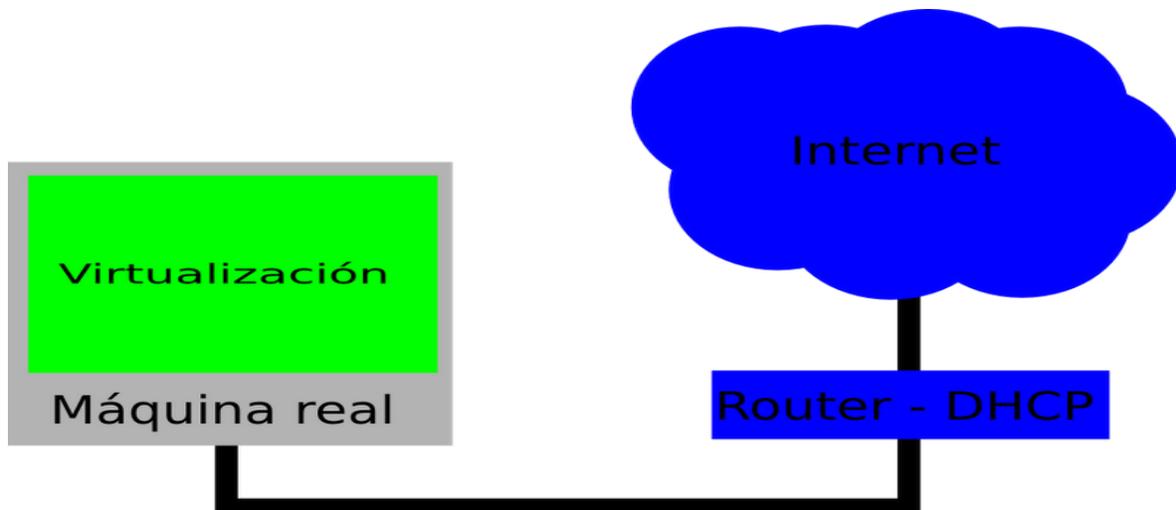


Figura 6 Ilustración de maquina en modo Host-only.

NAT: La máquina virtual se esconde detrás de la IP de la máquina real. Las máquinas virtuales en la misma dirección de red pueden acceder a ella directamente. Cuando la máquina virtual intenta comunicarse con la red real o Internet, lo hace a través de un Firewall propio dentro de la aplicación de virtualización ya que no se encuentra dentro de la red de la máquina real. La máquina virtual recibirá una dirección IP de un servidor DHCP dentro del equipo real. Sin embargo, el que pide la ip será el firewall dentro de la aplicación de virtualización, que sustituye a tu máquina virtual. Así, el que se encarga de comunicarse con la red fuera de tu equipo será tu firewall, no tu máquina virtual. Visto de otro modo, la máquina virtual comparte la ip del anfitrión.

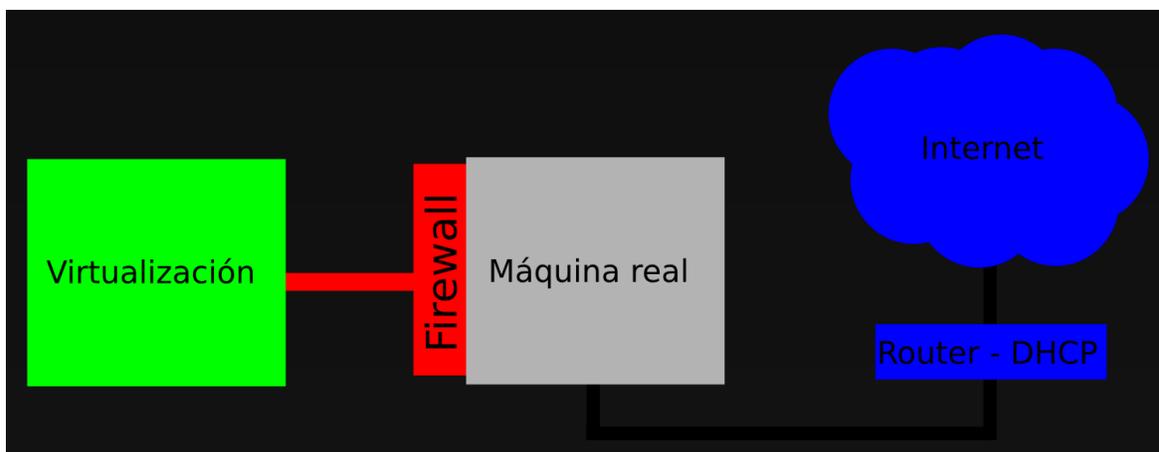


Figura 7 Ilustración de maquina en modo NAT.

3.4 OpenVZ

3.4.1 Virtualización a nivel de SO (virtualización con contenedores)

- Múltiples aplicaciones pueden ser ejecutadas en el mismo servidor, sin embargo se presentan retos, tales como la falta de aislamiento entre las aplicaciones. Para afrontar estos retos múltiples instancias del sistema operativo pueden ser ejecutadas al mismo tiempo sobre el mismo hardware.
- La Virtualización a nivel de SO ha sido diseñada para proveer el aislamiento que se requiere sin perder rendimiento.
- La aplicación por excelencia utilizada para la Virtualización a nivel de SO es OpenVZ.

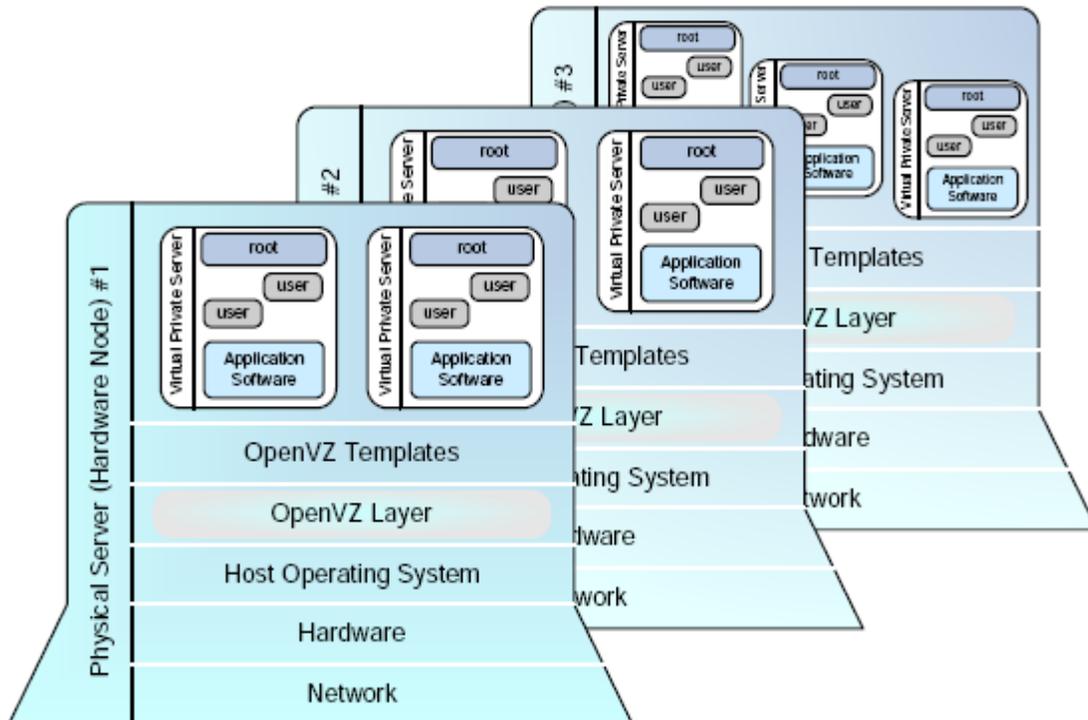


Figura 8 Múltiples VPS con OpenVZ.

3.4.2 OpenVZ

OpenVZ es una tecnología de virtualización en el nivel de sistema operativo para Linux. OpenVZ permite que un servidor físico ejecute múltiples instancias de sistemas

operativos aislados, conocidos como Servidores Privados Virtuales (SPV o VPS en inglés) o Entornos Virtuales (EV).

Ambiente virtual (Virtual Environment VE) también conocido como contenedor, es un ambiente de ejecución aislado el cual desde su punto de vista se está ejecutando en un servidor dedicado. Al no tener que virtualizar un sistema operativo completo (con su propio kernel, sistema de entrada/salida, etc) el consumo de la CPU es mucho menor. La memoria se asigna de forma dinámica.

Un VE tiene sus propios archivos de configuración, sistema de archivos, usuarios, y sus propios ejecutables, adicionalmente a esto el VE posee su propia interfaz de red, y pila de protocolos. Múltiples instancias de las VE pueden ser ejecutadas al mismo tiempo en un mismo hardware, sin embargo todas las instancias deben correr el mismo kernel.

El kernel para ejecutar OpenVZ debe ser modificado en orden de poder proveer la virtualización y el aislamiento, manejo de recursos y puntos de restauración.

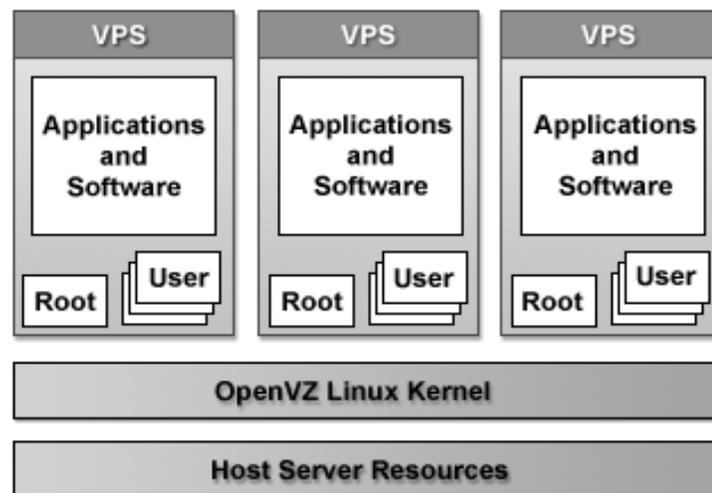


Figura 9 Aislamiento de contenedores OpenVZ.

3.4.3 Virtualización y el aislamiento

Cada máquina virtual debe poseer su propio set de recurso el cual es proveído por el kernel. Estos recursos son:

- Sistema de archivos.

- Usuarios y grupos.
- Árbol de procesos.
- Dispositivos.
- Objetos IPC.

3.4.4 Manejo de recursos

El subsistema de Manejo de recursos está dividido en tres instancias:

- Dos niveles de cuota de disco.
- CPU scheduler.
- User Beancounters: Esto limita y garantiza que ningún VE consuma todos los recursos del sistema.

3.4.5 Punto restauración

Esto permite al sistema crear puntos de restauración debido a que todas las variables y estado del sistema se pueden guardar en el disco duro, esto también permite que el sistema pueda ser migrado en vivo.

3.4.6 Web Panel OpenVZ

Web panel OpenVZ es una interfaz basada en Web GUI para el control del hardware y servidores virtuales con la tecnología de virtualización OpenVZ. Muy simple, sólo para hacer la gestión OpenVZ más fácil. Utilizando web panel OpenVZ, cualquier persona puede crear, eliminar, administrar máquinas virtuales a través de un navegador web en cuestión de minutos sin necesidad de conocer todos los comandos OpenVZ.

OpenVZ es un sistema de virtualización basado en contenedores. Permite que un servidor físico ejecute múltiples instancias de sistemas linux aislados, se les conoce como VPS (Virtual Private Server).

Administrar un servidor OpenVZ no es muy complejo pero requiere de un uso extensivo de un shell de linux pues sus herramientas son un conjunto de scripts.

Desde una sola interfaz de OpenVZ Web Panel podemos controlar varios nodos hardware real, es decir, varios servidores físicos. Al crear los servicios virtualizados, un asistente nos permite descargar las plantillas de SO desde el repositorio oficial de OpenVZ o desde el repositorio de contribuciones.

Se pueden crear copias de seguridad de las máquinas virtuales, asignar rangos de IP, pausar, parar, arrancar, todo lo que haría cualquier otro panel web para OpenVZ.

3.5 Alta disponibilidad y balanceo de carga

3.5.1 Alta disponibilidad

La alta disponibilidad está basada en la replicación de elementos. Se logra con:

- Comunicaciones redundantes.
- Acceso redundante a la data.
- Mecanismo de detección de fallas.
- Mecanismo de contención de fallas.

El proyecto de Linux-HA (High-Availability Linux) es un proyecto dedicado a crear soluciones de alta disponibilidad para Linux, el mismo se basa en promover la confiabilidad, disponibilidad.

HA-linux capacidades:

- Puede usar conexiones seriales, UDP bcast, mcast, ucast comm.
- Conmutación por error en caso de fallo del nodo.
- Conmutación por error en la pérdida de la conectividad IP.
- Herramientas de línea de comandos administrativos para conmutar por error, el estado actual de consultas, etc.
- Activo/Activo o Activo/Pasivo.
- Utiliza herramientas externas para el monitoreo de los recursos.
- Monitoreo SNMP.

3.5.2 Sistemas de archivos de Alta disponibilidad

Para lograr que un sistema tenga alta disponibilidad los datos deben ser compartidos entre todas las máquinas que conforman el clúster. De esto surge la necesidad de un Sistema de archivos de Alta disponibilidad.

Un clúster son dos o más sistemas informáticos que mediante la utilización de componentes de hardware comunes con la principal característica de que se comportan o funcionan como si fuesen uno. Estos ordenadores pueden estar unidos mediante una red

informática. Los clúster se utilizan para mejorar el rendimiento y la disponibilidad de los servicios ofrecidos.

3.5.3 Clúster de alta disponibilidad

- Es un grupo de servidores los cuales cooperan entre ellos para proveer un servicio, incluso al momento de la falla de un componente.
- Cuando un nodo pierde conectividad, los otros retoman su rol.
- Esto involucra apropiarse de la ip y de los servicios que proveía el nodo que perdió conectividad.

3.5.4 Balanceo de carga

El balanceo de la carga es una técnica que se utiliza para compartir el trabajo entre varios procesos, ordenadores, discos u otros recursos. El balanceo de carga se puede hacer mediante diferentes algoritmos. Una vez que los datos están disponibles en el clúster, otro punto importante es el hecho de que los servicios deben responder de forma oportuna, para ello se deben asignar las consultas a cada nodo disponible (balanceo de carga). De forma ideal el balanceo de carga se debe basar en las capacidades de respuesta y la carga de cada nodo del clúster.

3.5.5 RAID

En términos generales y partiendo de su acrónimo en inglés Redundant Array of Independent Disks (RAID) es un sistema que permite combinar el almacenamiento de un grupo de dispositivos independientes, en una única unidad virtual de almacenamiento o RAID múltiples unidades virtuales.



Figura 10 Ilustración de RAID.

RAID es el método que se usa para expandir información en diversos discos utilizando técnicas como el vaciado del disco (RAID Nivel 0), la creación de réplicas del disco (RAID nivel 1) y el vaciado del disco con paridad (RAID Nivel 5) para obtener

redundancia, menos latencia y/o aumentar el ancho de banda para leer o escribir en discos y maximizar así la posibilidad de recuperar información cuando el disco duro no funciona.

RAID está basado en el concepto de que los datos tienen que distribuirse en cada conjunto de discos de manera consistente. Para ello, los datos se rompen en pedazos o grupos de datos con un tamaño que varía normalmente entre 32K y 64K aunque se pueden usar otros tamaños.

3.5.5.1 Particularidades

- Mayor rendimiento y confiabilidad mediante lectura/escritura simultánea de datos en múltiples dispositivos (físicos o virtuales).
- Diferentes esquemas de lectura/escritura conocidos como niveles (RAID levels).
- El nivel a elegir depende de necesidades en cuanto a:
 - Rendimiento y redundancia.
 - Costos de hardware.
 - Capacidad almacenamiento (escalabilidad).
- Soluciones de RAID pueden estar basadas en hardware especializado o herramientas de software.

Hardware RAID

- El sistema basado en el hardware gestiona el subsistema independientemente de la máquina y presenta a la máquina un único disco por conjunto de discos RAID.
- Los discos se conectan a la controladora RAID hardware.
- En el manejador de la controladora se define el nivel y modo de funcionamiento del RAID.
- Tienen lectura escritura en paralelo. Memoria caché para lectura escritura.

Software RAID:

- El software RAID implementa los diversos niveles de RAID en el código del kernel (dispositivo de bloque).
- Ofrece la solución más barata ya que las tarjetas de controladores de disco o los chasis "hot-swap" son bastante caros. (permite quitar un disco duro sin tener que apagar el ordenador) no son requeridos.
- El software RAID también funciona con discos IDE más baratos así como también con discos SCSI.

- Con los CPUs rápidos de hoy en día, el rendimiento del software RAID aumenta considerablemente con respecto al hardware RAID.

3.5.5.2 Niveles de RAID

Diferentes aplicaciones requieren implementaciones de diferentes estructuras de RAID. Estas diferentes estructuras se conocen como niveles.

Diferentes niveles de RAID ofrecen diversidad de compromiso entre rendimiento y redundancia.

La selección del nivel adecuado requiere un alto entendimiento de las necesidades de sus aplicaciones y usuarios (incluyendo escalabilidad). Por ejemplo es posible que según sus necesidades deba sacrificar rendimiento para lograr un RAID de mayor redundancia.

Conceptos relacionados con RAID

Striping: Técnica de segmentación lógica de los datos que luego son accedidos de forma secuencial en diferentes dispositivos de almacenamiento. Provee un mayor rendimiento en la velocidad de acceso a los datos almacenados en múltiples dispositivos. La falla de un dispositivo causa la pérdida de todos los datos.

Mirroring: Técnica en la cual se crean réplicas en tiempo real del volumen lógico de un dispositivo en otros dispositivos físicos. Ofrece alta disponibilidad de los datos debido a las múltiples réplicas de los datos (redundancia). De forma adicional provee mejoras en el rendimiento de acceso a los datos (lectura) en múltiples dispositivos. Esta técnica representa un alto costo.

Parity: Se genera un conjunto de datos de redundancia a partir de dos o más conjuntos de datos primarios aplicando la función Booleana XOR. Con los datos de redundancia se puede reconstruir los datos de algunos de los conjuntos de datos primarios. Aunque no implica duplicar por completo los datos primarios, esta técnica puede ocasionar bajo rendimiento en la velocidad de escritura en un RAID.

RAID 0 (disk striping)

- Cada dispositivo del RAID se divide en segmentos de tamaño similar (ej. entre 8 KB a 1024 KB).
- Estos segmentos son intercalados de manera secuencial y repetida.

- El espacio de almacenamiento está compuesto por segmentos de todo el grupo de dispositivos.
- Ofrece un alto rendimiento ya que múltiples dispositivos son accedidos (lectura/escritura) simultáneamente.

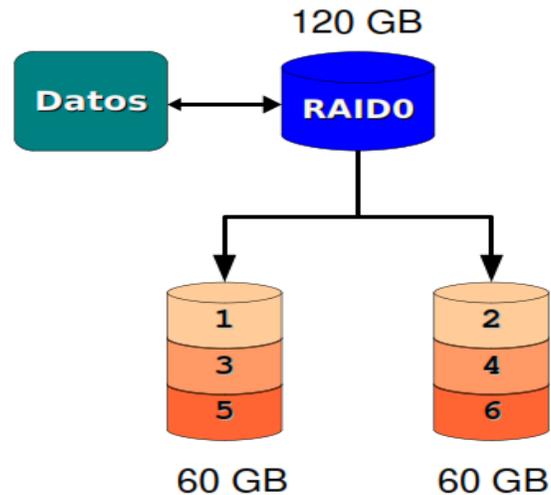


Figura 11 RAID 0 (120GB).

- La cantidad total de almacenamiento es la suma de la capacidad de todos los dispositivos del grupo.
- Se pueden usar dispositivos de diferentes tamaños, recordando que el dispositivo de menor tamaño limita la cantidad de espacio usado en los demás dispositivos.
- Un daño en cualquiera de los dispositivos hará inutilizable el RAID.

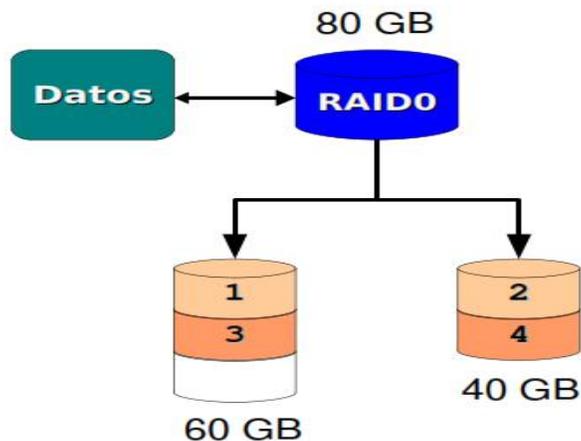


Figura 12 RAID 0 (80GB).

RAID 2 (bit striping)

- Divide los datos a nivel de bits distribuyéndolos entre los dispositivos del arreglo.
- Hace uso de código Hamming para el chequeo de paridad.
- La segmentación a nivel de bits crea un alto impacto (lectura/escritura) en los recursos del sistema lo que lo hace inviable a nivel práctico.

RAID 3 (byte striping)

- Divide los datos a nivel de byte distribuyéndolos entre los dispositivos del arreglo.
- De forma similar que RAID 2 la segmentación de datos a nivel de byte crea un alto impacto (lectura/escritura) en los recursos del sistema haciendo inviable su implementación.

RAID 1 (disk mirroring)

- Todos los datos escritos son duplicados (replica) en cada dispositivo del RAID.
- Según lo anterior ofrece 100% de redundancia.
- Un alto rendimiento ya que lo conforman múltiples dispositivos que pueden ser accedidos (lectura) mientras uno o más están ocupados.

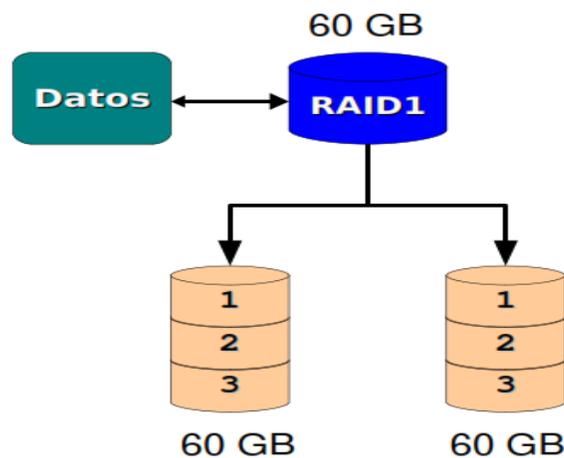


Figura 13 RAID 1 (60GB).

- La cantidad total de almacenamiento es igual al tamaño del dispositivo de menor capacidad.
- El uso de dispositivos de capacidad similar proporcionan un RAID óptimo.
- Es un esquema de alto costo ya que cada dispositivo debe ser duplicado.

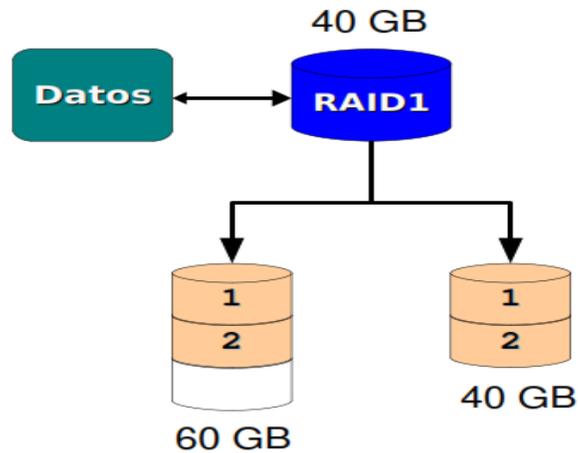


Figura 14 RAID 1 (40GB).

RAID 4 (block striping & dedicated parity)

- Segmenta los datos a nivel de bloques distribuyéndolos entre los dispositivos del arreglo.
- Dedicar por completo uno de los dispositivos del arreglo para paridad.
- Es un esquema similar a RAID2 y RAID3 pero la división en bloques evita un alto impacto (lectura/escritura) en los recursos del sistema.

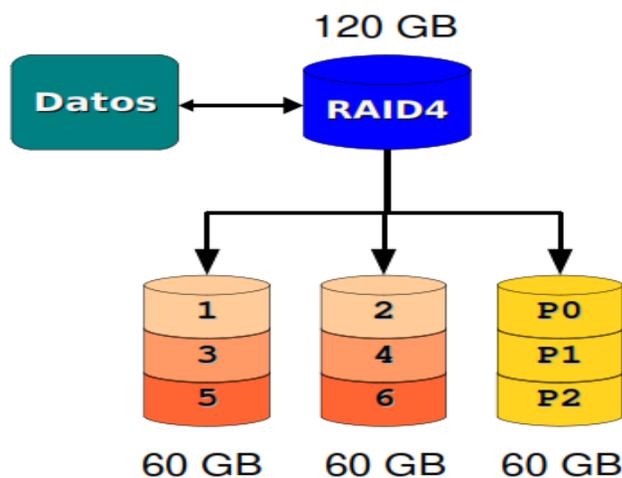


Figura 15 RAID 4 (120GB).

RAID 5 (block striping & distributed parity)

- Segmenta los datos a nivel de bloques distribuyéndolos entre los dispositivos del arreglo de forma similar a RAID 4.
- Distribuye los datos de paridad entre todos los dispositivos del arreglo.

- RAID 4 y RAID 5 proveen redundancia ante la falla de un dispositivo en base a la información de paridad.

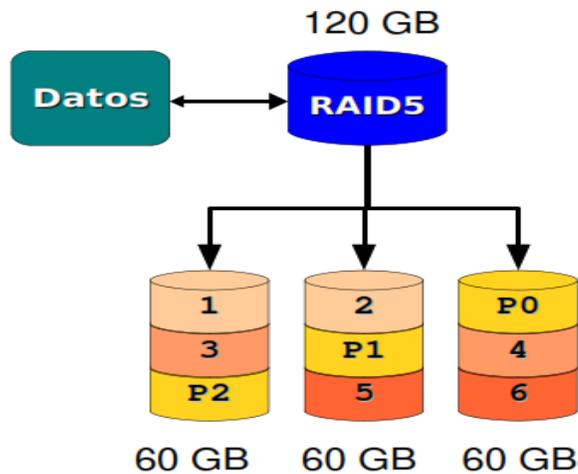


Figura 16 RAID 5 (120GB).

RAID 6 (block striping & distributed parity)

- Segmenta los datos a nivel de bloques distribuyéndolos entre los dispositivos del arreglo de forma similar a RAID 4.
- Distribuye los datos de paridad entre todos los dispositivos del arreglo de forma similar a RAID 5.
- Un segundo conjunto de datos de paridad lo que provee redundancia ante la falla de dos dispositivos.

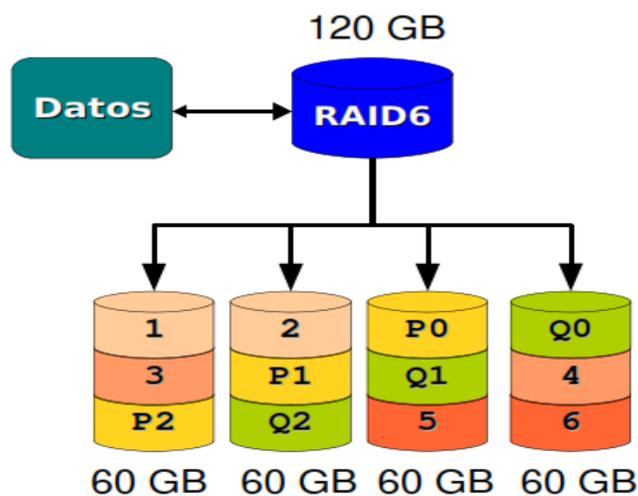


Figura 17 RAID 6 (120GB).

Es posible incrementar el rendimiento y la redundancia de un sistema de almacenamiento combinando diferentes niveles de RAID, estas combinaciones se conocen como arreglos híbridos. La mayoría de las tarjetas controladoras, gabinetes externos y software RAID soportan combinaciones de dos o más niveles. No todas las combinaciones soportadas y permitidas en hardware y software para RAID ofrecen beneficios.

RAID 10 (striping & mirror)

- A pesar de su alto costo es muy utilizados.
- Combina el rendimiento (velocidad de acceso) de la segmentación de datos (striping) con las propiedades de la redundancia (mirroring).
- Si un dispositivo falla ambos lados del RAID 10 seguirán funcionando (aunque un lado en modo degradado).

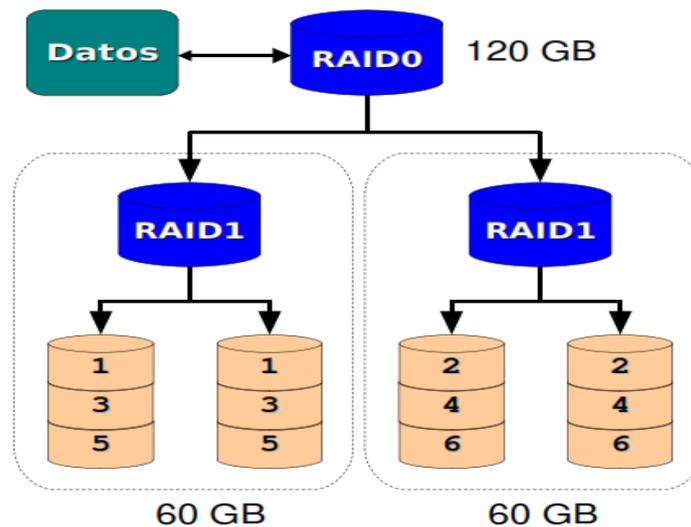


Figura 18 RAID 10 (120GB).

RAID 50 (striping & parity)

- En referencia a un RAID 10:
 - Menor costo.
 - Rendimiento de lectura más bajo (aunque sigue siendo bueno).
 - A nivel de escritura hay mayor rendimiento (velocidad).
 - Cada RAID 5 puede soportar la falla de un dispositivo.

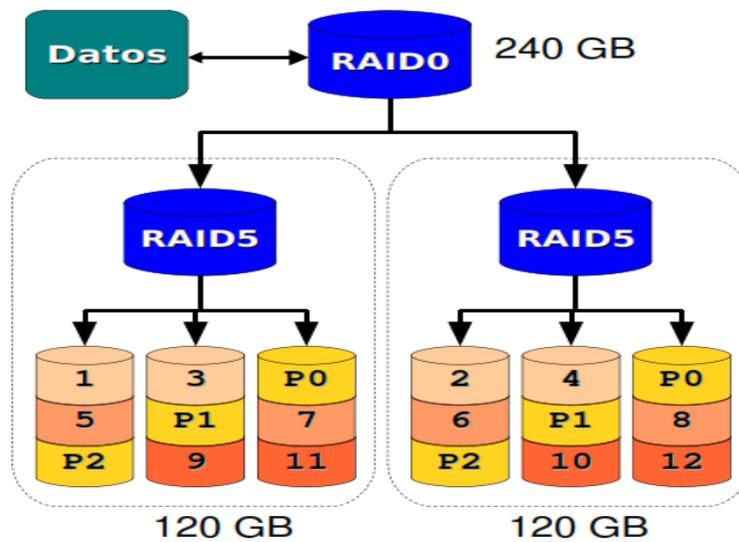


Figura 19 RAID 50 (240GB).

RAID 60 (striping & parity)

- En referencia a un RAID 10:
 - Menor costo.
 - Rendimiento de lectura más bajo (aunque es bueno).
 - A nivel de escritura hay mayor rendimiento (velocidad).
 - Cada RAID 6 puede soportar la falla de dos dispositivos (simultáneamente).

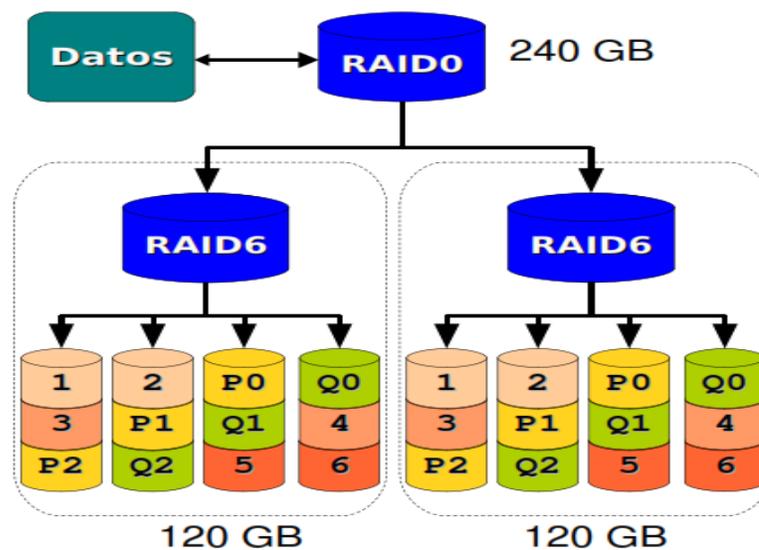


Figura 20 RAID 60 (240GB).

3.5.5.3 Tolerancia a fallas

Uno de los beneficios de RAID es la posibilidad de manejar fallas en los dispositivos sin detener el sistema de almacenamiento y sin intervención de un administrador (redundancia). Un RAID pre-configurado con los dispositivos necesarios, puede recuperarse de una falla por sí mismo.

En este sentido es importante revisar los siguientes conceptos:

- Modo de degradación.
- Hot spares.
- Hot swap.

Modo de degradación: El momento en que un grupo de dispositivos falla por cualquier razón en un RAID (con redundancia). Cuando ocurre:

- Funcionamiento no óptimo.
- Redundancia comprometida.
- Ocurre en RAID 1, 5, 10, 50 y 60.

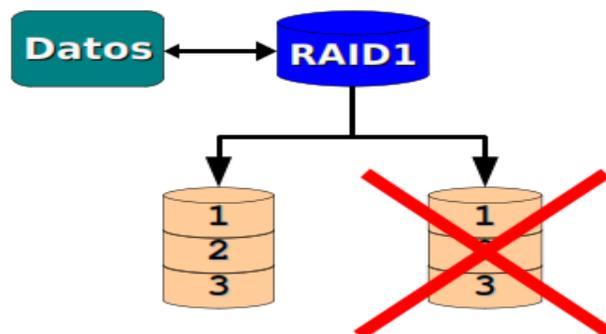


Figura 21 Modo de degradación.

Hot spares: Esta característica en niveles con soporte de redundancia permite que RAID se recupere de una falla por sí mismo.

- Soporte en hardware y software RAID.
- Dispositivos extras en stand by esperando un falla para ocupar el lugar del dispositivo dañado.
- El RAID pasa a modo degradado mientras hot spares actúa.

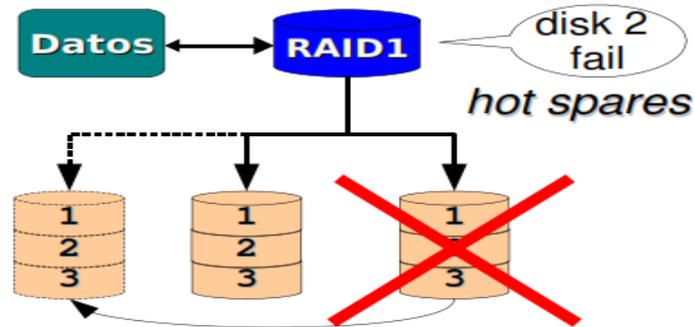


Figura 22 Host spares.

Hot swap: Estas características en niveles con soporte de redundancia permite que en el RAID se puedan retirar dispositivos que han fallado con el sistema operando (power on) Generalmente se usa en caso de:

- No se tiene espacio físico para instalar dispositivos con soporte hot spares.
- Un dispositivo en modo hot spares está activo y se debe reemplazar el dispositivo dañado previendo fallas a futuro.

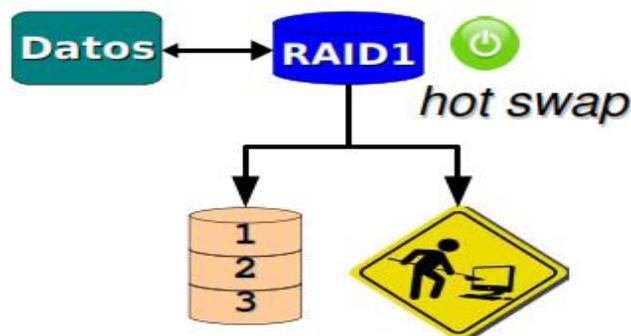


Figura 23 Hot swap.

3.5.5.4 Confiabilidad y Disponibilidad

Los sistemas RAID como cualquier otro pueden presentar fallas que lo desvíen de su funcionamiento normal.

Términos relacionados con fallas en el sistema:

- Failure rate.
- Mean time to data loss (MTTDL).
- Mean time to recovery (MTTR).

- Unrecoverable bit error rate (UBE)

Failure rate (Porcentaje de averías)

En términos generales se refiere a la frecuencia con la cual un sistema falla. En el contexto de RAID se aplican dos tipos de porcentajes de averías:

- Falla lógica: Si perdemos un solo dispositivo del RAID y su porcentaje está dado por la suma del porcentaje de falla de cada dispositivo del grupo.
- Falla del sistema: Si perdemos datos en el arreglo, este porcentaje depende del nivel de RAID implementado.

Mean time to data loss – MTDDL (Tiempo promedio antes de la pérdida de datos)

Tiempo promedio antes que la falla de uno o varios dispositivos pueda ocasionar pérdidas de datos en un arreglo. En conjunto con el tiempo promedio antes que ocurra una falla (mean time to failure - MTTF) son las dos métricas principales de confiabilidad en un arreglo.

- Si un arreglo no cuenta con redundancia $MTDDL = MTTF$.
- Si aumenta la redundancia en un arreglo MTDDL aumenta.
- $MTDDL \gg MTTF$ para un arreglo de alta confiabilidad.

Mean time to recovery – MTTR (Tiempo promedio de recuperación)

El tiempo que lleva recuperar un arreglo a su normal funcionamiento luego que ocurre una falla. Este tiempo incluiría:

- Tiempo en sustituir un dispositivo en falla.
- Tiempo para reconstruir el arreglo.
- En sistemas de alta disponibilidad el MTTR disminuye con el uso de arreglos (hardware o software RAID) que cuenten con soporte hot spares y/o hot swap.

Unrecoverable bit error rate – UBE (Tasa de error de bit irrecuperable)

Relacionado con el tiempo en el cual un dispositivo de un arreglo no tiene capacidad para recuperar los datos después de aplicar en varios intentos códigos de redundancia (ej. Códigos de Redundancia Cíclica – CRC)

En referencia a un RAID 5 o RAID 6 la UBE puede comprometer la reconstrucción de un arreglo (con redundancia) que ha entrado en un modo de degradación.

3.5.6 GlusterFS

Es un sistema de archivos de alta disponibilidad y escalabilidad que puede brindar almacenamiento a gran escala (peta bytes) a bajo costo (opensource) y manejo de hasta miles de clientes.

GlusterFS agrupa dispositivos de almacenamiento a través de la red y maneja la data como si fuese un solo bloque.

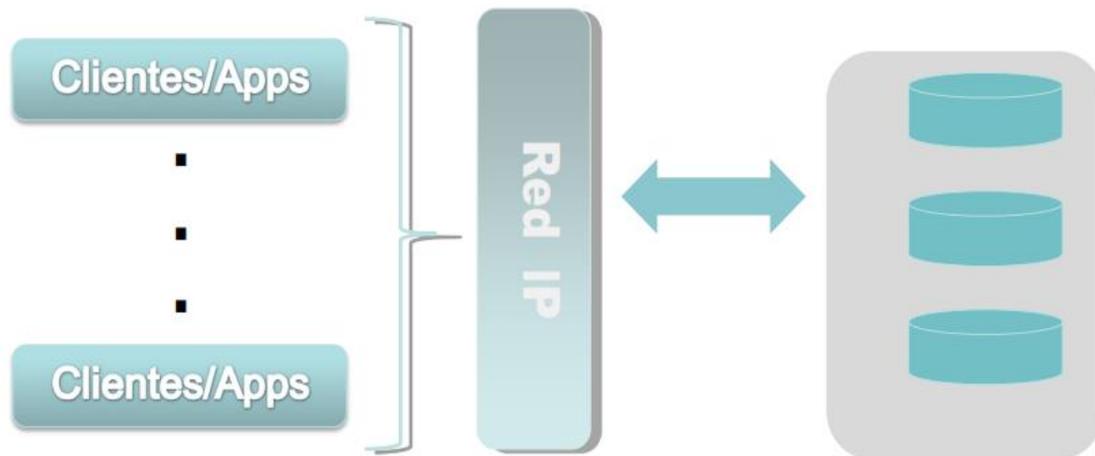


Figura 24 Ilustración de GlusterFS.

3.5.6.1 Ventajas de GlusterFS

- Flexibilidad para combinar diversos dispositivos, ya sean físicos virtuales o disco en la nube.
- Muy fácil de instalar y configurar.
- Diseño modular y apilable.
- Escalabilidad lineal.
- Completamente implementado en user-space (FUSE).
- Fácil de migrar, depurar y mantener.
- Diferentes modalidades de configuración.
- Se crea sobre un sistema de archivos existente (ext3, ext4) lo que hace que se puedan recuperar los archivos aun si Gluster deja de funcionar.

3.5.6.2 Método de acceso a la Data

Los volúmenes de Gluster se pueden acceder de diversas maneras, se puede utilizar NFS para exportar o también el protocolo nativo de Gluster que es el más

recomendado. Como fue mencionado anteriormente este protocolo está basado en FUSE (Filesystem user space) por lo cual hay que asegurarse que este módulo este cargado en el kernel. FUSE permite levantar el sistema de archivo de Gluster en el user space (espacio de memoria donde trabajan las aplicaciones del usuario).

Terminología

- Bloque (Brick): Es la capa física de almacenamiento para los volúmenes.
- Volumen: Un volumen está compuesto por múltiples bloques.
- Servidor: Equipo (virtual o real) que exporta el volumen. Es el que alberga el sistema de archivos donde se guardará la data, y provee acceso a los volúmenes.
- Cliente: Es el equipo que monta el volumen, este equipo también puede ser un servidor.

3.5.6.3 Distribución y Replicación de la Data

Volumen Distribuido: La configuración de Gluster en modo distribuido toma una lista de sub-volúmenes y distribuye los archivos entre ellos. Para calcular en cuál de los servidores será almacenado el archivo se realiza un hash al nombre del archivo.

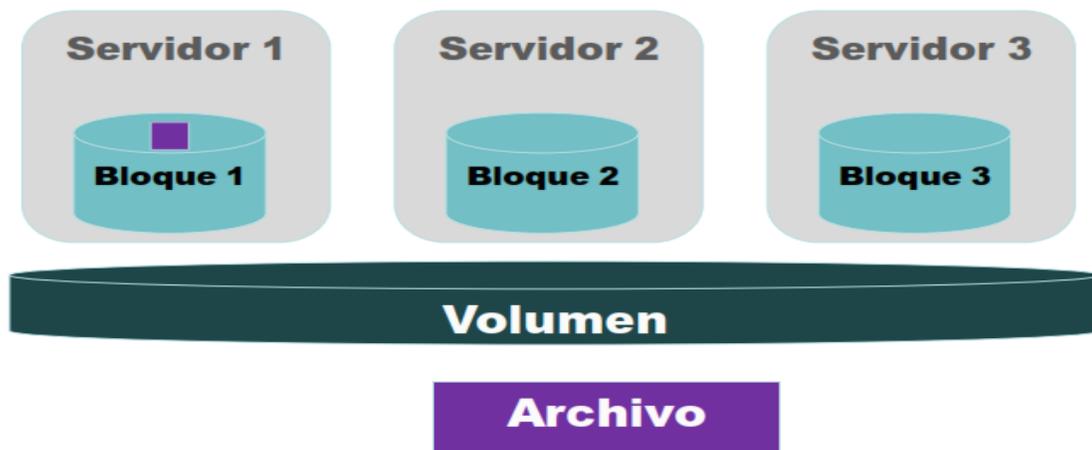


Figura 25 Distribución y Replicación de la data con: Volumen distribuido.

Este modo tiene ciertas desventajas como que si uno de los servidores esta fuera de servicio no se pueden recuperar los archivos que están almacenados en él. También pueden presentarse problemas de lectura/escritura si un archivo es de un tamaño mayor al sub-volumen en el que debe ser almacenado.

Volumen en Réplica: Es el modo más utilizado, este modo provee redundancia en la información almacenada en los servidores, lo que se encuentra en un servidor esta exactamente igual en los otros servidores, lo que ofrece no solo redundancia en la data sino en la disponibilidad del servicio. En cada operación sobre los archivos se hace la copia en la réplica por lo que podemos decir que hay sincronización en la réplica. Funciona de manera análoga a RAID1.

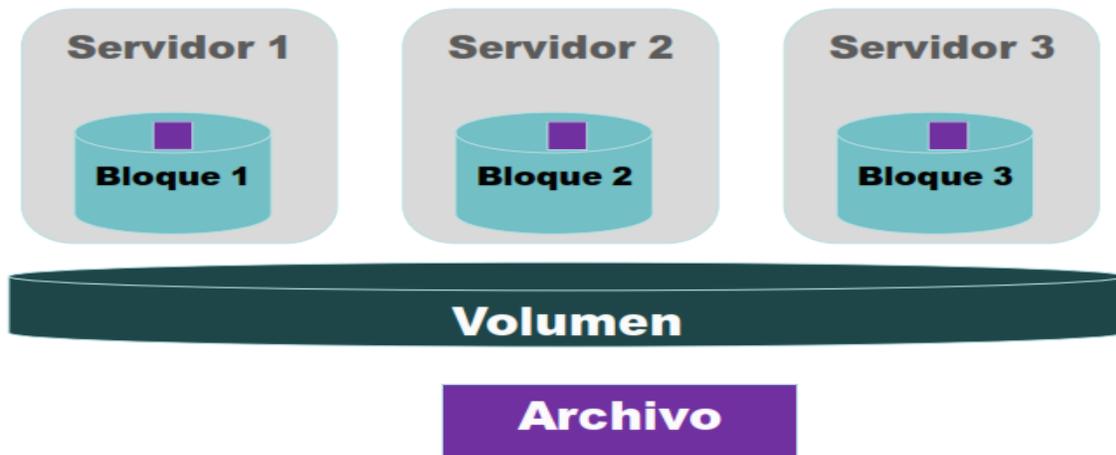


Figura 26 Distribución y Replicación de la data con: Volumen en réplica.

Volumen Distribuido en Réplica: Esta modalidad es una mezcla de las dos anteriores y distribuir los archivos a través de volúmenes en réplica. Este tipo de configuración ofrece mejoras considerables en operaciones de lectura en la mayoría de los ambientes. Es el recomendado cuando la alta disponibilidad y la alta confiabilidad es crítica.



Figura 27 Distribución y Replicación de la data con: Distribuido en Réplica.

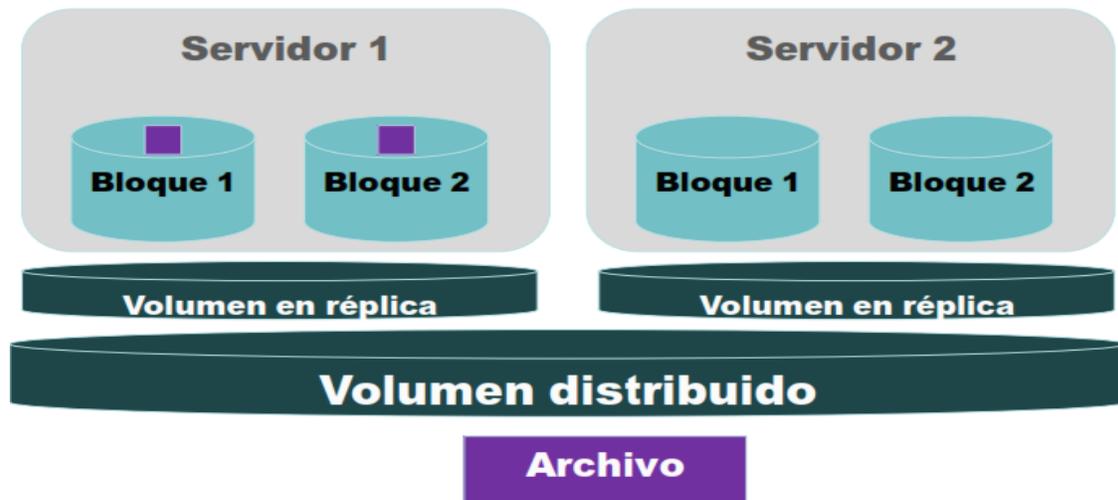


Figura 28 Distribución y Replicación de la data con: Réplica Distribuida.

Geo-Replicación: Esta modalidad provee de un servicio de replicación asíncrona a través de una red LAN o WAN o Internet. Trabaja basándose en un modelo maestro-esclavo, siendo el maestro un volumen GlusterFS, y el esclavo puede ser

- Un directorio Local.
- Otro volumen GlusterFS, que puede ser local o encontrarse en un host diferente.

Cuando la data en el maestro deja de estar disponible, puede restituirse a partir de cualquiera de los esclavos. La Geo-Replicación a diferencia del Volumen en Réplica mantiene la copia de la data de forma asíncrona, es decir chequea periódicamente cambios en la información sincroniza al detectar algún cambio.

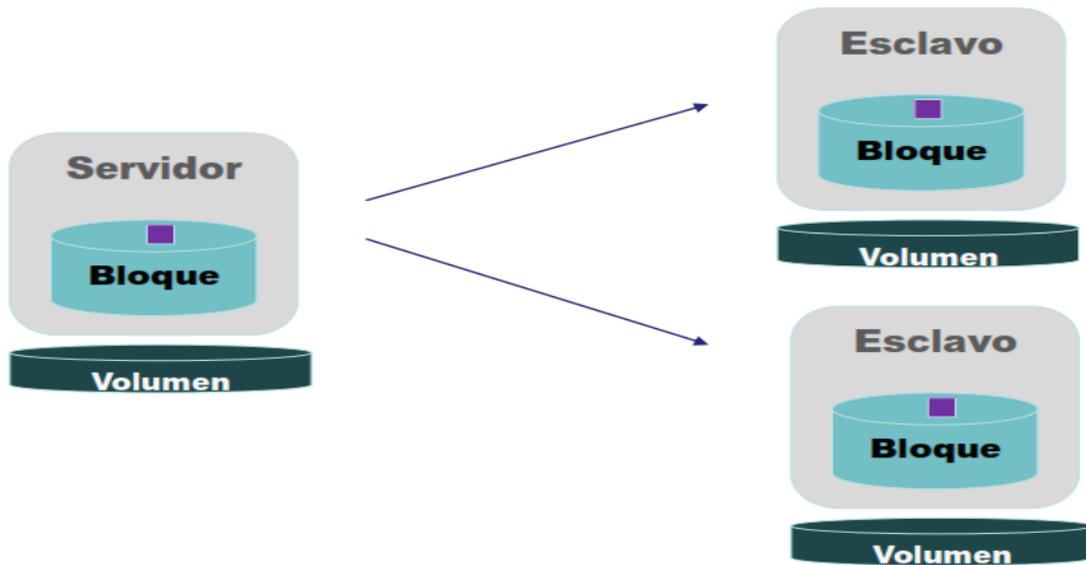


Figura 29 Distribución y Replicación de la data con: Geo-Replicación.

Volumen Stripped: Esta modalidad es generalmente utilizada para almacenar datos de computación de alto rendimiento, funciona repartiendo cada archivo en diferentes bloques, con lo que podemos inferir que es más óptimo para archivos de gran tamaño. Funciona de manera análoga a un RAID 0.

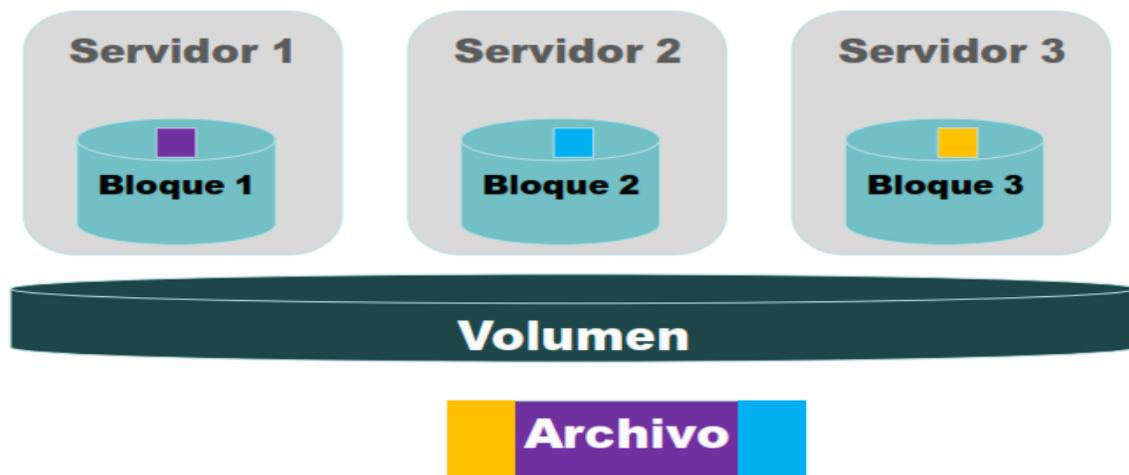


Figura 30 Distribución y Replicación de la data con: Volumen Stripped.

3.5.6.4 Componentes de Gluster

Glusterd: Demonio que gestiona el volumen. Debe correr en todos los servidores que exportan un sistema de archivo.

Gluster: Es el comando para abrir la consola de Gluster. En la consola de Gluster también se pueden configurar manualmente los volúmenes (aumentar, reducir, balancear, migrar...) sin tener que parar el servicio.

3.5.7 Heartbeat

Es un servicio (Demonio), que ofrece alta disponibilidad a determinados recursos mediante la creación y mantenimiento de un clúster completo por una serie de nodos. Permite a un conjunto de máquinas monitorearse entre sí y de esta forma ganar un nivel extra de fiabilidad de un servicio.

El software heartbeat trabaja enviando latidos (ping), los cuales verifican si el Server principal está activo o no, estos pings enviados por heartbeat requieren una respuesta por parte del Server principal o master, si al cabo de un cierto tiempo el Server no responde dichos ping, heartbeat determina que ese Server se encuentra inactivo/caído, y automáticamente activa al Server secundario para que asuma el control de la red.

Heartbeat es un demonio que proporciona la infraestructura de clúster de servicios a sus clientes. Esto permite a los clientes saber sobre la presencia (o no) De los procesos de pares en otras máquinas e intercambiar fácilmente mensajes con ellos.

Para realizar su trabajo, Heartbeat emite un latido el cual determina la presencia de los recursos requeridos en las otras máquinas, el mismo debe que ser combinado con un administrador de recursos de clúster (CRM), el cual la tarea de iniciar y detener los servicios (direcciones IP, servidores web, etc) que se agrupan, para así poder ofrecer la altamente disponible.

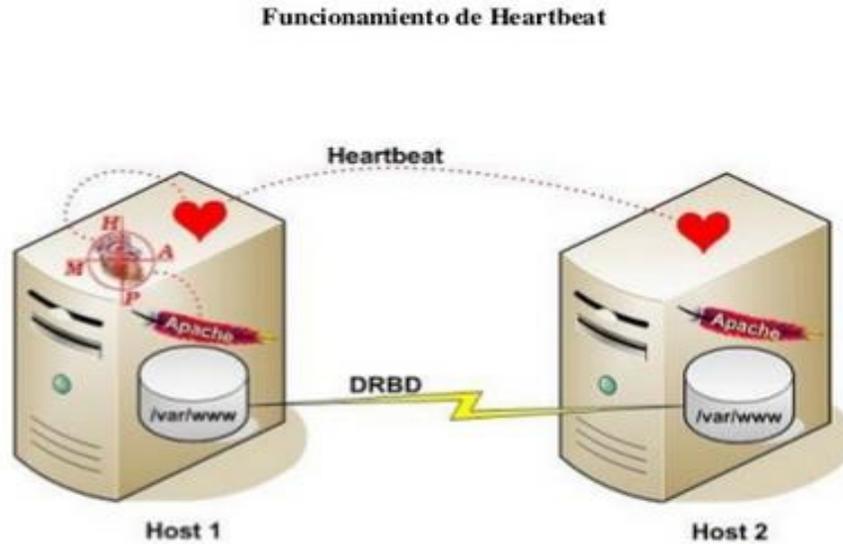


Figura 31 Funcionamiento de Heartbeat.

3.5.7.1 Proyectos de alta disponibilidad

Para GNU/Linux existen una gran variedad de proyectos que nos aportan las características de la alta disponibilidad, proyectos tanto para la alta disponibilidad en servicios como en datos.

Proyecto destinado a ofrecer alta disponibilidad en servicio:

- The High Availability Linux Project: Proporciona una solución de alta disponibilidad ofreciendo fiabilidad, disponibilidad.

Proyecto destinado a ofrecer alta disponibilidad en datos:

- DRBD: Software de replicación de dispositivos de bloque formando un RAID 1 a través de la red, es decir, replica los datos en distintas localizaciones. Datos de un sistema de archivos, de una base de datos, etc. Algunas de sus características son una replicación en tiempo real de manera continua, transparencia en las aplicaciones que estén almacenando datos en la unidad, posibilidad de recuperación de los datos ante un desastre, etc. Se complementa perfectamente con Heartbeat.

3.5.7.2 Clúster

Son un tipo de arquitectura paralela distribuida, que consiste en un conjunto de computadoras independientes interconectadas entre sí, operando de forma conjunta como un único recurso computacional.

Los clúster son usualmente empleados para mejorar el rendimiento y/o la disponibilidad por encima de la que es provista por un solo computador típicamente, siendo más económico que computadores individuales de rapidez y disponibilidad comparables.

De un clúster se espera que presente combinaciones de los siguientes servicios:

- Alto Rendimiento.
- Alta Disponibilidad.
- Balanceo de carga.
- Escalabilidad.

Clasificación de los Clúster:

Alto rendimiento: Son clústeres en los cuales se ejecutan tareas que se 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 clúster por largos periodos de tiempo.

Alta disponibilidad: Son clúster cuyo objetivo diseño es proveer disponibilidad y confiabilidad. Estos clúster 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.

Alta eficiencia: son clúster cuyo objetivo de diseño es la de 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 clúster no es considerado un gran problema.

3.5.7.3 Clúster de alta disponibilidad

Es conjunto de dos o más máquinas que se caracterizan por mantener una serie de servicios compartidos y por estar constantemente monitorizándose entre sí. Podemos dividirlos en dos clases:

Alta disponibilidad de infraestructura: Si se produce un fallo de hardware en una de las máquinas del clúster, el software de alta disponibilidad es capaz de arrancar automáticamente los servicios en cualquiera de las otras máquinas del clúster (failover). Y cuando la máquina que ha fallado se recupera, los servicios son nuevamente migrados a la máquina original (failback). Esta capacidad de recuperación automática de servicios nos garantiza la alta disponibilidad de los servicios ofrecidos por el clúster, minimizando así la percepción de fallo por parte de los usuarios.

Alta disponibilidad de aplicación: Si se produce un fallo de hardware o de las aplicaciones de alguna de las máquinas del clúster, el software de alta disponibilidad es capaz de arrancar automáticamente los servicios que han fallado en cualquiera de las otras máquinas del clúster. Y cuando la máquina que ha fallado se recupera, los servicios son nuevamente migrados a la máquina original. Esta capacidad de recuperación automática de servicios nos garantiza la integridad de la información, ya que no hay pérdidas de datos, y además evita molestia a los usuarios, que no tiene por qué notar que se ha producido un problema.

No hay que confundir un clúster de alta disponibilidad con un clúster de alto rendimiento. El segundo es una configuración de equipos diseñados para proporcionar capacidades de cálculo muchos mayores que la proporcionan los equipos individuales, mientras el primer tipo de clúster está diseñado para garantizar el funcionamiento ininterrumpido de ciertas aplicaciones.

3.5.8 LVS (Linux Virtual Server)

Un clúster LVS es un grupo de servidores reales que de forma transparente para el usuario actúa como si fuera un único y gran servidor. A este servidor homogéneo se le llama "Virtual Server". Los servidores reales son gestionados por un Director (también denominado load balancer o balanceador de carga). La parte esencial del director es ipvs. IPVS (IP Virtual Server) implementa una capa de transporte dentro del kernel de Linux que distribuye las peticiones de los clientes entre los servidores reales (balance de carga).

Balanceador de carga (Load balancer): es la interfaz de usuario para el servicio, tal y como es visto por el mundo exterior. El Balanceador de carga dirige las conexiones de red de los clientes que conocen una única dirección IP para los servicios, a un conjunto de servidores que son los que realmente realizan el trabajo.

Piscina de servidores (Server Pools): consiste en un clúster de servidores que implementan los servicios tales como web, ftp, correo, DNS, etc.

Almacenamiento (Backend storage), proporciona el almacenamiento compartido para los servidores, por lo que es fácil para los servidores para mantener el mismo contenido y proporcionar los mismos servicios.

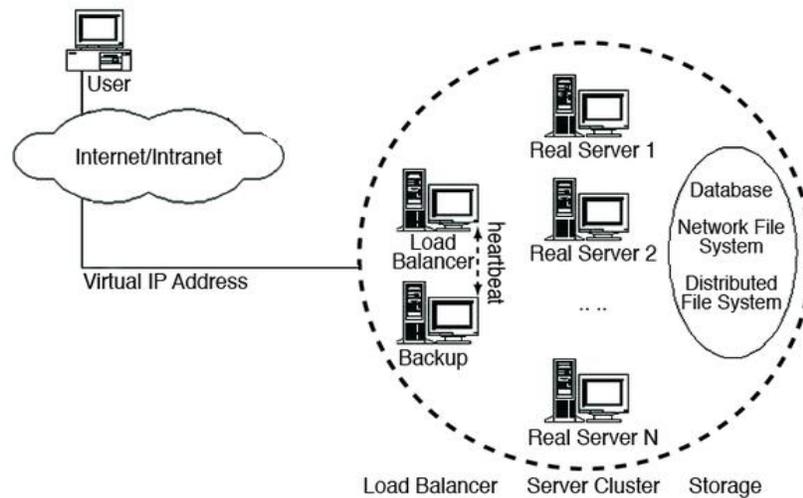


Figura 32 Funcionamiento de LVS.

3.5.8.1 Técnicas de balanceador de carga con LVS

LVS-NAT

- El balanceador de carga y los servidores reales están interconectados en el mismo segmento de red.
- Cuando llega una petición al balanceador de carga el mismo, la reenvía al servidor real (para esta analiza la ip y puerto destino), dependiendo del algoritmo de balanceo de carga.
- La dirección IP y el puerto destino son reescritos y reenviado a uno de los servidores reales. Cuando el paquete pertenece a una conexión establecida, el servidor busca en la tabla hash para reescribir el destino.
- Cuando el paquete es respondido por el servidor real el mismo se lo entrega al balanceador de carga y se reescribe la dirección IP y puerto origen.
- Al terminarse la conexión se borra la entrada de la tabla hash del balanceador de carga.

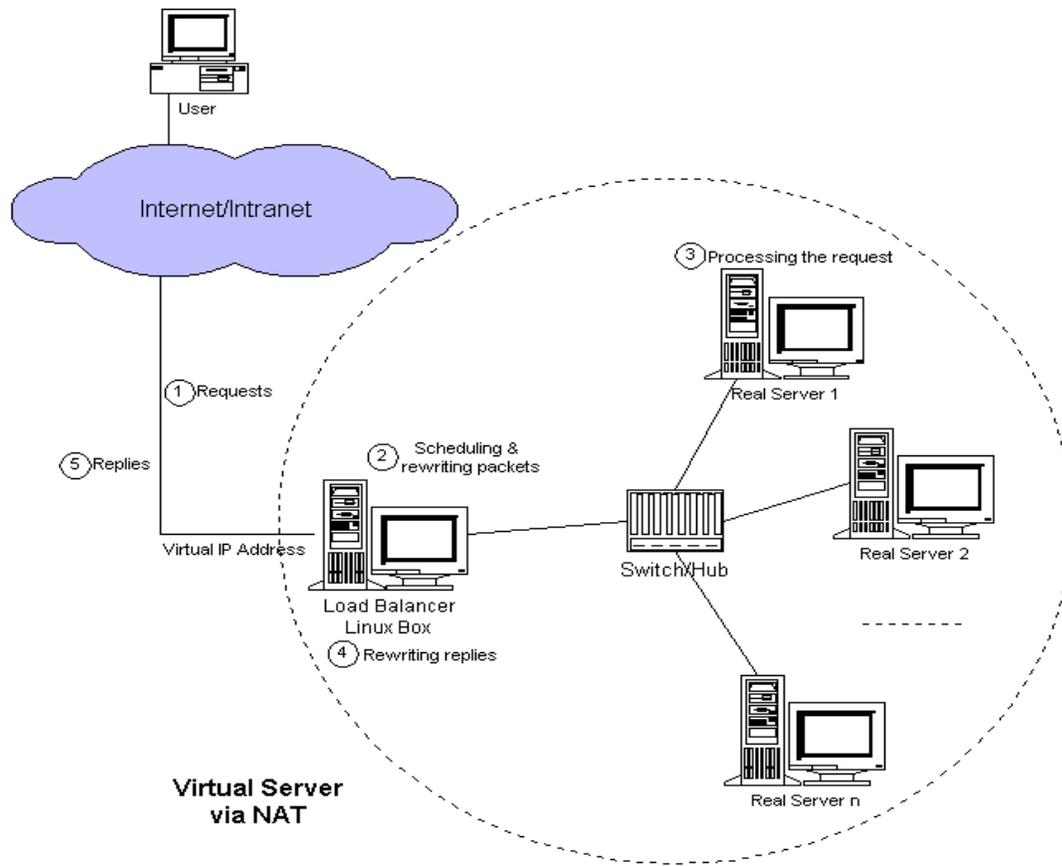


Figura 33 Estructura del modo LVS-NAT.

LVS-TUN

- El balanceador de carga y los servidores reales están interconectados sin importar su ubicación lógica ni física red.
- Cuando llega una petición al balanceador de carga el mismo, la encapsula y reenvía al servidor real (para esta analiza la ip y puerto destino), dependiendo del algoritmo de balanceo de carga.
- Cuando el paquete pertenece a una conexión establecida, el servidor busca en la tabla hash para reescribir el destino.
- Cuando el paquete es recibido por el servidor real el mismo desencapsula la petición y responde la petición reescribiendo la dirección IP y puerto origen.
- Al terminarse la conexión se borra la entrada de la tabla hash del balanceador de carga.

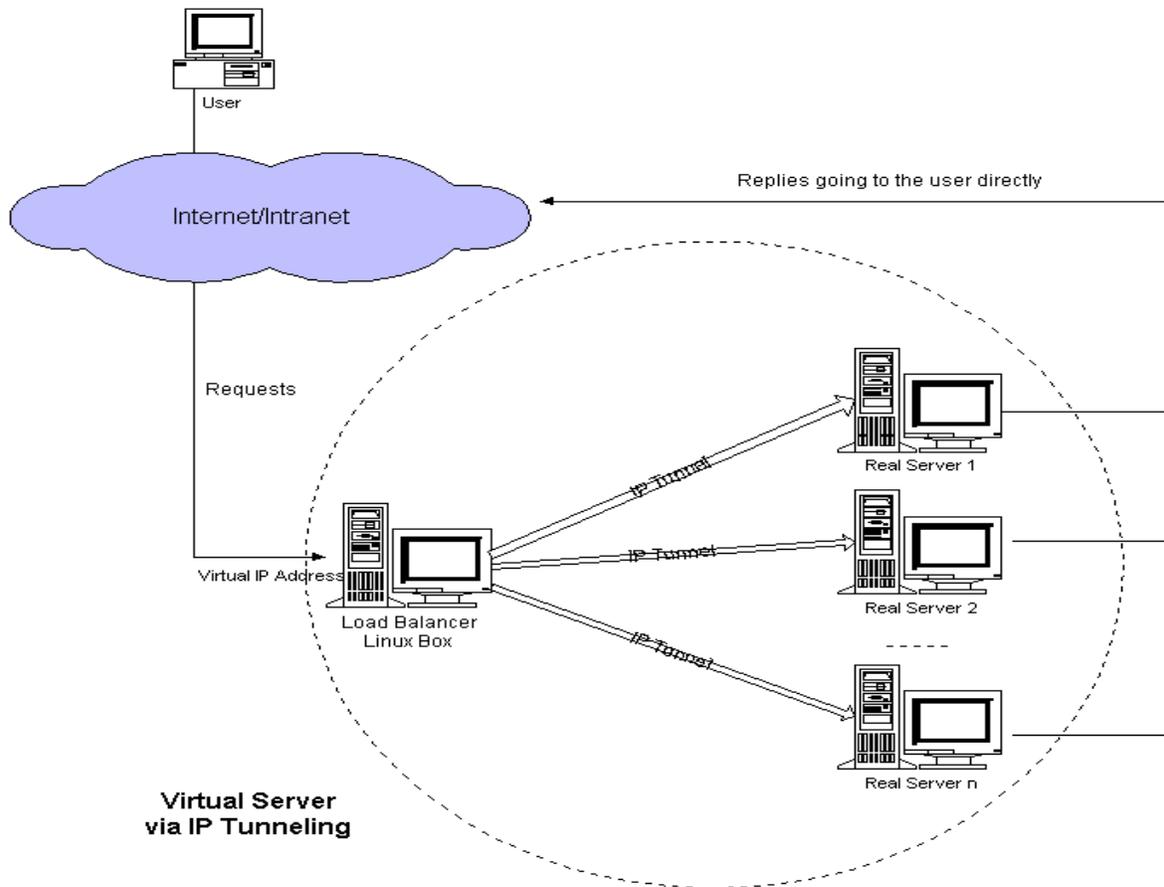


Figura 34 Estructura del modo LVS-TUN.

LVS-DR

- El balanceador de carga y los servidores reales están interconectados en el mismo segmento de red.
- La ip real es compartida por el balanceador de carga y los servidores reales, y en los servidores reales la misma se encuentra configurada en la interfaz loopback.
- Cuando llega una petición al balanceador de carga el mismo, la reenvía al servidor real (para esta solo se cambia la dirección Mac de destino), dependiendo del algoritmo de balanceo de carga.
- Cuando el paquete pertenece a una conexión establecida, el servidor busca en la tabla hash para reescribir la dirección Mac del destino.
- Cuando el paquete es recibido por el servidor real el mismo responde la petición.
- Al terminarse la conexión se borra la entrada de la tabla hash del balanceador de carga.
- Los servidores reales no deben responder a peticiones MAC (ARP response).

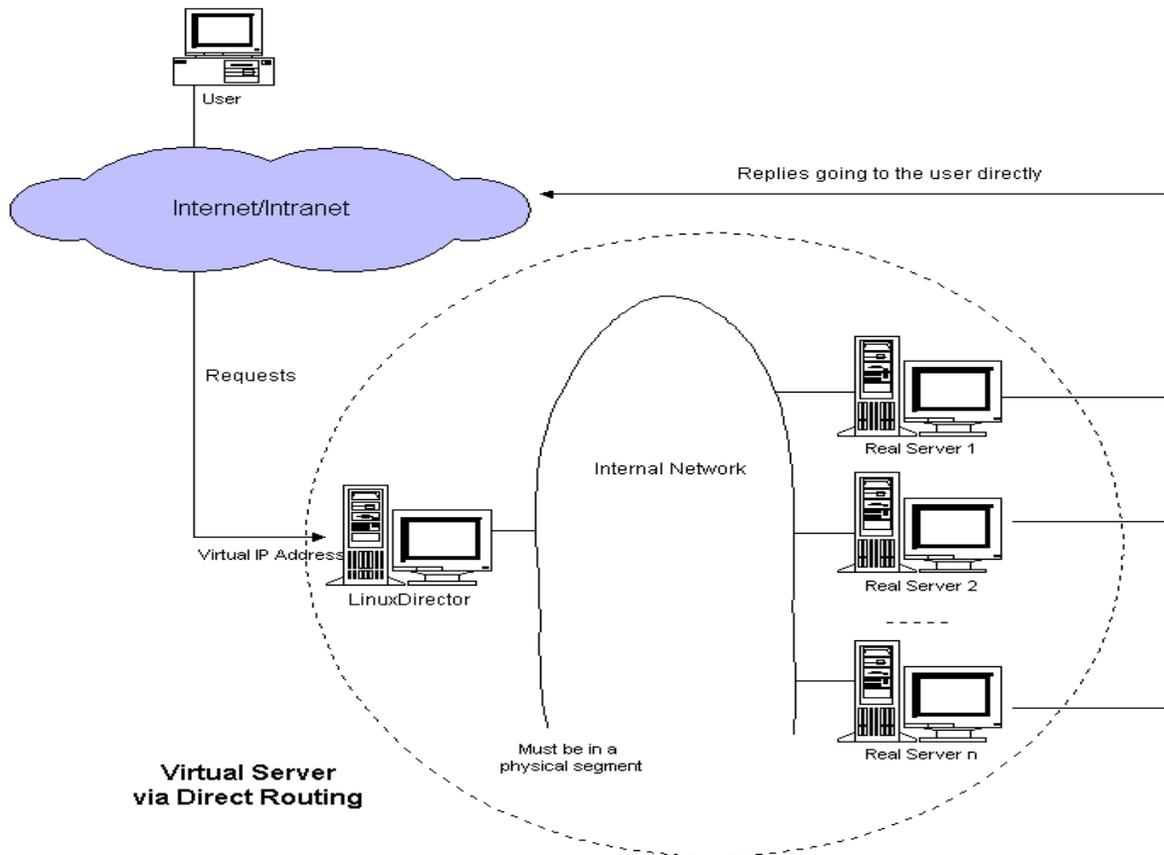


Figura 35 Estructura del modo LVS-DR.

3.5.8.2 Técnicas de programación de conexiones

Las Técnicas de programación de conexiones (Connection Scheduling) usadas por lvs son:

- Round-Robin.
- Weighted Round-Robin.
- LeastConnection.
- Weighted Least-Connection.

El algoritmo de planificación Round-Robin dirige las conexiones de red a los diferentes servidores tratando a todos los servidores reales de igual forma independientemente del número de conexiones o el tiempo de respuesta.

El algoritmo de planificación Weighted Round-Robin dirige las conexiones de red a los diferentes servidores tratando a todos los servidores reales según sus capacidades.

El algoritmo de planificación LeastConnection dirige las conexiones de red a los servidores con menos conexiones.

El algoritmo de planificación Weighted LeastConnection dirige las conexiones de red a los servidores con un cálculo entre sus capacidades y su número de conexiones.

4 Prácticas de Laboratorio

Introducción

La asignatura Electiva VIII: Computación en la Nube pertenece al plan académico 2011 de la carrera de Ingeniería en Telemática del Departamento de Computación de la UNAN-León y se imparte en el I semestre del V año de la carrera, 2 horas semanales de teoría y 2 horas semanales de laboratorio. El contenido de este documento podrá ser revisado (de forma no significativa) para adaptar los trabajos y tareas del laboratorio a la evolución de la asignatura en su conjunto.

Programación

Práctica 1: Redes de Ordenadores en VMWare.

Práctica 2: Virtualización por contenedores OpenVZ.

Práctica 3: Web Panel OpenVZ.

Práctica 4: RAID.

Práctica 5: GlusterFS.

Práctica 6: Heartbeat.

Práctica 7: LVS.

Práctica 8: Creación de una pequeña nube.

Evaluación

Como parte de un proceso de evaluación continua, y al ser un componente teórico práctico el puntaje para cada evaluación parcial se dividirá de la siguiente manera: 50% para los elementos teóricos y 50% para los elementos prácticos.

Práctica 1: Redes de Ordenadores en VMWare.

Introducción:

Esta práctica será desarrollada en el entorno de virtualización de VMWare. Se deberá crear 3 máquinas virtuales e interconectarlas.

Objetivos:

- Conocer el entorno de virtualización y el proceso de creación de máquinas virtuales.
- Configurar redes de computadores mediante el escenario de máquinas virtuales.

Requerimientos:

Hardware	Software
Computadora con los siguientes requisitos: <ul style="list-style-type: none"> • Procesador mínimo de velocidad de 2.1 GHz • Memoria RAM de 2 GB. 	Software de virtualización VMware con siguientes elementos: <ul style="list-style-type: none"> • Router (CentOS 6.6-i386 server) • PC1 Cliente (lubuntu12.04) • PC2 Cliente (lubuntu12.04 o Windows XP)

Duración de la práctica

- Una sesión de laboratorio.

Topología

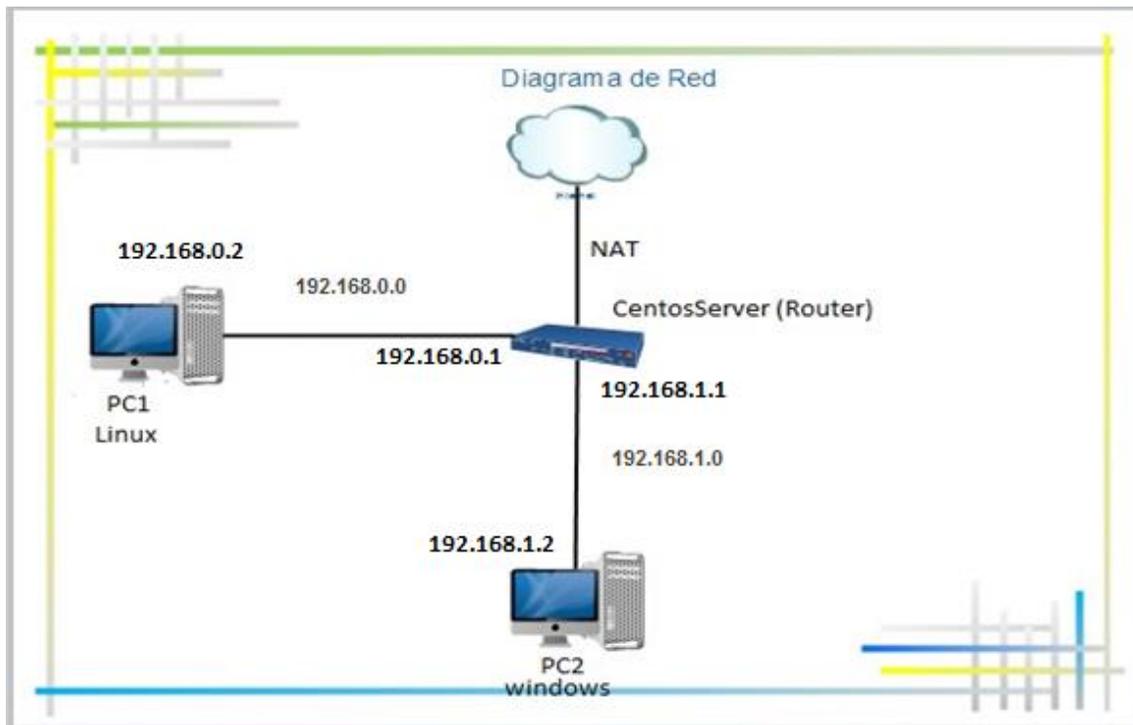


Figura 36 Topología a seguir para la práctica numero 1

Comandos de ayuda

Comando	Descripción
# ping "Dirección ip"	Comprueba el estado de la comunicación.
# vi /etc/network/interfaces	Contiene la configuración de interfaz de red.
# ifup "Nombre de la interfaz"	Es para levantar la interfaz.
# ifdown "Nombre de la interfaz"	Es para bajar la interfaz.

Enunciado:

Para la realización de esta práctica se deberán realizar los siguientes pasos:

1. Agregar al CentOS dos nuevos adaptadores de red. Éste contará en total con tres adaptadores 1 en modo NAT y dos en red interna para comunicar con los clientes ya que funcionará como Router.
2. Configurar en CentOS los nuevos adaptadores de manera que sean reconocidos por el S.O.
3. Con virtual network editor de VMware crear dos segmentos de red y asignar a cada subred las IP's de red descritas en el diagrama.

4. Cambiar el tipo de conexión de los adaptadores de red de las PC y server, estos deberán tomar una IP según el diagrama. Ej. PC1 -> IP: 192.168.0.2
5. Probar la comunicación entre:

PC1 -> Server

```
# ping 192.168.0.1
```

Server -> PC2

```
# ping 192.168.1.2
```

6. Configurar el reenvío de paquetes en el CentOS para que pueda realizar enrutamiento a ambas subredes.

```
# vi /etc/sysctl.conf
```

Y deberá quedarles como lo siguiente:

```
# Kernel sysctl configuration file for Red Hat Linux
#
# For binary values, 0 is disabled, 1 is enabled. See sysctl(8) and
# sysctl.conf(5) for more details.

# Controls IP packet forwarding
net.ipv4.ip_forward = 1

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Do not accept source routing
net.ipv4.conf.default.accept_source_route = 0

# Controls the System Request debugging functionality of the kernel
kernel.sysrq = 0

# Controls whether core dumps will append the PID to the core filename.
# Useful for debugging multi-threaded applications.
kernel.core_uses_pid = 1

# Controls the use of TCP syncookies
net.ipv4.tcp_syncookies = 1
```

7. Asignar rutas estáticas en PC1 y PC2 en el siguiente ruta:

```
# vi /etc/network/interfaces
```

8. Probar la conexión enviando un ping desde la PC1 hacia la PC2:

```
# ping 192.168.1.2
```

Y debe aparecer lo siguiente:

```
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
```

```
64 bytes from 192.168.1.2: icmp_req=1 ttl=63 time=2.79 ms
64 bytes from 192.168.1.2: icmp_req=2 ttl=63 time=1.79 ms
64 bytes from 192.168.1.2: icmp_req=3 ttl=63 time=1.77 ms
64 bytes from 192.168.1.2: icmp_req=4 ttl=63 time=1.77 ms
```

9. Hacer un clonado de una de las máquinas (probar su funcionamiento).
10. Hacer el Snapshot en una de las máquinas (probar su funcionamiento).

Preguntas de control

1. ¿Cuál es la importancia de la virtualización?
2. ¿Qué son los discos virtuales?
3. ¿Qué es NAT?
4. Explique el funcionamiento de reenvío de paquete.

Práctica 2: Virtualización por contenedores OpenVZ.

Introducción:

OpenVZ es una tecnología de virtualización en el nivel de sistema operativo para Linux. OpenVZ permite que un servidor físico ejecute múltiples instancias de sistemas operativos aislados, conocidos como Servidores Privados Virtuales (SPV o VPS en inglés). **OpenVZ** es un software que nos permite virtualizar sistemas operativos en entornos aislados. En este laboratorio configuraremos a nivel básico un contenedor (OS *virtualization*) basado en OpenVZ. Durante el desarrollo de las actividades de laboratorio tendrá los privilegios de administración del usuario **root** de cada servidor.

Objetivos:

- Conocer el entorno OpenVZ (instalación y configuración).
- Conocer el entorno de virtualización implementado contenedores.

Requerimientos:

Hardware	Software
Computadora con los siguientes requisitos: <ul style="list-style-type: none"> • Procesador mínimo de velocidad de 2.1 GHz • Memoria RAM de 4 GB. 	Software de virtualización VMware con siguientes elementos: <ul style="list-style-type: none"> • Router (CentOS 6.6-i386 server) • PC1 Servidor OpenVZ (CentOS 6.6-i386) • PC2 Cliente (lubuntu12.04)

Duración de la práctica

- Dos sesiones de laboratorio.

Topología

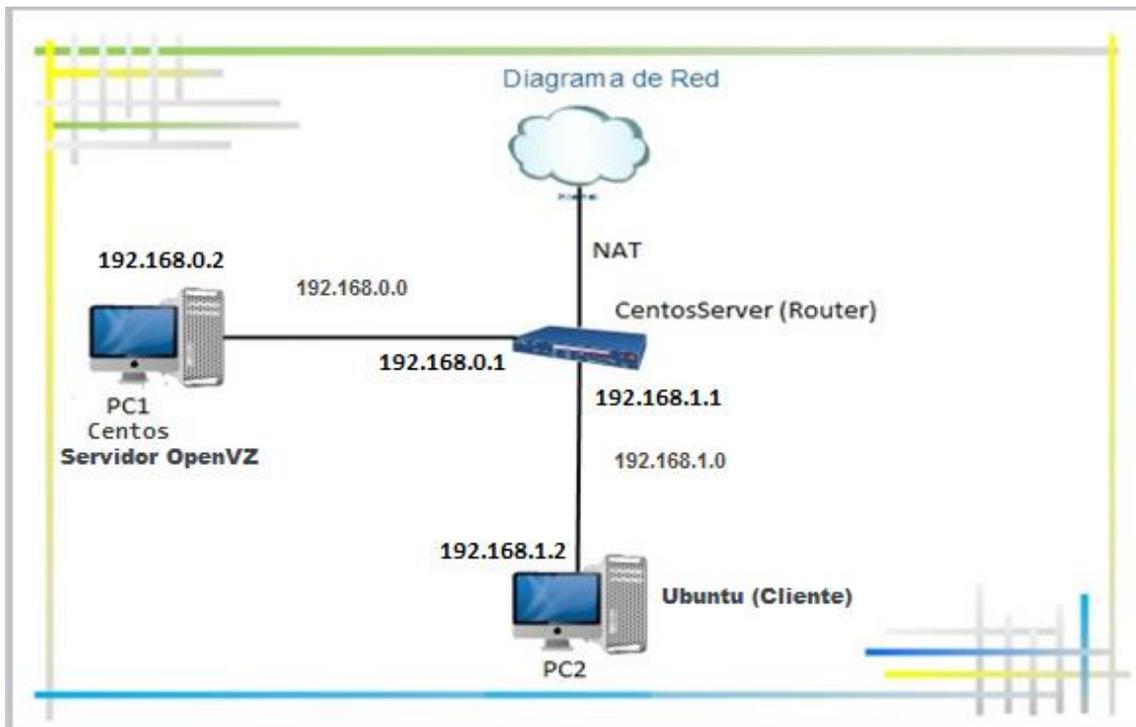


Figura 37 Topología a seguir para la práctica numero 2

Comandos de ayuda

Comando	Descripción
# yum install vzkernel	Instala OpenVZ.
# yum install openssh-server openssh-clients	Instala paquete ssh.
# Reboot	Reinicia el PC.
# yum update	Actualización de fuentes de software.
# vzctl create id --ostemplate name-plantilla	Crear un contenedor con identificador "id" a partir de una plantilla.
# vzctl set id --ipadd 192.168.0.101 --save	Añadir dirección estática al contenedor.
# vzctl set id --ipdel 192.168.0.101 --save	Eliminar la dirección estática al contenedor.
# vzctl enter id	Acceder o entrar al contenedor.
# vzctl start id	Encender el contenedor.

<pre># vzctl set 101 --onboot no --save</pre>	El contenedor arranca cuando el equipo físico arranca.
<pre># vzctl set id --disable no --save</pre>	Activar para que el contenedor no encienda al iniciar el servidor OpenVZ

Enunciado:

Para la realización de esta práctica se deberán realizar los siguientes pasos:

1. Actualizar fuentes de software.

```
# yum update
```

2. Añadir OpenVZ repositorios:

```
cd /etc/yum.repos.d
wget http://download.openvz.org/openvz.repo
rpm --import http://download.openvz.org/RPM-GPG-Key-OpenVZ
```

3. Abrir el archivo `openvz.repo` y deshabilitar el repositorio `[openvz-kernel-rhel5]` (`enabled=0`) y habilitar el repositorio `[openvz-kernel-rhel6]` (`enabled=1`) y nos quedara como lo siguiente:

```
[...]
[openvz-kernel-rhel5]
name=OpenVZ RHEL5-based kernel
#baseurl=http://download.openvz.org/kernel/branches/rhel5-2.6.18/current/
mirrorlist=http://download.openvz.org/kernel/mirrors-rhel5-2.6.18
enabled=0
gpgcheck=1
gpgkey=http://download.openvz.org/RPM-GPG-Key-OpenVZ
[...]
[openvz-kernel-rhel6]
name=OpenVZ RHEL6-based kernel
#baseurl=http://download.openvz.org/kernel/branches/rhel6-2.6.32/current/
mirrorlist=http://download.openvz.org/kernel/mirrors-rhel6-2.6.32
enabled=1
```

```
gpgcheck=1
gpgkey=http://download.openvz.org/RPM-GPG-Key-OpenVZ
[...]
```

4. Instalar OpenVZ.
5. Editamos el archivo `/etc/sysctl.conf` para poder enrutar (el reenvío de paquetes ip). añadir los siguientes parámetros del kernel antes de iniciar sesión en el núcleo vz:

```
net.ipv4.ip_forward = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
kernel.sysrq = 1
net.ipv4.tcp_syncookies = 1
net.ipv4.conf.default.proxy_arp = 0
net.ipv4.conf.default.send_redirects = 1
net.ipv4.conf.all.send_redirects = 0
net.ipv4.icmp_echo_ignore_broadcasts = 1
net.ipv4.conf.default.forwarding = 1
net.ipv4.conf.all.forwarding = 1
```

6. Comprobamos las opciones de `/etc/sysctl.conf`:

```
# sysctl -p
```

7. Abrir el archivo `/etc/vz/vz.conf` y deberá quedar como lo siguiente:

```
[...]
NEIGHBOUR_DEVS=all
[...]
```

8. SELinux tiene que ser desactivado si desea utilizar OpenVZ. Abrir el archivo `/etc/sysconfig/selinux` y establecer el valor de SELINUX a deshabilitado:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
```

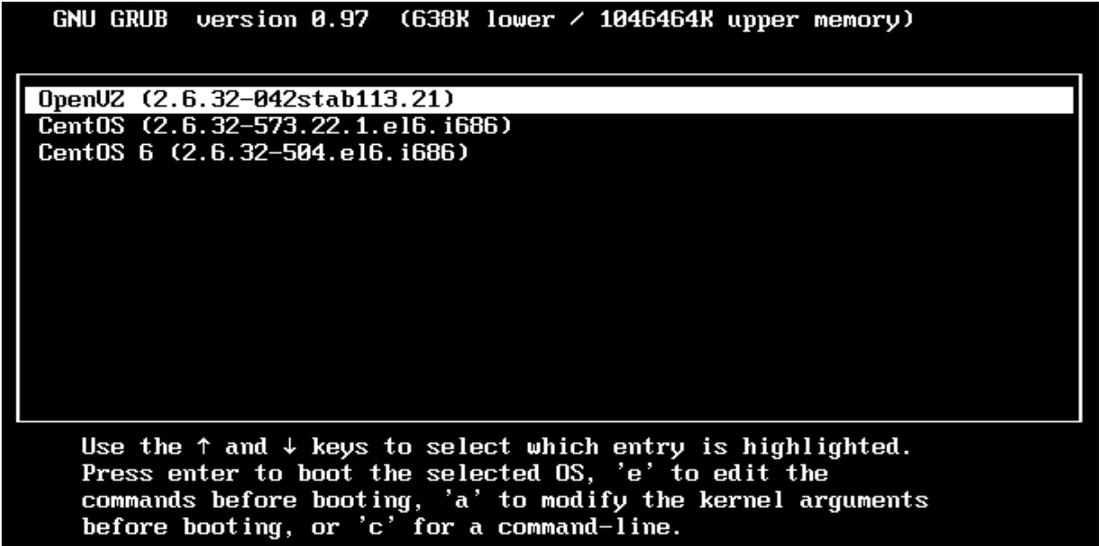
```
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
#   targeted - Targeted processes are protected,
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

9. Instalar las siguientes herramientas para recopilar las estadísticas de uso OpenVZ:

```
# yum install vzctl vzquota ploop vzstats
```

10. Hemos instalado con éxito OpenVZ.

11. Reiniciar la PC e iniciar sesión en kernel de OpenVZ



```
GNU GRUB version 0.97 (638K lower / 1046464K upper memory)
OpenVZ (2.6.32-042stab113.21)
CentOS (2.6.32-573.22.1.el6.i686)
CentOS 6 (2.6.32-504.el6.i686)

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, 'a' to modify the kernel arguments
before booting, or 'c' for a command-line.
```

12. Descargar una plantilla (Nosotros usaremos la plantilla: centos-6-standard_6.3-1_i386.tar.gz).

13. Copiar la plantilla a utilizar en /vz/template/cache/ (directorio predeterminado para almacenar las plantillas descargadas).

14. Crear más de un contenedor. Crearemos el contenedor 101.

```
# vzctl create 101 --ostemplate centos-6-standard_6.3-1_i386
```

Y debe aparecer lo siguiente:

```
Creating image: /vz/private/101.tmp/root.hdd/root.hdd size=2306867K
Creating delta /vz/private/101.tmp/root.hdd/root.hdd bs=2048 size=4614144
sectors v2
```

```
Storing /vz/private/101.tmp/root.hdd/DiskDescriptor.xml
Opening delta /vz/private/101.tmp/root.hdd/root.hdd
Adding delta dev=/dev/ploop32072
img=/vz/private/101.tmp/root.hdd/root.hdd (rw)
mke2fs 1.41.12 (17-May-2010)
Discarding device blocks: hecho
Etiqueta del sistema de ficheros=
Tipo de SO: Linux
Tamaño del bloque=4096 (bitácora=2)
Tamaño del fragmento=4096 (bitácora=2)
Stride=0 blocks, Stripe width=0 blocks
144288 nodos-i, 576256 bloques
28812 bloques (5.00%) reservados para el superusuario
Primer bloque de datos=0
Número máximo de bloques del sistema de ficheros=0
18 bloque de grupos
32768 bloques por grupo, 32768 fragmentos por grupo
8016 nodos-i por grupo
Respaldo del superbloque guardado en Los bloques:
    32768, 98304, 163840, 229376, 294912

Escribiendo las tablas de nodos-i: hecho
Creating journal (32768 blocks): hecho
Escribiendo superbloques y la información contable del sistema de
ficheros: hecho

Este sistema de ficheros se revisará automáticamente cada 21 montajes o
180 días, lo que suceda primero. Utilice tune2fs -c o -i para cambiarlo.
tune2fs 1.41.12 (17-May-2010)
Se pone la cuenta de montajes máxima a -1
Se pone el comportamiento de errores a 2
Se pone el intervalo entre revisiones en 0 segundos
Creating balloon file .balloon-c3a5ae3d-ce7f-43c4-a1ea-c61e2b4504e8
```

```
Mounting /dev/ploop32072p1 at /vz/private/101.tmp/root.hdd/root.hdd.mnt
fstype=ext4 data=''
Unmounting device /dev/ploop32072
Opening delta /vz/private/101.tmp/root.hdd/root.hdd
Adding delta dev=/dev/ploop32072
img=/vz/private/101.tmp/root.hdd/root.hdd (rw)
Mounting /dev/ploop32072p1 at /vz/root/101 fstype=ext4
data='balloon_ino=12,'
Creating container private area (centos-6-standard_6.3-1_i386)
Unmounting file system at /vz/root/101
Unmounting device /dev/ploop32072
Opening delta /vz/private/101/root.hdd/root.hdd
Adding delta dev=/dev/ploop32072 img=/vz/private/101/root.hdd/root.hdd
(rw)
Mounting /dev/ploop32072p1 at /vz/root/101 fstype=ext4
data='balloon_ino=12,'
Performing postcreate actions
Unmounting file system at /vz/root/101
Unmounting device /dev/ploop32072
CT configuration saved to /etc/vz/conf/101.conf
Container private area was created
```

15. Asignamos la dirección IP y un nombre al servidor contenedor.

```
# vzctl set 101 --ipadd 192.168.0.101 --save
# vzctl set 101 --nameserver 8.8.8.8 --save
```

16. Con cada configuración hecha nos mostrara lo siguiente:

```
CT configuration saved to /etc/vz/conf/101.conf
```

17. Donde se guardan las configuraciones de los contenedores

```
# cat /etc/vz/conf/101.conf
```

18. Encendemos el contenedor

```
# vzctl start 101
```

Y debe aparecer una línea como la siguiente:

```
Starting container...
Opening delta /vz/private/101/root.hdd/root.hdd
Error in ploop_check (check.c:574): Dirty flag is set
Adding delta dev=/dev/ploop57989 img=/vz/private/101/root.hdd/root.hdd
(rw)
Mounting /dev/ploop57989p1 at /vz/root/101 fstype=ext4
data='balloon_ino=12,'
Container is mounted
Adding IP address(es): 192.168.0.101
vps-net_add WARNING: Device list is empty
vps-net_add WARNING: Device list is empty
Setting CPU units: 1000
Container start in progress...
```

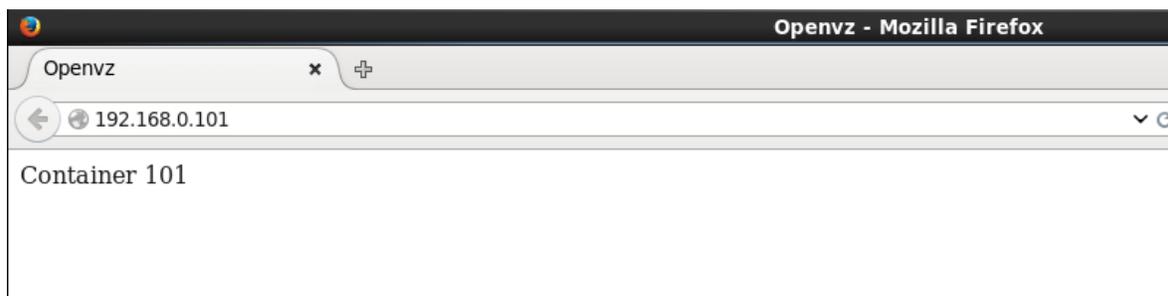
19. Comprobarlo haciendo ping del servidor físico al contenedor

```
# ping 192.168.0.101
```

Y debe aparecer lo siguiente:

```
PING 192.168.0.101 (192.168.0.101) 56(84) bytes of data:
64 bytes from 192.168.0.101: icmp_req=1 ttl=64 time=0.026 ms
64 bytes from 192.168.0.101: icmp_req=2 ttl=64 time=0.030 ms
64 bytes from 192.168.0.101: icmp_req=3 ttl=64 time=0.030 ms
64 bytes from 192.168.0.101: icmp_req=4 ttl=64 time=0.030 ms
```

20. Configurar apache en los contenedores para que desde el navegador web del servidor OpenVZ poder visualizar la página de los contenedores.



21. Configuraremos el servidor OpenVZ y los contenedores para que estos tenga acceso a internet.

- a) Agregar esta regla en el servidor OpenVZ para realizar las traducciones por medio de nat de las ip de los contenedores.

```
# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

- b) Entraremos al contenedor 101, borraremos la ruta por defecto que estos tienen y agregaremos una nueva ruta por defecto. Hacer en cada uno de los contenedores.

```
# vzctl enter 101
# route del default
# route add default dev venet0:0
```

- c) Accederemos al archivo `/etc/resolv.conf` del servidor OpenVZ copiar la ip del nameserver. Agregar la dirección ip en el archivo `/etc/resolv.conf` del contenedor 101. Esta ip es la encargada de resolver nuestras peticiones. Hacer esto en cada uno de los contenedores. Ej: "nameserver ip".
- d) Con esto cada uno los contenedores ya tendrían acceso a internet.
- e) Instalar el paquete ssh en el servidor OpenVZ y en cada uno de los contenedores

- f) Realizar pruebas de conexión a través de ssh desde el servidor OpenVZ hacia los contenedores y entre contenedores.

The image shows three terminal windows illustrating SSH connections:

- Top Left (root@localhost):** Shows the process of logging into container 101 from the host, then logging into container 102 from container 101, and finally logging into container 102 from the host.
- Top Right (root@host-101):** Shows logging into container 101, then logging into container 102 from container 101, and finally logging into container 102 from the host.
- Bottom (root@host-102):** Shows logging into container 102, then attempting to SSH to container 101 (which fails due to a fingerprint warning), and finally logging into container 101 from container 102.

- g) Instalar ssh en la maquina cliente.
- h) Desde la maquina cliente hacer una conexión ssh a la maquina servidor OpenVZ (CentOS).
- i) Repita los incisos del 14 en adelante (estos se realizarán desde la maquina cliente vía ssh). Crear más de un contenedor.
- j) Acceder a cada contenedor y probar ping entre cada uno de estos.
- k) Acceder vía ssh desde la maquina cliente a cada uno de los contenedores.

Algunos comandos básicos de administración bajo *OpenVZ*:

- Detener un contenedor.

```
# vzctl stop id
```

- Añadir un nombre de hostname al contenedor.

```
# vzctl set id --hostname name --save
```

- Asignar un nombre al contenedor.

```
# vzctl set id --name name --save
```

- Asignar una descripción al contenedor.

```
# vzctl set id --description descripcion --save
```

- Asignar contraseña al usuario root.

```
# vzctl exec id passwd
```

- Destruir el contenedor.

```
# vzctl destroy id
```

- Indicar la cantidad de memoria que tendrá el contenedor.

```
# vzctl set id --ram 1G --save
```

- Lista los contenedores que están corriendo.

```
# vzlist -a
```

- Mostrará el uso de la memoria por parte de los contenedores en el sistema.

```
# vzmemcheck
```

- Añadimos el parámetro -v mostrara por separado el uso de memoria de cada contenedor.

```
# vzmemcheck -v
```

Preguntas de control

1. ¿Qué es un contenedor?
2. ¿Qué es root y para qué sirve?
3. Explique ¿Qué es el grub?
4. Razone y Explique cada paso del inciso 21.

Práctica 3: Web Panel OpenVZ.

Introducción:

Web Panel OpenVZ es una interfaz basada en Web GUI para el control del hardware y servidores virtuales con la tecnología de virtualización OpenVZ.

Objetivos:

- Instalación y configuración del Web panel OpenVZ.
- Que la gestión OpenVZ sea lo más fácil de comprender para el usuario.
- Que el usuario no tenga que trabajar en la línea de comandos.
- La persona puede crear, eliminar, administrar máquinas virtuales a través de un navegador web.

Requerimientos:

Hardware	Software
Computadora con los siguientes requisitos: <ul style="list-style-type: none">• Procesador mínimo de velocidad de 2.1 GHz• Memoria RAM de 4 GB.	Software de virtualización VMware con siguientes elementos: <ul style="list-style-type: none">• Router (CentOS 6.6-i386 server)• PC1 Servidor OpenVZ (CentOS 6.6-i386)• PC2 Cliente (lubuntu12.04)

“Por simplicidad se recomienda haber realizado la practica anterior (OpenVZ)”.

Duración de la práctica

- Dos sesiones de laboratorio.

Topología

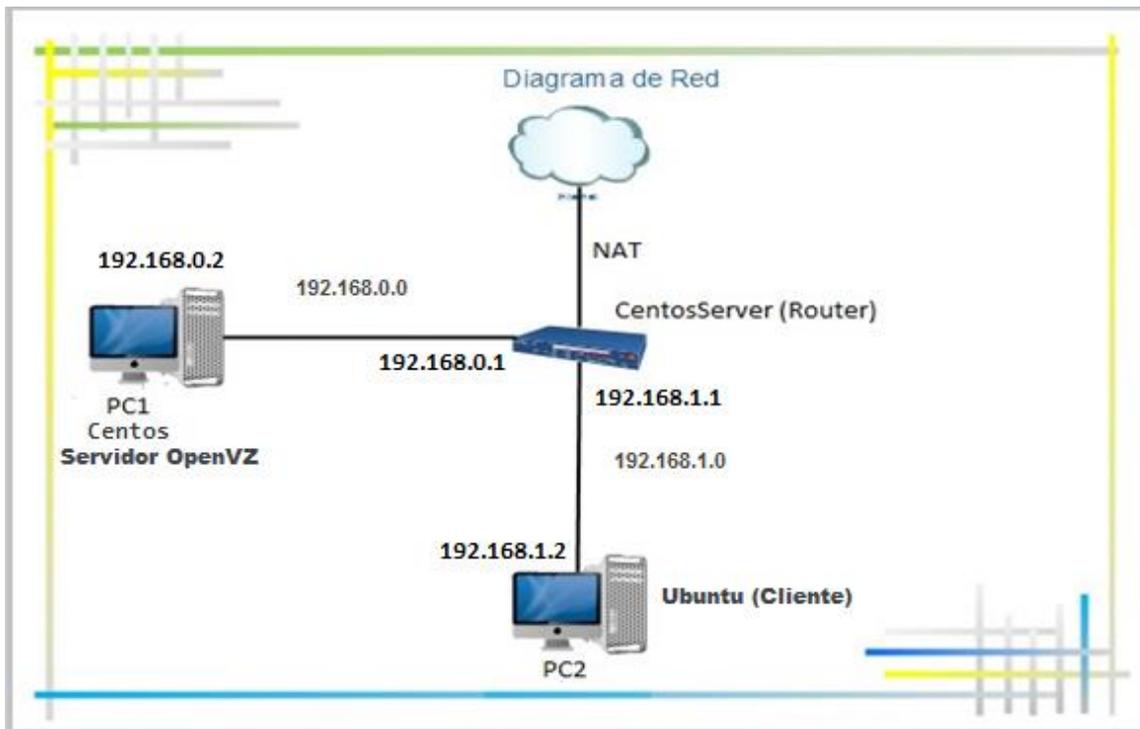


Figura 38 Topología a seguir para la práctica numero 3

Enunciado:

1. Instalar OpenVZ Web Panel.

```
# wget -O - http://ovz-web-panel.googlecode.com/svn/installer/ai.sh | sh
```

2. Si usted está accediendo a la URL desde un sistema remoto, debe permitir el puerto predeterminado web panel 3000 a través de su firewall/router en el archivo `/etc/sysconfig/iptables` agregar la siguiente regla y guardar:

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 3000 -j ACCEPT
```

3. Reiniciar el servicio iptables:

```
# service iptables restart
```

4. Acceder a OpenVZ Web Panel.

5. Abra su navegador web e introduzca la siguiente URL en la barra de direcciones.

```
http: // <ip del host>: 3000
```

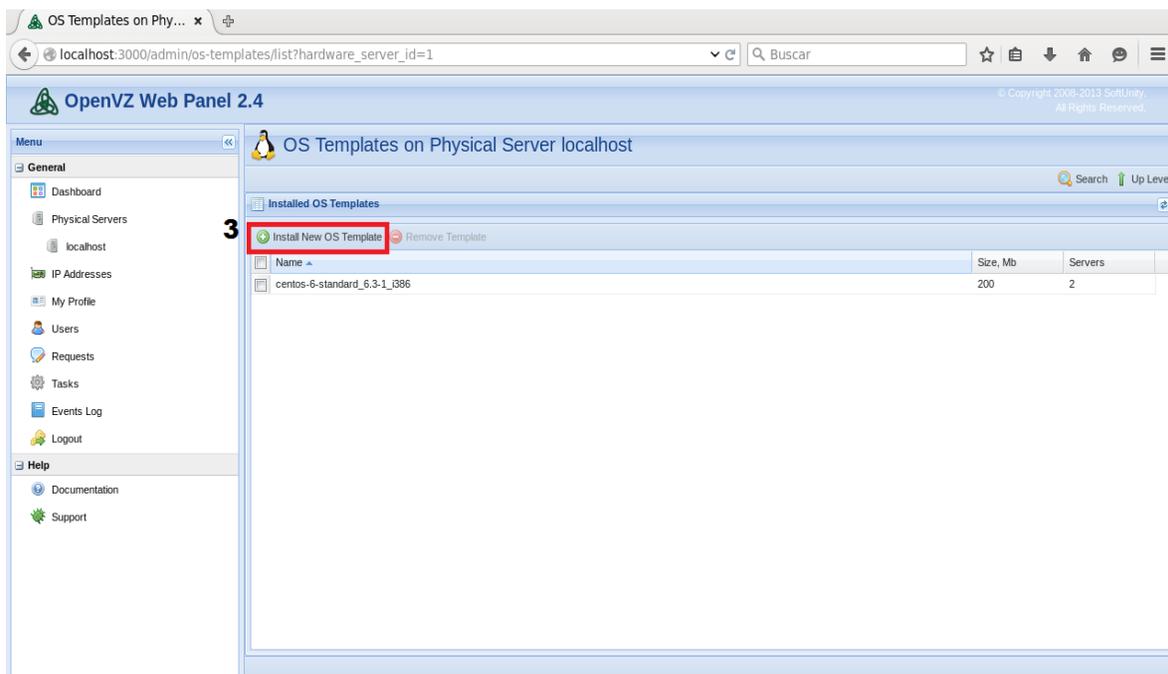
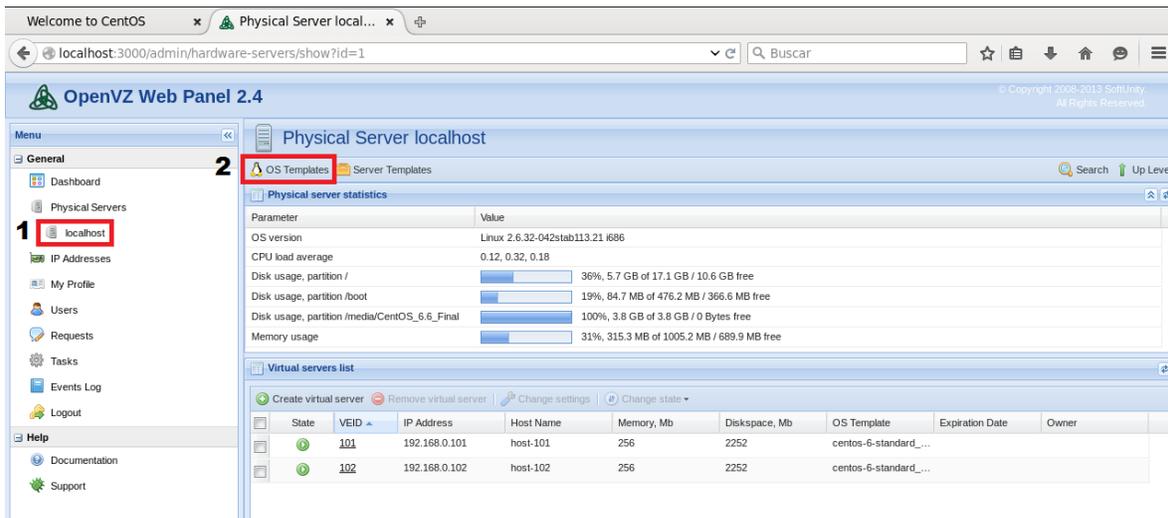
Las credenciales de administrador por defecto son **admin / admin**. No se olvide de cambiar la contraseña por defecto.

6. Cambiar contraseña de usuario predeterminado del administrador.

Para ello, haz clic en el apartado **mi perfil** en el panel izquierdo del tablero de instrumentos. Introduzca la nueva contraseña dos veces y haga clic en **Save**.

7. Descargar e instalar plantillas OS.

a) Antes de la creación de máquinas virtuales o VPS, el primer paso es la instalación de plantillas de SO.



- b) Acceder a la ruta `/opt/ovz-web-panel/config/` y hacer una copia del archivo `config.yml.sample` a `config.yml` y deberá quedar como lo siguiente:

```

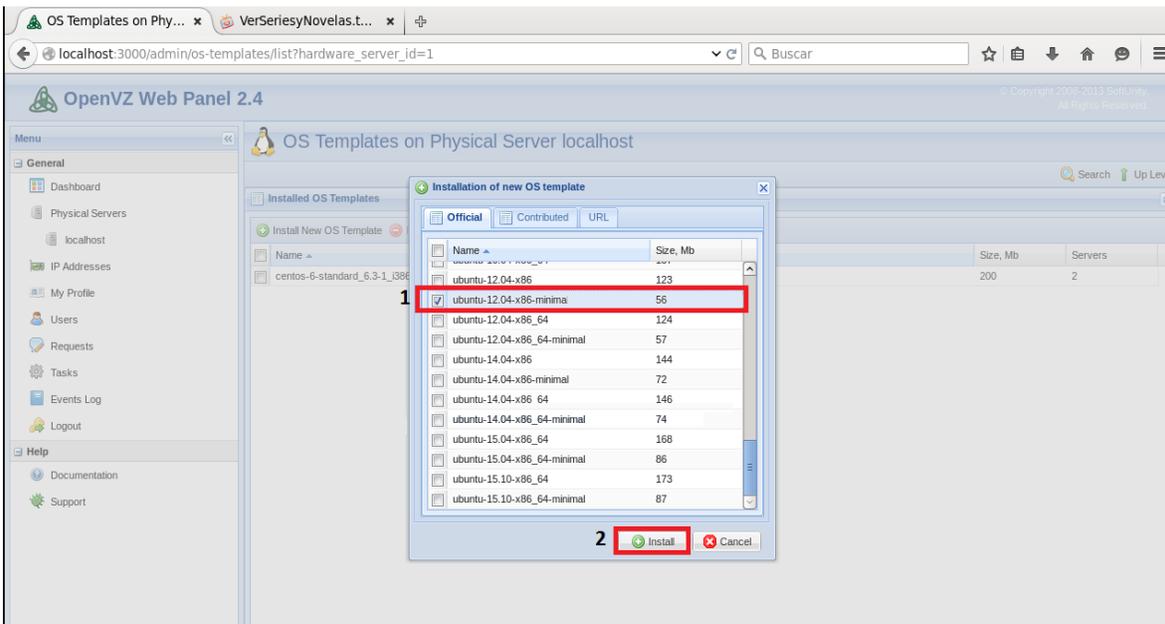
root@localhost:~# nano /opt/ovz-web-panel/config/config.yml
GNU nano 2.0.9 Fichero: /opt/ovz-web-panel/config/config.yml

#w_daemon:
  port: 7767

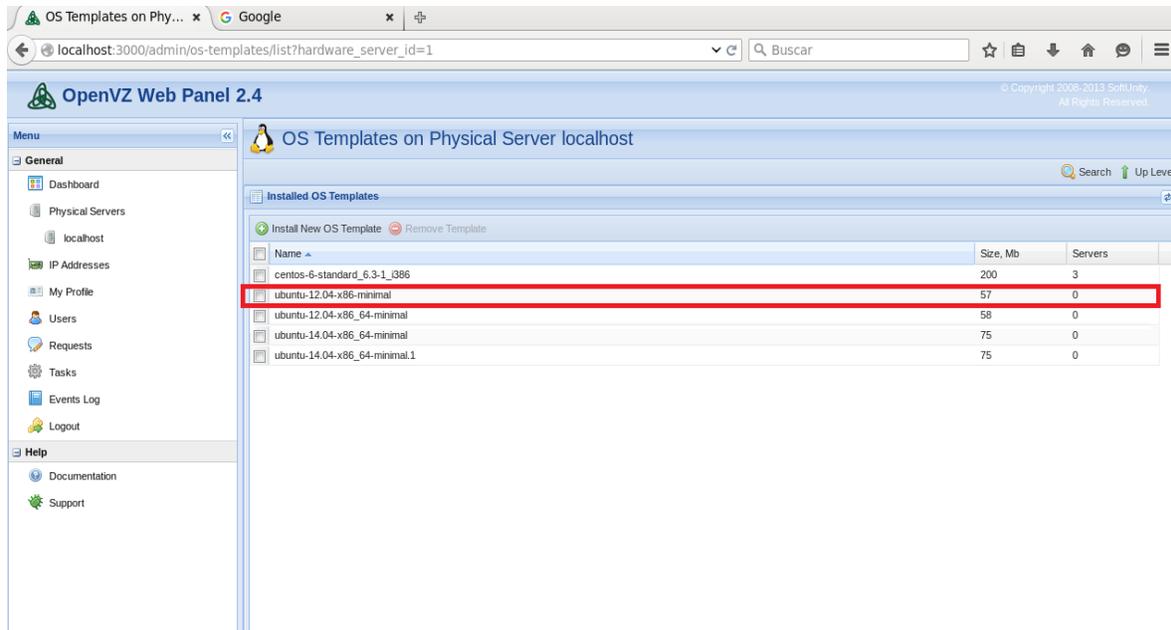
os_templates:
  mirror:
    host: download.openvz.org
    path: /template/

#ip_restriction:
# admin_ips: "192.168.0.1, 192.168.0.2"
    
```

- c) La lista de plantillas de SO disponibles se mostrará en la ventana siguiente. Seleccione la plantilla de SO de su elección y haga clic en el botón Instalar.

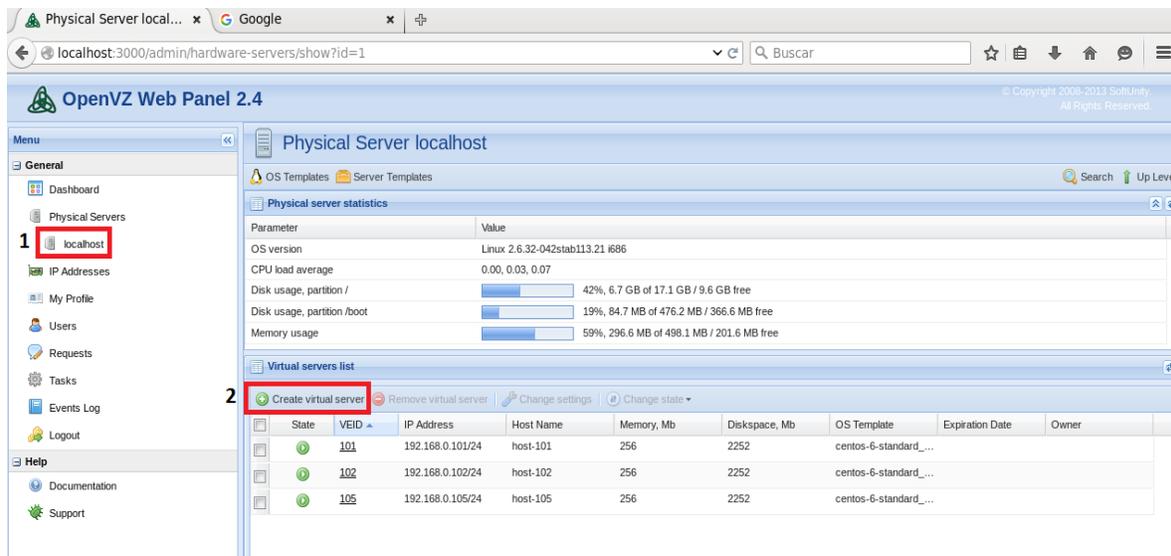


- d) Después de unos minutos (dependiendo del tamaño de la plantilla), actualizar la lista de plantillas de SO. Allí podrás ver las plantillas de SO recién instalados.



8. La creación de servidores virtuales.

- a) Hemos instalado las plantillas de SO en nuestros pasos anteriores. Ahora, vamos a crear nuevos servidores virtuales utilizando las plantillas de SO.



- b) Introduzca el ID de servidor, seleccione la plantilla de SO, introduzca la dirección IP y el nombre de host, introduzca la contraseña de root. Además, se puede definir el tamaño de espacio en disco, memoria RAM, Swap, servidor DNS, entre otros, en los ajustes adicionales.

The screenshot shows the OpenVZ Web Panel 2.4 interface. The left sidebar contains a menu with options like Dashboard, Physical Servers, IP Addresses, My Profile, Users, Requests, Tasks, Events Log, Logout, Documentation, and Support. The main content area displays 'Physical Server localhost' with various statistics and a 'Virtual servers list' table. The table has columns for State, VEID, IP Address, Host Name, Memory, Diskspace, OS Template, Expiration Date, and Owner. The row for VEID 106 is highlighted with a red box.

State	VEID	IP Address	Host Name	Memory, Mb	Diskspace, Mb	OS Template	Expiration Date	Owner
🟢	101	192.168.0.101/24	host-101	256	2252	centos-6-standard_6.3-1_i...		
🟢	102	192.168.0.102/24	host-102	256	2252	centos-6-standard_6.3-1_i...		
🟢	105	192.168.0.105/24	host-105	256	2252	centos-6-standard_6.3-1_i...		
🟢	106	192.168.0.106/24	host-106	256	2252	ubuntu-12.04-x86-minimal		

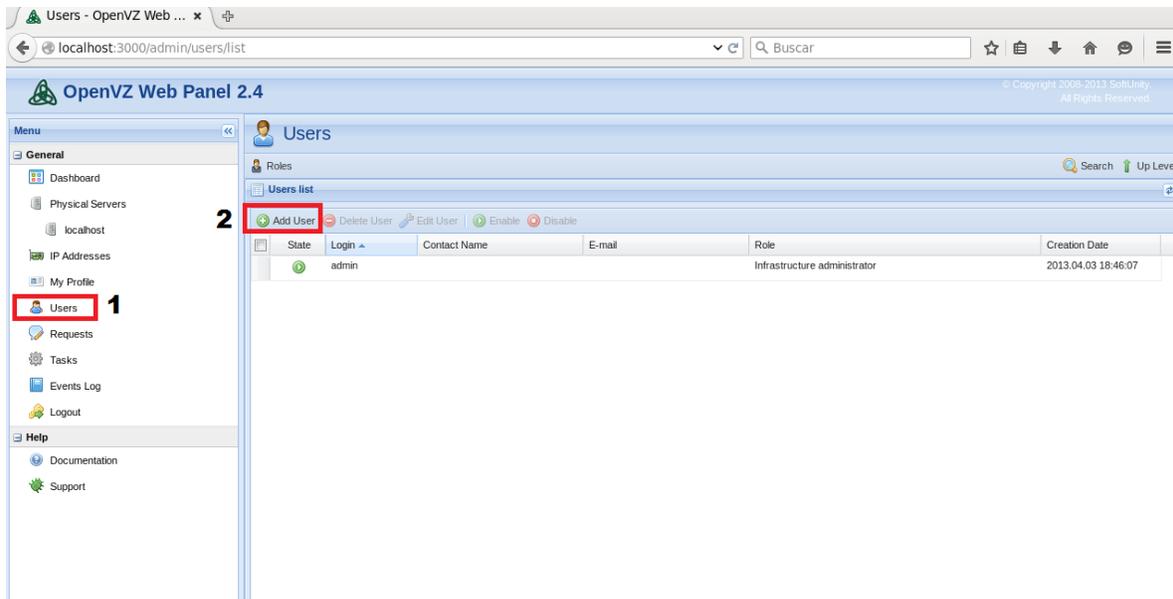
This is a close-up view of the 'Virtual servers list' table from the previous screenshot. The row for VEID 106 is highlighted with a red box.

State	VEID	IP Address	Host Name	Memory, Mb	Diskspace, Mb	OS Template	Expiration Date	Owner
🟢	101	192.168.0.101/24	host-101	256	2252	centos-6-standard_6.3-1_i...		
🟢	102	192.168.0.102/24	host-102	256	2252	centos-6-standard_6.3-1_i...		
🟢	105	192.168.0.105/24	host-105	256	2252	centos-6-standard_6.3-1_i...		
🟢	106	192.168.0.106/24	host-106	256	2252	ubuntu-12.04-x86-minimal		

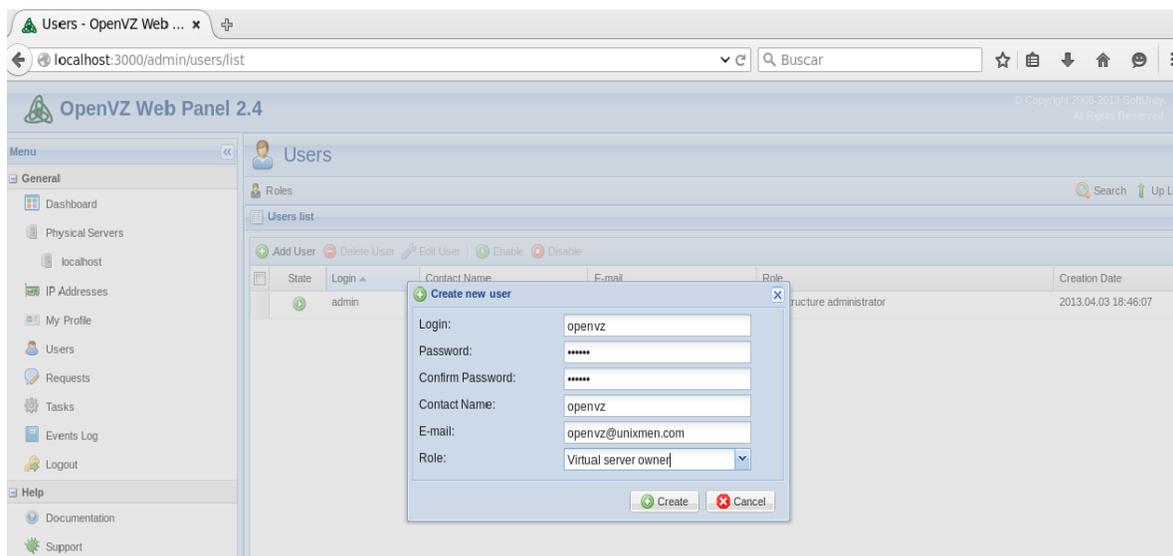
- c) Si no desea el servidor virtual, haga clic en **Remove virtual server**. También, usted puede cambiar la configuración del servidor virtual en **Change setting**. Al igual que, puede iniciar/detener/reiniciar el servidor virtual en **Change state**.

9. Creación de usuarios.

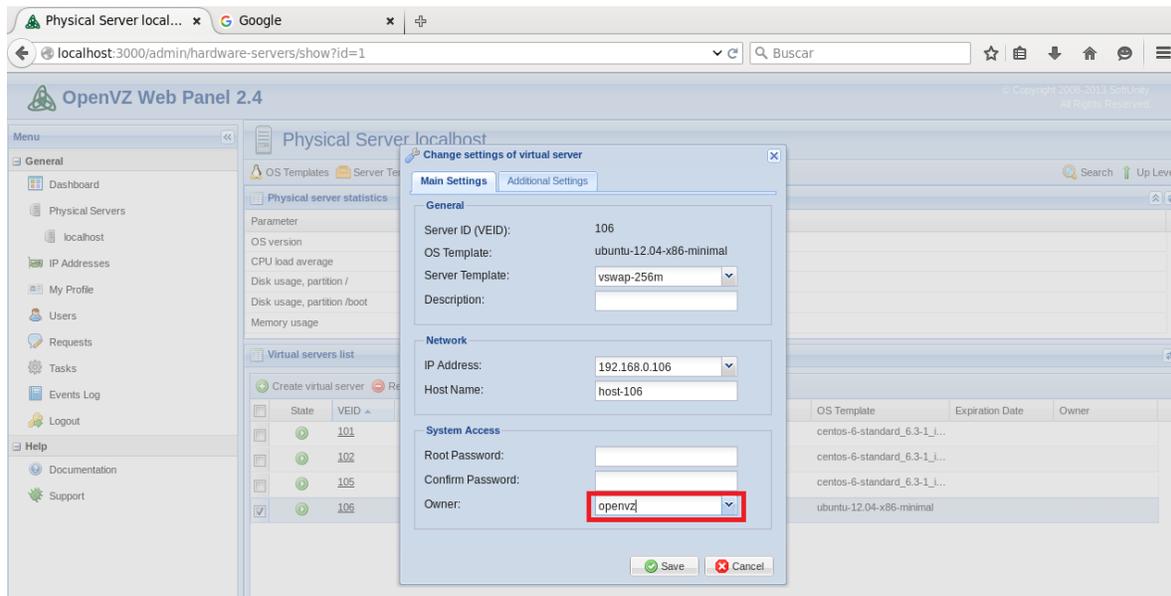
- a) Por defecto, habrá dos usuarios creados automáticamente durante la instalación de OpenVZ. Uno es administrador, (administrador de la infraestructura), y la otra es propietario servidor (virtual). Si quieres nuevos usuarios, puede agregar también. Además, puede asignar cualquier usuario como propietario para cualquier VPS. Los propietarios pueden gestionar, editar los detalles de VPS particulares.



- b) Rellene los detalles de usuario y seleccione el rol del nuevo usuario (propietario de la infraestructura o el propietario del servidor virtual). Por último, haga clic en Crear.



c) Podemos asignar el nuevo usuario a cualquier VPS.



10. Repita los incisos del 8 en adelante (estos se realizarán desde la maquina cliente).
Crear más de un contenedor. Probar su funcionamiento (haciendo ping los contenedores, Acceder a vía a los contenedores, Acceder vía web a los contenedores y hacer ping entre contenedores)

Preguntas de control

1. Explique con sus palabras ¿qué es open VZ web panel?
2. ¿Qué facilidades ofrece Open VZ web panel?
3. ¿Qué es un VPS?
4. ¿Qué es una plantilla y para qué sirve?

Práctica 4: RAID.

Introducción:

Redundant Array of Independent Disks (RAID) es un sistema que permite combinar el almacenamiento de un grupo de dispositivos independientes, en una única unidad virtual de almacenamiento o múltiples unidades virtuales. Usando las particiones distribuidas en un dispositivo físico y usando *software* RAID basado en Linux vamos a crear arreglos de diferentes niveles. Durante el desarrollo de las actividades de laboratorio tendrá los privilegios de administración del usuario **root** de cada servidor.

Objetivos:

- Manipular de particiones en Linux.
- Crear las particiones.
- Que el usuario tenga la conocimiento de cómo funcionan los raid, para que sepa elegir el tipo de RAID que necesita.

Requerimientos:

Hardware	Software
Computadora con los siguientes requisitos: <ul style="list-style-type: none">• Procesador mínimo de velocidad de 2.1 GHz• Memoria RAM de 2 GB.	Software de virtualización VMware con siguientes elementos: <ul style="list-style-type: none">• PC1 (lubuntu12.04)• VMWare

Duración de la práctica

- Dos sesiones de laboratorio.

Topología

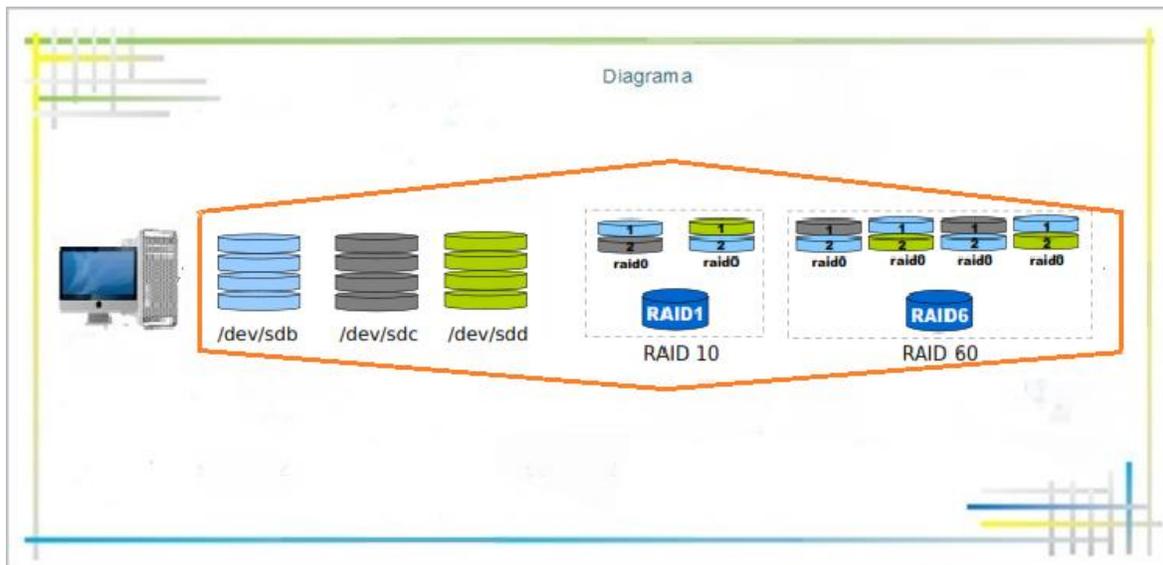


Figura 39 Topología a seguir para la práctica numero 4

Comandos de ayuda

Comando	Descripción
# apt-get install mdadm	Instalación del software necesario
# mdadm -C /dev/md0 -a yes -l 0 -n 2 /dev/sda10 /dev/sda11	Crea RAID0
# mdadm -C /dev/md10 -a yes -l 1 -n 2 /dev/md0 /dev/md1	Crea RAID10
# mdadm -C /dev/md60 -a yes -l 6 -n 4 /dev/md2 /dev/md3 /dev/md4 /dev/md5	Crea RAID60
# mdadm /dev/md0 -f /dev/sda10	Simula la falla de un dispositivo del arreglo.
# mdadm /dev/md0 -r /dev/sda10	Elimina un dispositivo del arreglo
# mdadm /dev/md0 -a /dev/sda10	Restaura un dispositivo a el arreglo

Enunciado:

Para la realización de esta práctica se deberán realizar los siguientes pasos:

Verificando que dispositivos están disponibles.

1. Crear 3 discos duro desde la interfaz gráfica de VMWare.
2. En nuestro caso crearemos 12 particiones de 5GB cada una. 4 particiones en /dev/sdb, 4 particiones en /dev/sdc y 4 particiones en /dev/sdd.

3. Usamos un manipulador de particiones en Linux:

```
# fdisk -l
```

4. En nuestro caso serán `/dev/sdb`, `/dev/sdc` y `/dev/sdd`.

```
# fdisk /dev/sdb
```

Usando la opción “m” vemos los comandos necesarios. Deberíamos obtener una salida como lo siguiente:

```
Orden (m para obtener ayuda): m
Orden  Acción
a      Conmuta el indicador de iniciable
b      Modifica la etiqueta de disco bsd
c      Conmuta el indicador de compatibilidad con DOS
d      Suprime una partición
l      Lista los tipos de particiones conocidos
m      Imprime este menú
n      Añade una nueva partición
o      Crea una nueva tabla de particiones DOS vacía
p      Imprime la tabla de particiones
q      Sale sin guardar los cambios
s      Crea una nueva etiqueta de disco Sun
t      Cambia el identificador de sistema de una partición
u      Cambia las unidades de visualización/entrada
v      Verifica la tabla de particiones
w      Escribe la tabla en el disco y sale
x      Funciones adicionales (sólo para usuarios avanzados)
```

Usemos las siguientes opciones:

- `n` creamos una nueva partición
- `t` definamos el tipo de partición en nuestro caso `fd` (con el tipo `fd` garantizamos que el `kernel` reconozca el arreglo)
- `w` escribimos los cambios anteriores.

5. Este ejemplo les ayudara a crear particiones pero solo para `/dev/sdb` se deberán las particiones tanto en `/dev/sdc` y `/dev/sdd`.

```

Orden (m para obtener ayuda): n
Partition type:
p  primary (0 primary, 0 extended, 4 free)
e  extended
Select (default p): p
Número de partición (1-4, valor predeterminado 1): 1
Primer sector (2048-46137343, valor predeterminado 2048):
Se está utilizando el valor predeterminado 2048
Último sector, +sectores o +tamaño{K,M,G} (2048-46137343, valor
predeterminado 46137343): +5G

Orden (m para obtener ayuda): t
Se ha seleccionado la partición 1
Código hexadecimal (escriba L para ver los códigos): fd
Se ha cambiado el tipo de sistema de la partición 1 por fd (Linux
raid autodetect)

Orden (m para obtener ayuda): w
¡Se ha modificado la tabla de particiones!

Llamando a ioctl() para volver a leer la tabla de particiones.
Se están sincronizando los discos.

```

Crear los RAID

1. Reiniciemos nuestra PC e instalamos el siguiente paquete.

```
# apt-get install mdadm
```

Una vez que contemos con las particiones necesarias vamos a crear RAID0, RAID10 y RAID60 usando una de las herramientas de software en Linux.

2. Creamos los RAID0 (repita este paso con cada una de las particiones necesarias).

```
# mdadm -C /dev/md0 -a yes -l 0 -n 2 /dev/sdb1 /dev/sdb2
```

Y nos mostrara el siguiente mensaje:

```
mdadm: Defaulting to version 1.2 metadata
```

```
mdadm: array /dev/md0 started.
```

3. Verifique que el arreglo se creó y algunos parámetros generales.

```
# cat /proc/mdstat
```

Y nos mostrara el siguiente mensaje:

```
Personalities : [raid0]
md5 : active raid0 sdd4[1] sdd3[0]
      10484736 blocks super 1.2 512k chunks

md4 : active raid0 sdd2[1] sdd1[0]
      10484736 blocks super 1.2 512k chunks

md3 : active raid0 sdc4[1] sdc3[0]
      10484736 blocks super 1.2 512k chunks

md2 : active raid0 sdc2[1] sdc1[0]
      10484736 blocks super 1.2 512k chunks

md1 : active raid0 sdb4[1] sdb3[0]
      10484736 blocks super 1.2 512k chunks

md0 : active raid0 sdb2[1] sdb1[0]
      10484736 blocks super 1.2 512k chunks
```

4. Crear arreglos híbridos usando los RAID0 ya creados en lugar de particiones del disco.

4.1 Crear RAID10.

```
# mdadm -C /dev/md10 -a yes -l 1 -n 2 /dev/md0 /dev/md1
```

Y nos mostrara el siguiente mensaje:

```
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md10 started.
```

Y nos mostrara el siguiente mensaje al hacer `cat /proc/mdstat`:

```
Personalities : [raid0] [raid1]
md10 : active raid1 md1[1] md0[0]
        10476416 blocks super 1.2 [2/2] [UU]
        [>.....] resync = 4.3% (457856/10476416)
finish=1.0min speed=152618K/sec
```

4.2 Crear RAID60.

```
# mdadm -C /dev/md60 -a yes -l 6 -n 4 /dev/md2 /dev/md3 /dev/md4 /dev/md5
```

Y nos mostrara el siguiente mensaje:

```
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md60 started.
```

Y nos mostrara el siguiente mensaje al hacer `cat /proc/mdstat`:

```
Personalities : [raid0] [raid1] [raid6] [raid5] [raid4]
md60 : active raid6 md5[3] md4[2] md3[1] md2[0]
        20952064 blocks super 1.2 level 6, 512k chunk, algorithm 2
        [4/4] [UUUU]
        [=>.....] resync = 9.7% (1018916/10476032)
finish=5.2min speed=30029K/sec
```

Realizar pruebas básicas de RAID.

1. Simulando la falla de un dispositivos del arreglo.

```
# mdadm /dev/md0 -f /dev/sdb1
```

2. Eliminado un dispositivo del arreglo:

```
# mdadm /dev/md0 -r /dev/sdb1
```

3. Restaurando un dispositivo a el arreglo:

```
# mdadm /dev/md0 -a /dev/sdb1
```

Preguntas de control

1. Explique ventajas y desventaja con respecto a RAID por hardware vs RAID por software
2. Enumere los beneficios de utilizar RAID
3. Que es striping, mirroring, parity?
4. En que consiste RAID 60?

Nota: En el apartado 1 vemos diferentes dispositivos físicos instalados y sus particiones. En esta práctica vamos a crear algunos arreglos usando particiones de un mismo disco (ojo esto solo con fines prácticos a nivel de producción. No tiene sentido crear arreglos con particiones de un mismo dispositivo físico).

Práctica 5: GlusterFS.

Introducción:

Es un sistema de archivos de alta disponibilidad y escalabilidad que puede brindar almacenamiento a gran escala. Durante el desarrollo de las actividades de laboratorio tendrá los privilegios de administración del usuario *root* de cada servidor.

Objetivos:

- Conocer el funcionamiento de GlusterFS.
- Instalación y configuración de GlusterFS (Servidores y Clientes).
- Que el usuario obtenga la capacidad de poder combinar diversos dispositivos a través de la red.

Requerimientos:

Hardware	Software
Computadora con los siguientes requisitos: <ul style="list-style-type: none">• Procesador mínimo de velocidad de 2.1 GHz• Memoria RAM de 4 GB.	Software de virtualización VMware con siguientes elementos: <ul style="list-style-type: none">• PC Servidor 1 (lubuntu12.04)• PC Servidor 2 (lubuntu12.04)• PC Cliente 1 (lubuntu12.04)• PC Cliente 2 (lubuntu12.04)

Duración de la práctica

- Dos sesiones de laboratorio.

Topología

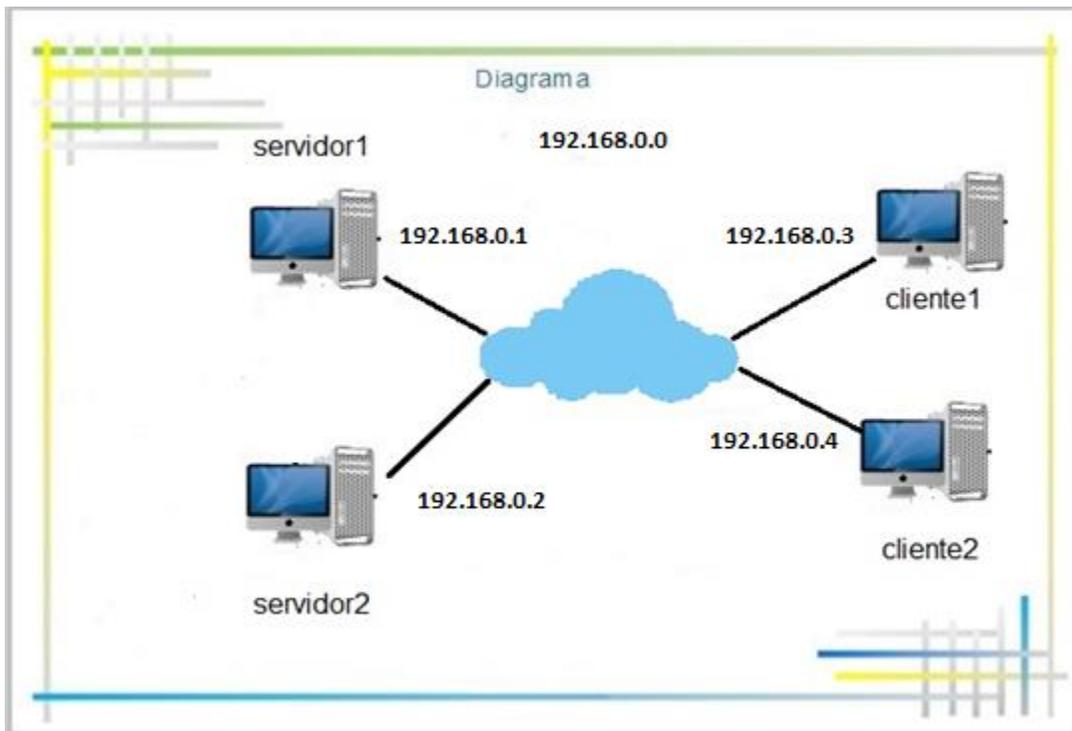


Figura 40 Topología a seguir para la práctica numero 5

Comandos de ayuda	
Comando	Descripción
<code># apt-get install glusterfs-server</code>	Instalación del software necesario
<code># update-rc.d glusterfs-server defaults</code>	Agrega el servicio glusterfsd al boot.
<code># vi /etc/hosts</code>	Accede al fichero hosts.
<code># mkdir /directorio/</code>	Crea los directorios.
<code># /etc/init.d/glusterfs-server start</code>	Arranca el servidor glusterFS.
<code># apt-get install glusterfs-client</code>	Instalación del software necesario
<code># mkdir /mnt/glusterfs</code>	Crea el punto de montaje.
<code># df -h</code>	Verifica los sistemas de archivos actuales.
<code># mount</code>	Verifica el punto de montaje.
<code># apt-get install Portmap</code>	Portmap es el mapeador de puertos de linux. Este mapeador se encarga de gestionar la asignación entre aplicaciones y puertos.

Enunciado:

1. En primera instancia configuraremos 2 servidores que compartirán directorios con 2 clientes usando *GlusterFS* como sistema de archivo en un ambiente distribuido.
2. Instalar *glusterfs-server* en los servidores.
3. Agregamos el servicio *glusterfsd* al *boot*.
4. Configurar los servidores GlusterFS.
 - a) Agregar IP de los servidores, clientes y los nombres de los servidores, clientes en el fichero *hosts*.

```
192.168.0.1    server1-virtual-machine
192.168.0.2    server2-virtual-machine
192.168.0.3    client1-virtual-machine
192.168.0.4    client2-virtual-machine
```

- b) Crear los directorios que vamos a compartir usando *GlusterFS*.
- c) Editamos el archivo de configuración.

```
# vi /etc/glusterfs/glusterd.vol
```

- d) Arrancamos el servidor *glusterFS*.
- e) Verificar que nuestro servicio está corriendo y que está escuchando.

```
netstat -tap | grep glusterd
```

Y debe aparecer una línea como la siguiente:

```
tcp        0      0  *:24007                *.*                    ESCUCHAR
          1125/glusterd
```

5. Instalar *glusterfs-client* en los clientes.
6. Configurar los clientes GlusterFS
 - a) Agregar IP de los servidores, clientes y los nombres de los servidores, clientes en el fichero *hosts*.

```
192.168.0.1    server1-virtual-machine
192.168.0.2    server2-virtual-machine
192.168.0.3    client1-virtual-machine
192.168.0.4    client2-virtual-machine
```

- b) Crear el directorio para el montaje.
- c) Editamos el archivo de configuración

```
# vi /etc/glusterfs/glusterfs.vol
```

- d) Montamos el volumen lógico en el punto de montaje creado previamente.

```
# glusterfs -f /etc/glusterfs/glusterfs.vol /mnt/glusterfs
```

e) Verificamos el punto de montaje.

Y debe aparecer una línea como la siguiente:

```
/etc/glusterfs/glusterfs.vol on /mnt/glusterfs type fuse.glusterfs  
(rw,allow_other,default_permissions,max_read=131072)
```

f) Verificamos los sistemas de archivos.

Y debe aparecer una línea como la siguiente:

```
/etc/glusterfs/glusterfs.vol  
22G 4.9G 16G 25% /mnt/glusterfs
```

7. Una vez los servidores y los clientes estén configurados y probar que todo funciona correctamente.

a) Verificamos que en el punto de montaje se está compartiendo el volumen de los servidores:

```
# cd /mnt/glusterfs  
# touch helloWorld  
# ls  
# helloWorld
```

b) Esto debería ser consistente con los datos en los servidores:

```
# ls /datos/exporta  
# helloWorld
```

Preguntas de control

1. ¿Qué es un cliente? ¿Qué es un servidor?
2. ¿Cuál es la ventaja de utilizar GlusterFS?
3. Explique los métodos de acceso a la Data
4. ¿Qué son los bricks?
5. Explique los Tipos de volúmenes que posee GlusterFS?

Práctica 6: Heartbeat.

Introducción:

Es un demonio que proporciona la infraestructura de clúster de servicios a sus clientes. En este laboratorio configuraremos Heartbeat logrando en nuestro caso que dos servidores Web sirvan de failover en caso de fallas. Durante el desarrollo de las actividades de laboratorio tendrá los privilegios de administración del usuario root de cada servidor.

Objetivos:

- Obtener conocimientos sobre Heartbeat.
- Instalación y configuración de Heartbeat.
- Conocer el concepto Heartbeat para implementar en servicios que requieran alta disponibilidad y alto rendimiento.

Requerimientos:

Hardware	Software
Computadora con los siguientes requisitos: <ul style="list-style-type: none">• Procesador mínimo de velocidad de 2.1 GHz• Memoria RAM de 4 GB.	Software de virtualización VMware con siguientes elementos: <ul style="list-style-type: none">• Router (CentOS 6.6-i386 server)• PC Servidor principal (lubuntu12.04)• PC Servidor failover (lubuntu12.04)• PC1 Cliente (lubuntu12.04)

Duración de la práctica

- Dos sesiones de laboratorio.

Topología

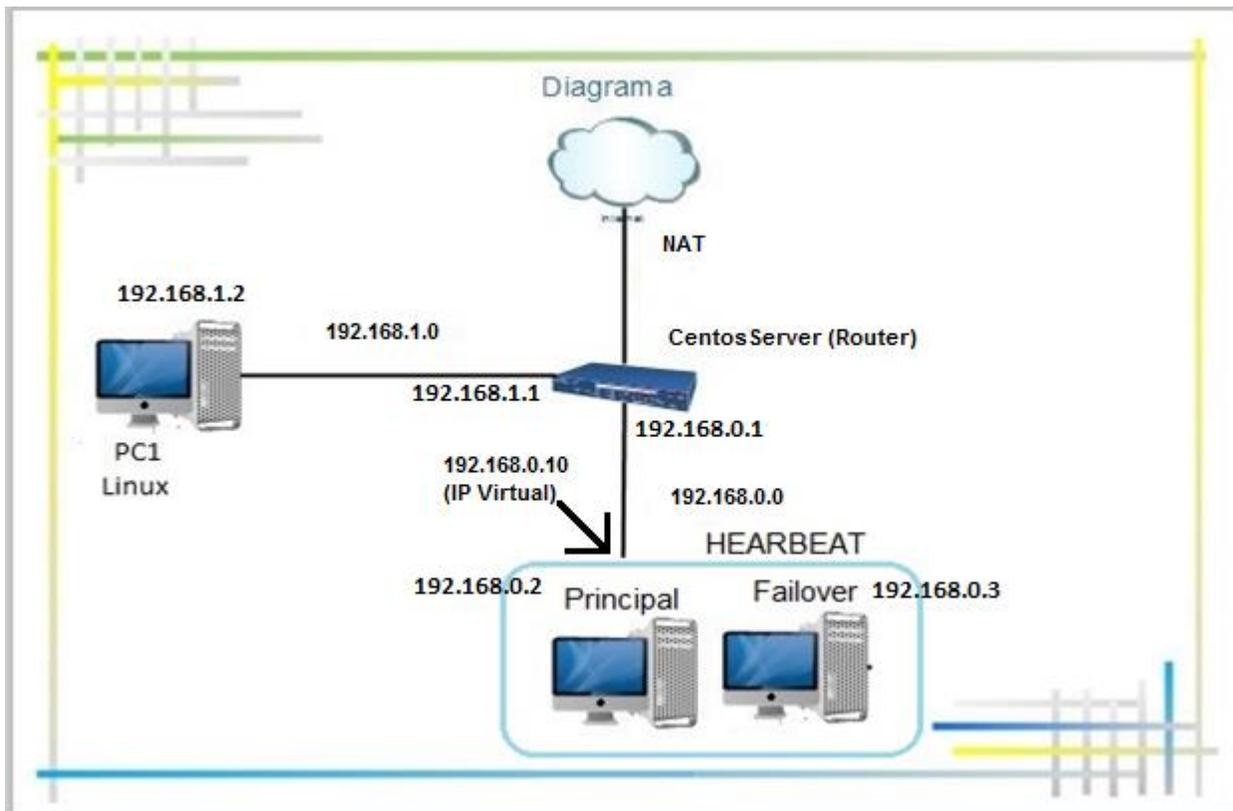


Figura 41 Topología a seguir para la práctica número 6

Comandos de ayuda

Comando	Descripción
<code># apt-get install heartbeat</code>	Instalación del software necesario.
<code># /etc/init.d/heartbeat start tail -f /var/log/ha-log</code>	Iniciar el servicio <i>heartbeat</i> y revisar el log del servicio.

Enunciado:

1. Instalar *Heartbeat* la aplicación que nos permite configurar y administrar.
2. En cada uno de los servidores copiar y descomprimir los archivos de configuración.

```
# cd /usr/share/doc/heartbeat
# cp authkeys ha.cf.* haresources.* /etc/ha.d/
# cd !$
# gunzip haresources.gz
# gunzip ha.cf.gz
```

3. Para cada uno de los dos servidores configurar los parámetros básicos de balanceo. Debemos tener cuidado con las direcciones IP asignadas.
4. En este ejemplo el servidor principal tiene IP 192.168.0.2 y el servidor failover 192.168.0.3. La IP virtual para el servicio es 192.168.0.10.
5. En el nodo servidor Principal (192.168.0.2):

```
# vi /etc/ha.d/ha.cf
```

Y deberá tener las siguientes configuraciones:

```
keepalive 2
logfile /var/log/ha-log
deadtime 9
warntime 5
initdead 15
udpport 694
auto_failback on
ucast eth0 192.168.0.3
node "name pc principal"
node "name pc failover"
```

6. Configurar `/etc/ha.d/haresources` de la siguiente forma:

```
"name pc principal" IPaddr::192.168.0.10 apache2
```

7. Configurar `/etc/ha.d/authkeys` de la siguiente forma:

```
auth 1
#1 crc
1 sha1 HI!
#3 md5 Hello!
```

8. Cambiar los permisos de `authkeys`

```
# chmod 600 /etc/ha.d/authkeys
```

9. Configurar `/etc/apache2/apache2.conf` de la siguiente forma:

```
Listen 192.168.0.10:80
```

10. Configurar `/etc/apache2/ports.conf` de la siguiente forma:

```
#NameVirtualHost *:80
```

```
#Listen 80
```

11. Configurar `/etc/hosts` de la siguiente forma:

```
127.0.0.1    localhost
127.0.1.1    "name pc principal" drbd1
192.168.0.3  "name pc failover"  drbd2
192.168.0.10 test
```

12. La configuración de ambos servidores es prácticamente la misma, sólo cambia en el archivo `/etc/ha.d/ha.cf` y el archivo `/etc/hosts`.

13. En servidor failover (192.168.0.3) solo hay que acceder a `/etc/ha.d/ha.cf` y cambiar la siguiente línea:

```
ucast eth0 192.168.0.2
```

14. Configurar `/etc/hosts` de la siguiente forma:

```
127.0.0.1    localhost
127.0.1.1    "name pc failover"  drbd2
192.168.0.2  "name pc principal"  drbd1
192.168.0.10 test
```

15. Iniciar el servicio heartbeat en ambos servidores y revisar el log del servicio.

16. Verificar que la interfaz para heartbeat este corriendo:

```
# ifconfig
```

Y debe aparecer una línea como la siguiente:

```
eth0:0    Link encap:Ethernet direcciónHW 00:0c:29:19:64:ad
          Direc. inet:192.168.0.10 Difus.:192.168.0.255
Másc:255.255.255.0
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500
          Métrica:1
          Interrupción:19 Dirección base: 0x2000
```

17. Probar Heartbeat.

Es necesario que desde una PC que funcione como cliente y usando un navegador accedemos al URL: **http://192.168.0.10**. Apague o suspenda el equipo servidor principal para detener el servicio para que heartbeat levante al servidor failover y

pase a funcionar sin que el cliente se cuenta que ha fallado el servidor principal y chequee el log del servidor failover. Desde el cliente vuelva a cargar la URL desde un navegador y observaremos que el que nos responde es el servidor failover.

Preguntas de control

1. ¿Qué es un clúster?
2. ¿Cuál es la función de apache2?
3. Explique a que se le conoce como demonio
4. Mencione las ventajas de Heartbeat
5. Explique con sus palabras el funcionamiento de Heartbeat

Nota: Antes de comenzar el desarrollo de ésta práctica, por favor asegúrese de que el servicio web (apache2) en ambos servidores no esté corriendo.

Práctica 7: LVS.

Introducción:

El balanceador de carga dirige las conexiones de red de los clientes que conocen una única dirección IP para los servicios, a un conjunto de servidores que son los que realmente realizan el trabajo. En este laboratorio configuraremos Linux Virtual Server (LVS) en modo Direct Routing (DR) de esta forma lograremos el balanceo de carga entre diferentes servidores Web. Durante el desarrollo de las actividades de laboratorio tendrá los privilegios de administración del usuario **root** de cada servidor.

Objetivos:

- Obtener conocimientos sobre LVS.
- Instalación (servicio) y configuración (servidores).
- Que el usuario haga uso óptimo del director (load balancer) para obtener el máximo desempeño del clúster.

Requerimientos:

Hardware	Software
Computadora con los siguientes requisitos: <ul style="list-style-type: none">• Procesador mínimo de velocidad de 2.1 GHz• Memoria RAM de 4 GB.	Software de virtualización VMware con siguientes elementos: <ul style="list-style-type: none">• PC Director(lubuntu12.04)• PC Servidor real (lubuntu12.04)• PC Servidor real (lubuntu12.04)• PC Cliente (lubuntu12.04)

Duración de la práctica

- Dos sesiones de laboratorio.

Topología

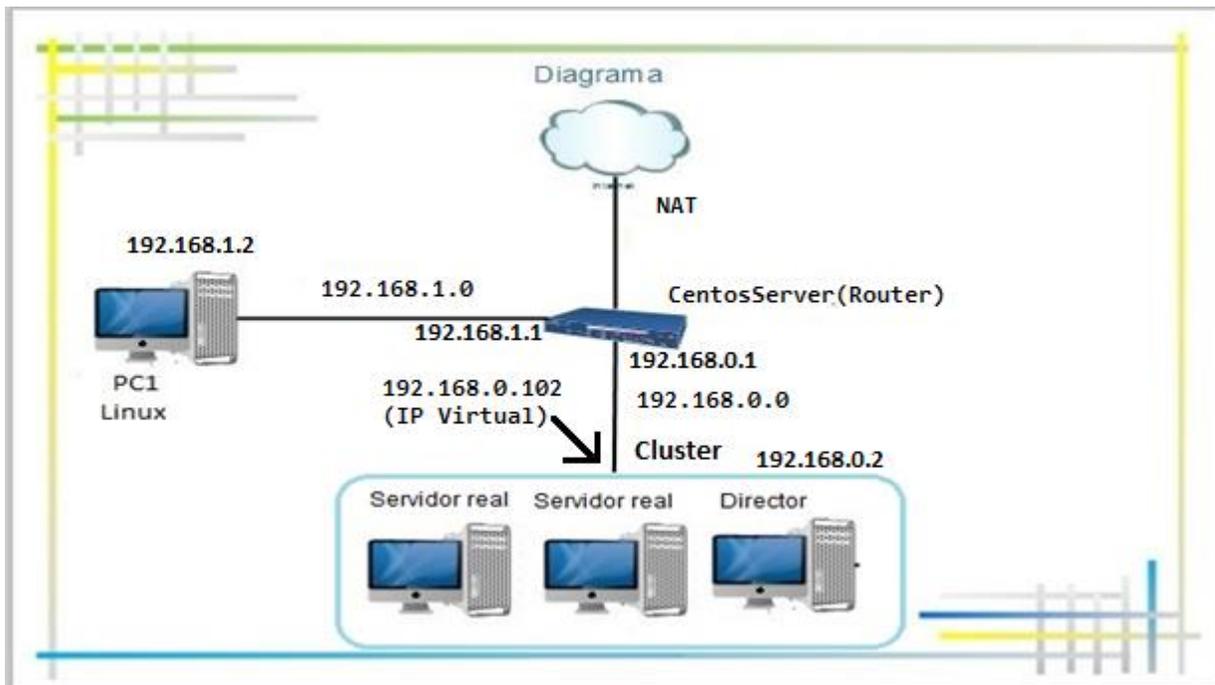


Figura 42 Topología a seguir para la práctica numero 7

Comandos de ayuda	
Comando	Descripción
# apt-get install ipvsadm	Instalación del software necesario.
# ipvsadm -A -t 192.168.0.117:80 -s rr	Configura la tabla de servidores Linux.
# apt-get install apache2	Instala el servicio Web usando apache2.
# /etc/init.d/apache2 restart	Reinicia el servicio de apache2.
# ipvsadm -l -stats	Verifica la tabla de servidores Linux en el Director.

Enunciado:

Configurando el servidor Director

1. Instalar ipvsadm que nos permite configurar y administrar la tabla de servidores.

```
# aptitude install ipvsadm
```

2. Activar la opción que permita el reenvío de paquetes IP. Des comentar la línea `net.ipv4.ip_forward = 1` quitando el #.

```
# nano /etc/sysctl.conf
```

```
# sysctl -p
```

Deberíamos obtener una salida como la siguiente indicando que el reenvío de paquetes está activo:

```
net.ipv4.ip_forward = 1
```

2. Configurar la dirección IP de la PC. En este ejemplo la dirección IP real es 192.168.0.2 por lo tanto la dirección IP virtual sería 192.168.0.102.
3. Configurar la tabla de servidores Linux.

```
# ipvsadm -A -t 192.168.0.102:80 -s rr
```

4. Agregar a la tabla la dirección de los servidores reales (en este caso son los demás PC de su grupo de laboratorio).

```
# ipvsadm -a -t 192.168.0.102:80 -r 192.168.0.3:80 -g
```

```
# ipvsadm -a -t 192.168.0.102:80 -r 192.168.0.4:80 -g
```

Configurar los Servidores Reales.

1. Instalar apache2.

```
# apt-get install apache2
```

2. En cada servidor editar el archivo index.html para identificar cada uno de los servidores. Ejemplo: Servidor Real 1, Servidor Real 2.

```
# cd /var/www/  
# nano index.html
```

3. Reiniciar el servicio apache2.

```
# /etc/init.d/apache2 restart
```

4. Activar la opción que permita el reenvío de paquetes IP. Des comentar la línea `net.ipv4.ip_forward = 1` quitando el #.

```
# nano /etc/sysctl.conf  
# sysctl -p
```

Deberíamos obtener una salida como la siguiente indicando que el reenvío de paquetes está activo:

```
net.ipv4.ip_forward = 1
```

- Configurar la dirección IP pero ahora en cada Servidor Real. En este ejemplo la dirección IP para el servidor real 1 es 192.168.0.3 y para el servidor real 2 es 192.168.0.4 estos tendrán como puerta de salida la IP del servidor director en este caso es 192.168.0.2.

Probar *Ivs-DR*.

- Verificamos la tabla de servidores Linux.

```
# ipvsadm -l --stats
```

Deberíamos obtener una salida parecida a la siguiente:

```
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port      Conns   InPkts  OutPkts  InBytes  OutBytes
-> RemoteAddress:Port
TCP 192.168.0.102:http        11      183     107      38395    30008
-> 192.168.0.3:http         5        31      23       3447     2959
-> 192.168.0.4:http         6       152      84      34948    27049
```

- Para seguir probando es necesario que desde una PC que funcione como cliente se abran diferentes pestañas del navegador con el URL **http://192.168.0.102**. Mientras más pestañas (conexiones http) abra y las actualice constantemente se mostrara la conexión de forma alternada a los Servidores Reales. Baje el servicio de apache2 en alguno de los servidores mientras navega y verifique que se está usando el otro servidor que aún tiene apache2 corriendo.

Verifique constantemente en el Director la tabla de servidores Linux y compare como varia el número de conexión a cada servidor (cuando trabajan los dos servidores y cuando trabaja solo uno).

Preguntas de control

- En que consiste el balanceo de carga
- ¿Cómo es posible obtener el balanceo? Explique
- ¿Es necesario que los servidores de encuentren en la misma red fisica? Argumente su respuesta
- ¿Qué sucedería si el balanceador o director falla?

Práctica 8: Creación de una pequeña nube.

Introducción:

En esta práctica crearemos una pequeña nube haciendo uso de los conceptos y herramientas vistas anteriormente que ofrezca servicios de alta disponibilidad a nivel de datos y de servicios. Durante el desarrollo de las actividades de laboratorio tendrá los privilegios de administración del usuario *root* de cada servidor.

Objetivos:

- Implementar los conocimientos adquiridos después de haber resuelto las prácticas de laboratorio para la creación de una pequeña nube.
- Configurar la aplicación ownCloud sobre la nube configurada y comprobar que tiene alta disponibilidad ante fallos del servicio.

Requerimientos:

Hardware	Software
Computadora con los siguientes requisitos: <ul style="list-style-type: none">• Procesador mínimo de velocidad de 2.1 GHz• Memoria RAM de 4 GB.	Software de virtualización VMware con siguientes elementos: <ul style="list-style-type: none">• PC Servidor 1 (lubuntu12.04)• PC Servidor 2 (lubuntu12.04)• PC Maquina 1 (lubuntu12.04)• PC Maquina 2 (lubuntu12.04)

Duración de la práctica

- Tres sesiones de laboratorio.

Topología

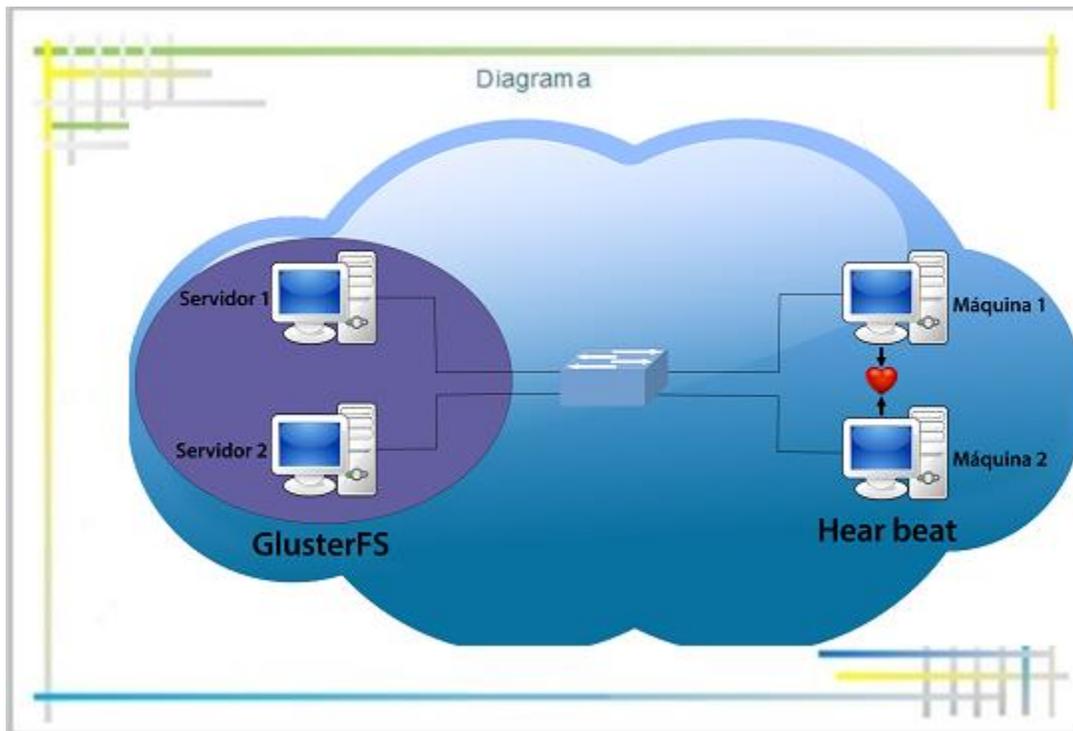


Figura 43 Topología a seguir para la práctica numero 8

Comandos de ayuda	
Comando	Descripción
<code># apt-get install heartbeat</code>	Instalación del software necesario.
<code># apt-get install glusterfs-server</code>	Instalación del software necesario.
<code># apt-get install glusterfs-client</code>	Instalación del software necesario.
<code># /etc/init.d/apache2 restart</code>	Reinicia el servicio de apache2.
<code># apt-get install apache2</code>	Instalar apache
<code># apt-get install mysql-server phpmyadimn</code>	Instalar mysql
<code># apt-get install php5</code>	Instalar php
<code># /etc/init.d/heartbeat restart</code>	Reinicia el servicio de heartbeat.

Enunciado:

Configurando:

1. En primera instancia hacer que los servidores y máquinas tengan conectividad entre ellos.
2. Instalar apache, mysql y php en las Máquina1 y Máquina2.

3. Instalar y configurar GlusterFS en los 2 servidores y las 2 máquinas y configurarlos para que compartan un directorio.
4. Instalar y configurar *Heartbeat* en las máquinas 1 y 2.
5. Instalar y configurar ownCloud en las máquinas 1 y 2.
6. Configurando Máquina 1 y 2
 - a) Descargar owncloud server desde la página oficial.
 - b) Descomprimir el archivo y mover la carpeta descomprimida a `/var/www/`
 - c) Cambiar los permisos a la carpeta y todo lo que se encuentra dentro de ella

```
# chmod -R 755 owncloud/  
# chown -R www-data:www-data owncloud/
```

- d) Crear la base de datos que la llamaremos owncloud. Acceder desde el navegador con el URL **`http://192.168.0.10/phpmyadmin`** y crearla.
- e) Acceder desde el navegador con el URL **`http://192.168.0.10/owncloud`**. Insertar los datos que nos pedirá cuando cargue interfaz gráfica de owncloud.
- f) Desmontar la carpeta compartida de gluster.
- g) Copiar los archivos que se encuentra en `/var/www/owncloud` a la carpeta compartida de gluster en nuestro caso es `/mnt/glusterfs`.

```
# cp -pR /var/www/owncloud/ /mnt/glusterfs/
```

- h) Borrar la carpeta owncloud que está en `/var/www/`. Montar la carpeta compartida de gluster ahora será `/var/www/owncloud/`.
- i) Acceder desde el navegador con el URL **`http://192.168.0.10/owncloud`**.
- j) Copiar la carpeta que contiene la bases de datos a `/mnt/glusterfs`

```
# cp -pR /var/lib/mysql/ /var/www/owncloud/
```

- k) Editar los archivos de configuración de mysql para que las base de datos que cargue sean las que estas en `/var/www/owncloud/`. Editar el archivo `/etc/mysql/my.cnf` deberá ser como lo siguiente

```
datadir = /var/www/owncloud/mysql
```

Comentar la línea donde se encuentra `chroot`. Editar el archivo `/etc/apparmor.d/usr.sbin.mysql` deberá ser como lo siguiente

```
GNU nano 2.2.6 Archivo: /etc/apparmor.d/usr.sbin.mysqld
/etc/mysql/*.pem r,
/etc/mysql/conf.d/ r,
/etc/mysql/conf.d/* r,
/etc/mysql/*.cnf r,
/usr/lib/mysql/plugin/ r,
/usr/lib/mysql/plugin/*.so* mr,
/usr/sbin/mysqld mr,
/usr/share/mysql/** r,
/var/log/mysql.log rw,
/var/log/mysql.err rw,
/var/www/owncloud/mysql/ r,
#/var/lib/mysql/ r,
/var/www/owncloud/mysql/** rwk,
#/var/lib/mysql/** rwk,
/var/log/mysql/ r,
/var/log/mysql/* rw,
/var/run/mysqld/mysqld.pid rw,
/var/run/mysqld/mysqld.sock w,
```

- l) Acceder desde el navegador con el URL **http://192.168.0.10/owncloud**. Registrarse.

Probar su funcionamiento

Desde una maquina cliente (cualquiera otra máquina) acceder desde su navegador a owncloud subir cualquier imagen. Luego suspender la maquina 2 y servidor 2. Intentar acceder desde el cliente a owncloud y el servicio continua funcionando.

5 Conclusiones

Al finalizar el trabajo monográfico, cumpliendo con la finalidad de los objetivos establecidos llegamos a las siguientes conclusiones:

1. El desarrollo de la teoría llevara al estudiante a una mejor comprensión de los temas abordados.
2. La secuencia de las prácticas permitirá que el estudiante alcance un alto nivel de aprendizaje lo que conllevará a obtener conocimientos concretos.
3. El diseño de las prácticas facilitará al estudiante a la asimilación de los conocimientos expuestos para la rápida solución de las mismas.
4. Las preguntas de control al final de las prácticas reforzaran aspectos puntuales sobre dicho tema.

Con estas conclusiones, podemos afirmar que hemos logrado desarrollar un documento sencillo y práctico que los estudiantes podrán usarlo para aprender los conceptos teóricos prácticos de Computación en la Nube y enfrentar desafíos laborales.

6 Recomendaciones

Las recomendaciones que se describen a continuación se basan en una futura actualización del documento.

1. Los temas que presenta este documento, se ha intentado que estén lo más actualizados hasta la presentación de este trabajo, considerados de suma importancia para el desarrollo de aprendizaje de los alumnos de la carrera de Ingeniería en Telemática, sin embargo, debe actualizarse continuamente con nuevos temas de importancia.
2. En lo que corresponde a la parte de desarrollo de las prácticas en este documento, es necesario siempre hacer un evaluado al final de cada una, de manera que el alumno pueda medir el grado de aprendizaje adquirido en la teoría así como en la práctica.
3. Se sugiere conservar la secuencia de las prácticas ya que están ordenadas de nivel básico a complejo.
4. A la medida de lo posible se sugiere adquirir equipos físicos para que el estudiante implemente los conocimientos adquiridos en un ambiente real.
5. Se sugiere elaborar un documento con contenido teórico y a su vez ejercicios para complementar los conocimientos que se adquiere con este documento.
6. Se sugiere que como un trabajo extra migrar los servicios usados en este documento hacia windows ya que es un sistema operativo muy usado en la actualidad, en el anexo 2 se proporciona información que les ayudará a realizarlo.

7 Bibliografía

2012, W. (Junio de 2012). *Virtualización. Conceptos básicos*. Obtenido de EsLaRed: www.eslared.org.ve/walc2012/material/track5/virtualizacion_conceptos_basicos.pdf

Containers, O. V. (30 de September de 2015). *OpenVZ*. Obtenido de OpenVZ Virtuozzo Containers: <http://download.openvz.org/doc/openvzintro.pdf>

DesdeLinux.net. (s.f.). *DesdeLinux*. Obtenido de DesdeLinux: <http://blog.desdelinux.net/>

EsLaRed.org.ve. (2012). *Alta disponibilidad y Virtualización*. Obtenido de EsLaRed: <http://www.eslared.org.ve/walc2012/material/track5/Intro-Alta%20disponibilidad%20y%20Virtualizaci%F3n%20con%20%20soluciones%20de%20bajo.pdf>

EsLaRed.org.ve. (2012). *Fundamentos de alta disponibilidad y balanceo de carga*. Obtenido de EsLaRed: <http://www.eslared.org.ve/walc2012/material/track5/Fundamentos%20de%20Alta%20disponibilidad%20y%20balanceo%20de%20carga.pdf>

EsLaRed.org.ve. (2012). *GlusterFS*. Obtenido de EsLaRed: <http://www.eslared.org.ve/walc2012/material/track5/GlusterFS.pdf>

Kezherashvili, B. (12 de 2011). *Computación en la Nube*. Obtenido de Universidad de Almería: http://www.adminso.es/recursos/Proyectos/PFM/2011_12/PFM_cloud_beka.pdf

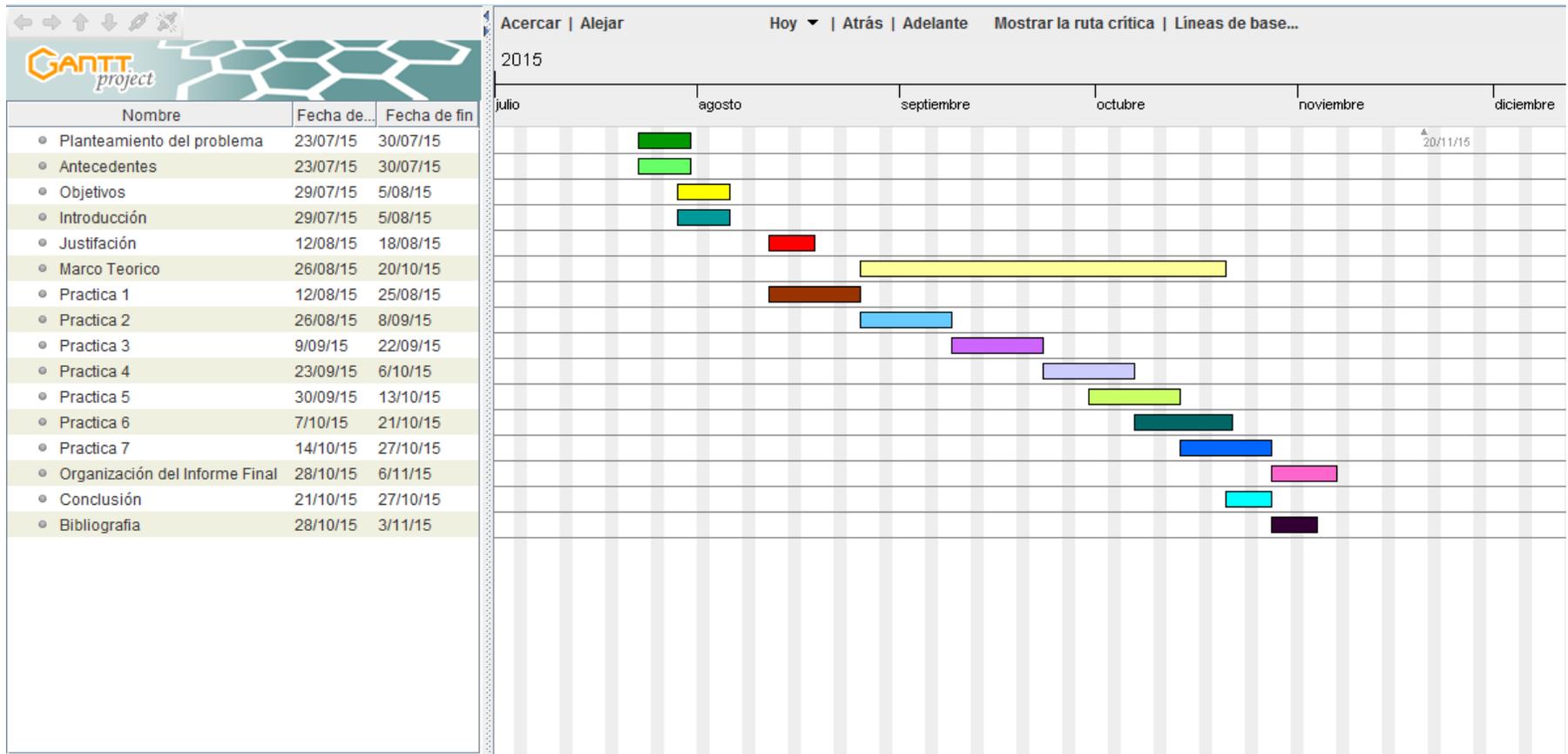
Kolyshkin, K. (1 de September de 2006). *Virtualization in Linux*. Obtenido de taifile.net: <http://mirror.nhanhoa.com/openvz/doc/openvz-intro.pdf>

Red Hat, I. (September de 2015). *GLUSTER*. Obtenido de GLUSTER: <http://www.gluster.org/>

VMware, I. (2015). *VMware*. Obtenido de VMware: <http://www.vmware.com/>

8 Anexos

Anexo 1. Cronograma de actividades



Anexo 2. Alternativas para Alta Disponibilidad y Balanceo de carga en Windows

El presente documento esta creado con la intención de proporcionare al docente una referencia de tecnologías la cual puede implementar para lograr la migración de los servicios expuestos en el documento monográfico hacia la plataforma de Windows.

Contenedores de Windows Server.

Ofrecen aislamiento de aplicaciones mediante tecnología de aislamiento de procesos y espacios de nombres. Un contenedor de Windows Server comparte el kernel con el host de contenedor y todos los contenedores que se ejecutan en el host.

Tecnología utilizada:

Docker: Implementa una API de alto nivel para proporcionar contenedores livianos que ejecutan procesos de manera aislada.

RAID.

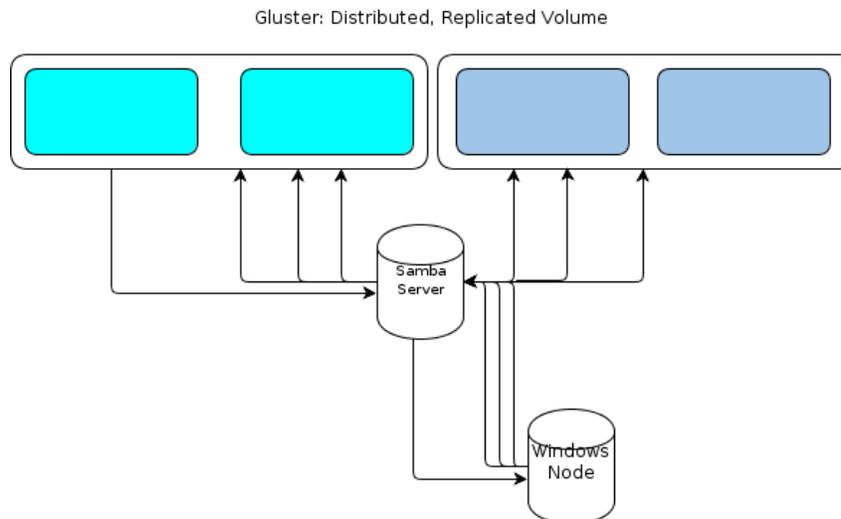
Windows ha construido en la funcionalidad para crear un software RAID (Redundant Array of Inexpensive Disks) sin necesidad de herramientas adicionales. Esto hace que sea fácil convertir sus actuales unidades de disco duro de repuesto en el almacenamiento masivo o incluso copias de seguridad redundantes.

En la siguiente página web encontraran una guía en la cual muestra su implementación:

<http://www.radians.com.ar/blog/?p=1396> (radians, s.f.)

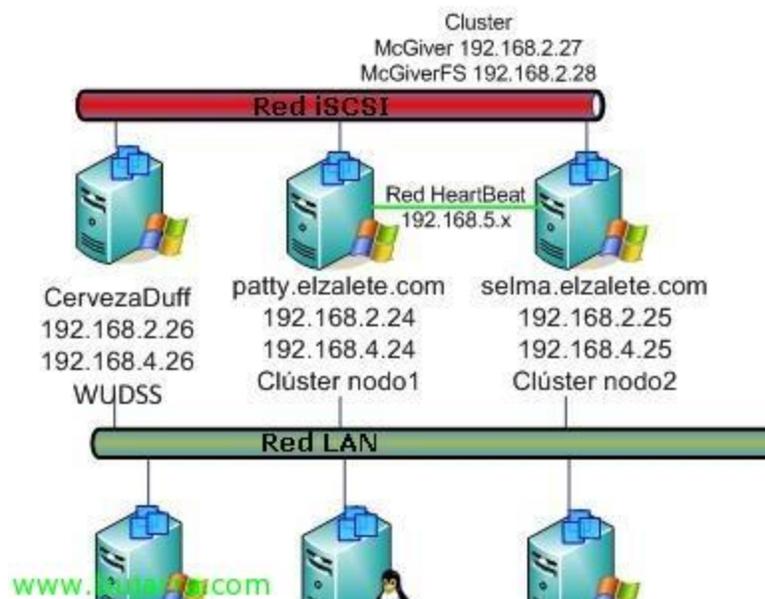
GlusterFS

Windows no tiene un nativo de glusterfs cliente necesitamos alguna manera para nuestro windows basado en el procesamiento de los nodos para interactuar con el almacén de archivos en una resistentes similares. La documentación de glusterfs indica que el camino para proporcionar acceso de Windows es configurar un servidor Samba.



Heartbeat.

Usando Windows server 2008, un clúster nuevo, y será para un servidor de ficheros, de tipo failover clustering, el proceso para otro tipo de clúster es parecido. Con esto conseguiremos alta disponibilidad, que si se nos cae uno de los servidores (nodo) no pase nada, ya que existen otros nodos que pueden coger los recursos y levantar los servicios.



En la siguiente página web encontrarán una guía en la cual muestra su implementación:

<http://www.tundra-it.com/creando-un-cluster-de-alta-disponibilidad-en-microsoft-windows-server-2008/>

LVS.

Equilibrio de Carga utilizando servidores Windows: Network Load Balancing (NLB) ofrece una solución de alta disponibilidad para aplicaciones de servidor basadas en TCP/IP, capaz de ofrecer escalabilidad y alto rendimiento.

La configuración de Network Load Balancing (NLB) sobre dos Máquinas Virtuales de Hyper-V (denominadas VRS01 y VRS02), ambas configuradas con una única tarjeta de red, utilizando el Modo de Operación Multicast.

En las siguientes páginas web encontrarán una guía en la cual muestra su implementación:

http://www.guillesql.es/Articulos/Instalar_Configurar_Network_Load_Balancing_NLB_Windows_Server_2008R2.aspx

http://www.guillesql.es/Articulos/Instalar_Configurar_Microsoft_Cluster_NLB_Windows_2003.aspx