

Universidad Nacional Autónoma de Nicaragua- León  
Facultad de Ciencias y Tecnología  
Departamento de Computación



Provisión de servicios de blogging escalable basado en virtualización con QEMU.

Tesis para optar al título de Ingeniería en Sistemas de Información

**Elaborado por:**

Br. Helen Anielka Martínez Quezada.

Br. Keving Jose Medina.

Br. Nuria de los Ángeles Velásquez Betanco.

**Tutores:**

MSc. Rina Arauz.

Dr. Oscar García Población.

León, Nicaragua 2017.

“A la Libertad por la Universidad”

## DEDICATORIA

Dedico este trabajo Principalmente:

**A: Dios y a la virgen** Por darme salud y fortaleza para realizar mis estudios.

**A: Mi Abuela Juana Medina y Mi Madre María Medina** por todo su apoyo incondicional y por hacerme un hombre de bien, sin ellas esto no lo hubiese logrado y a toda mi familia que de diferentes maneras recibí su apoyo.

**A: Nuestros tutores** MSc. Rina Arauz y Dr. Oscar García Población, por ayudarnos a realizar esta tesis para obtener el título.

**A: Mis amigos** por su apoyo en el transcurso de la universidad y a mis dos amigas y compañeras con las cual definiendo hoy esta tesis.

***Br. Keving José Medina.***

## DEDICATORIA

Con mucho amor dedico este trabajo:

**A: Dios**, nuestro padre creador de todas las cosas dador de sabiduría, porque me ha dado la fuerza necesaria para seguir adelante, y permitirme llegar a este momento tan especial en mi vida.

**A: Mi familia** quien siempre han estado dispuesto a apoyarme moral y económicamente, quienes siempre me han dado la motivación para incentivar me en mis estudios.

**A: Nuestros tutores** Msc. Rina Arauz y Dr. Oscar García Población, quienes nos han brindado de su tiempo, comprensión y espacio para atendernos.

En general a todas las personas que de una u otra manera han contribuido para llevar a cabo este proyecto.

***Br. Nuria de los Ángeles Velásquez Betanco.***

## **DEDICATORIA**

Dedico este trabajo principalmente a Dios, por haberme dado la vida y permitirme el haber llegado hasta este momento tan importante de mi formación profesional. A mi madre y padre, por ser los pilares más importantes en mi vida y por demostrarme siempre su cariño y apoyo incondicional pues ellos han marcado mi camino motivándome día a día ser mejor persona tanto emocional como profesionalmente. A mis compañeros Nuria y Keving porque si el equipo que formamos, no hubiéramos logrado esta meta.

***Br. Hellen Anielka Martinez Quezada.***

## **AGRADECIMIENTOS**

Agradecidos con Dios y la Virgen por darnos salud, sabiduría, perseverancia, inteligencia y deseo de superación para realizar nuestros estudios.

A nuestras familias por todo su apoyo incondicional por hacernos personas de bien y a todas aquellas personas que nos apoyaron en la realización de nuestro proyecto.

A la UNAN–León, como institución y en especial a los maestros que nos brindaron herramientas, técnicas, estrategias y conocimientos esenciales para culminar nuestros estudios universitarios y poder realizar este trabajo y en especial a nuestros tutores: Msc. Rina Arauz y Dr. Oscar García Población por brindarnos su apoyo y guiarnos para poder concluir nuestro proyecto.

**Los autores.**

## INDICE

<b>I. INTRODUCCION</b> .....	12
<b>II. PLANTEAMIENTO DEL PROBLEMA</b> .....	13
<b>III. JUSTIFICACIÓN</b> .....	14
<b>IV. OBJETIVOS</b> .....	15
4.1 Objetivo General .....	15
4.2 Objetivos Específicos.....	15
<b>V. ANTECEDENTES</b> .....	16
5.1 Motivación del proyecto .....	18
5.2 Bases teóricas.....	18
5.3 Teoría web de Anggy Gutierrez .....	19
5.5 Hosting o alojamiento web .....	19
5.6 Virtualización de Servidores.....	20
5.7 Bases legales.....	20
5.8 Definición de términos básicos.....	21
5.8.1 ¿Qué es QEMU.....	21
5.8.2 Virtualización.....	22
5.8.3 Hosting .....	22
5.8.4 Blogging .....	23
5.8.5 Software de blogging wordpress .....	23
5.8.6 Balanceo de carga .....	23
5.8.7 Balanceo de carga .....	24
<b>VI. MATERIALES SOFTWARE Y HARDWARE UTILIZADOS</b> .....	25
<b>VII. Provisión de servicios de blogging escalable basado en virtualización con QEMU.</b> .....	26
7.1 Instalar y configurar QEMU en una máquina de desarrollo.....	29
7.1.1 Emulación del modo usuario .....	29
7.1.2 Modo de emulación completo de sistema de ordenador .....	30
7.1.3 Instalación y configuración de QEMU. ....	30
7.1.4 Línea de código para verificar extensiones de virtualización hardware ....	30
7.2 Crear una máquina virtual llamada DB1. ....	31
7.2.1 Crearemos la máquina virtual .....	32

7.2.2 Instalacion de MYSQL en la máquina virtual llamada DB1. ....	33
7.3 Crear una máquina virtual llamada WP1.0.....	34
7.4 Instalación de WORDPRESS en la máquina virtual llamada WP1.0 .....	35
7.5 Configurar Wordpress para que haga uso de DB1 como servidor de base de datos. ....	42
7.5.1 Entradas.....	42
7.5.2 Comentarios.....	43
7.5.3 Páginas .....	43
7.5.4 Paginas estáticas .....	43
7.6 Configuración de Wordpress y el servidor de base de datos .....	43
7.7 Hacer que la maquina sea accesible desde una herramienta de administración remota vía web.....	47
7.7.1 ¿Qué es Webmin? .....	47
7.8 Crear dos máquinas virtuales.....	52
7.9 Instalar wordpress en las maquinas WP1.1 Y WP1.2 para que hagan uso de DB1 como servidor de base de datos. ....	53
7.10 Ahora hay tres clones del mismo blog lo siguiente es conseguir que se reparta el tráfico entre las tres.....	59
7.11 Instalar un balanceador de carga que reparta las peticiones entre las tres máquinas WP1.0, WP1.1, Y WP1.2. ....	60
<b>VIII. CONCLUSIONES</b> .....	<b>65</b>
<b>IX. REFERENCIAS</b> .....	<b>66</b>

## INDICE DE FIGURAS

<i>Figura (1).</i> Balanceador de carga.....	24
<i>Figura (2).</i> Proveer blogging en máquinas virtuales a clientes.....	26
<i>Figura (3).</i> Utilizando virtualización con QEMU para proveer servicios de blogging. ....	27
<i>Figura (4).</i> Código de verificación de extensiones de virtualización hardware ..	30
<i>Figura (5).</i> Imagen de la interfaz gráfica para administrar las máquinas virtuales .....	31
<i>Figura (6).</i> Línea de código para acceder al directorio libvirt .....	32
<i>Figura (7).</i> Creación del usuario y definición de la contraseña .....	34
<i>Figura (8).</i> Creación del usuario Wordp definiendo el host y contraseña.....	34
<i>Figura (9).</i> Autorización del servidor de base de datos .....	34
<i>Figura (10).</i> Línea de código para crear una máquina virtual llamada wp1.0.....	35
<i>Figura (11).</i> Contenido del archivo del sitio wordpress .....	37
<i>Figura (12).</i> Deshabilitar el sitio 000-default.....	38
<i>Figura (13).</i> Habilitación del sitio wordpress .....	38
<i>Figura (14).</i> Ajuste php adaptado a las necesidades de wordpress.....	39
<i>Figura (15).</i> Acceso a mysql para creación de usuario y base de datos para wordpress .....	40
<i>Figura (16).</i> Acceso a mysql mediante un password.....	40
<i>Figura (17).</i> Creación de la base de datos wordpress.....	40
<i>Figura (18).</i> Creación de un usuario wordpress en mysql.....	41
<i>Figura (19).</i> Descarga de worpress desde wget.....	41
<i>Figura (20).</i> Cambios de permisos para evitar futuros conflictos .....	41
<i>Figura (22).</i> Configuración de wordpress desde el navegador web .....	42

<i>Figura (23).</i> Creación del usuario en MYSQL en la maquina wp1.0 .....	44
<i>Figura (24).</i> Configuración del servidor MYSQL.....	44
<i>Figura (25).</i> Reiniciar el servidor MYSQL.....	44
<i>Figura (25).</i> Acceder al servidor MYSQL mediante un password.....	44
<i>Figura (26).</i> Línea de código para conectarse remotamente a la base de datos	45
<i>Figura (27).</i> Abrir puerto 3306 en el firewall para permitir acceso al host de wordpress .....	45
<i>Figura (28).</i> Verificación del acceso remoto al servidor MYSQL desde el wp1.0	45
<i>Figura (29).</i> Configuración de wordpress para que haga uso de Word-p.....	46
<i>Figura (30).</i> Configuración final de nuestro sitio en wordpress .....	47
<i>Figura (31).</i> Panel de acceso a nuestro sitio .....	47
<i>Figura (32).</i> Descarga del programa desde la página oficial de webmin.....	48
<i>Figura (33).</i> Confirmación de la excepción de seguridad .....	49
<i>Figura (34).</i> Acceso al webmin.....	49
<i>Figura (35).</i> Ventana del acceso del webmin .....	50
<i>Figura (36).</i> Ventana de acceso desde otro host remoto para acceder y administrar nuestra máquina virtual wp1.0.....	50
<i>Figura (37).</i> Ventana del acceso de la maquina wp1.0 .....	51
<i>Figura (38).</i> Configuración de la consola de la máquina virtual wp1.0 .....	51
<i>Figura (39).</i> Panel de administración de la máquina virtual .....	52
<i>Figura (40).</i> Creación de la máquina virtual wp1.1 .....	52
<i>Figura (41).</i> Creación de la máquina virtual wp1.2.....	53
<i>Figura (42).</i> Creación de archivos para la maquina wp1,1 .....	54
<i>Figura (43).</i> Creación de archivos para la maquina wp1.2.....	54
<i>Figura (44).</i> Descarga de wordpress desde wget.....	56
<i>Figura (45).</i> Descompresión de wordpress y cambios en los permisos .....	56

<i>Figura (46).</i> Verificación del acceso remoto al servidor MYSQL desde wp1.2...	57
<i>Figura (47).</i> Configuración de wordpress en wp1.....	58
<i>Figura (48).</i> Configuración de wordpress en wp1.2.....	58
<i>Figura (49).</i> Línea de instalación de squid .....	59
<i>Figura (50).</i> Configuración de squid.....	59
<i>Figura (50).</i> Ajuste de las máquinas según reglas establecidas por squid.....	60
<i>Figura (51).</i> Preparación del archivo /etc/host para colocación del ip de las máquinas .....	61
<i>Figura (52).</i> pasar la variable de sesión a los nodos.....	62
<i>Figura (52).</i> Guardar y reiniciar apache .....	62
<i>Figura (53).</i> Configuración de los nodos para que reciban la variable de sesión en wp1.1 .....	63
<i>Figura (54).</i> Guardado de los datos y reinicio de apache en wp1.1 .....	63
<i>Figura (55).</i> Configuración de los nodos para que reciban la variable de sesión en wp1.2 .....	63
<i>Figura (56).</i> Guardado de los datos y reinicio de apache en wp1.2 .....	64





## I. INTRODUCCION

En la actualidad no se podría imaginar nuestro país sin el avance de la tecnología ya que esta ha unificado de una forma más eficaz la interacción entre las personas siendo este un elemento que ha facilitado nuestras vidas, haciendo más productivas a las industrias y con mayor beneficio a la sociedad.

En nuestro país mediante la civilización, el crecimiento de las empresas y la abundante información que estas proporcionan se da la necesidad de recurrir a un medio tecnológico más competente, útil para la administración, el alojamiento, seguridad, propagación de su información y comunicación con sus clientes una de las formas más factibles es a través de sitios web haciendo uso de un hosting ya que es un sistema esencial para el funcionamiento de las páginas web en Internet.

Nuestro trabajo consiste en la creación de una empresa que proporciona wordpress a otros clientes, y para reducir costes vamos a asignarles máquinas virtuales en lugar de máquinas físicas y también les asignaremos un servidor de base de datos común para todas ellas.



## II. PLANTEAMIENTO DEL PROBLEMA

En un mundo en pleno desarrollo, la tecnología ha generado ansias de crecer cada día más. El ser humano ha presentado un cambio drástico en su calidad de vida, incrementando sus conocimientos y adaptándolos para su propia utilidad.

La tecnología se presenta ahora como herramienta que modifica y diversifica el desarrollo y desempeño del ser humano.

En la actualidad el ser humano, no sólo emplea la tecnología para su diario vivir, sino que también hace uso de ella para crecer en su ámbito laboral. Hay empresas que fundamentalmente no se dediquen a cuestiones tecnológicas, pero con el fin de lograr una mayor comunicación, ofrecer un buen servicio e interactuar con sus clientes tiene presencia en internet a través de páginas web y blogs para así también poder llegar al público y hacerse conocer.

En nuestro país escasamente existen empresas que brinden servicios de host o alojamiento web a usuarios para que además de gozar de diferentes servicios puedan tener de una manera segura su información. A partir de ahí nace nuestra pregunta general de investigación:

¿Es necesaria la provisión de servicios de hosting para blogs a través de máquinas virtuales para que los usuarios administren servicios que favorezcan la relación con sus clientes además de brindarles una mayor seguridad?



### **III. JUSTIFICACIÓN**

Con el avance de la tecnología, la amplia información y la carencia de empresas que brinden servicios de hosting o alojamiento web en nuestro país, se da la necesidad de implementar servicios para el funcionamiento de sitios web en internet.

A partir de esta problemática se piensa establecer un método que facilite el manejo, alojamiento y seguridad de la información a través de la provisión de servicios de hosting para blogs o sitios web a usuarios finales que no poseen recursos necesarios y espacios de almacenamiento para albergar dicha información.

La creación de este hosting a través de máquinas virtuales permitirá que los usuarios administren sistemas o servicios que facilite la comunicación con sus clientes.



## **IV. OBJETIVOS**

### **4.1 Objetivo General**

Proveer servicios de hosting para blogs, escalable y con balanceo de carga.

### **4.2 Objetivos Específicos**

- Proporcionar un software de blogging totalmente operativo que el sistema ofrecerá a los usuarios finales.
- Instalar y configurar el entorno QEMU operativo para la creación de máquinas virtuales.
- Implementar seguridad y tolerancia a fallos a los servicios que se ofrecerán.



## V. ANTECEDENTES

El objetivo de la recopilación es brindar una aproximación a los temas centrales de nuestro proyecto con el fin de encontrar la existencia de algunas líneas de investigación similares.

Un primer trabajo corresponde a Facultad de Matemáticas e informática, Universidad de Barcelona (2010), quien realizó un Análisis de la virtualización de sistemas operativos en este proyecto se refiere básicamente a los administradores de sistemas informáticos debido a que ellos deben estar al día sobre los cambios en tecnologías Software/Hardware y en cómo se puede mejorar la productividad en las empresas a la vez que se facilita la realización del trabajo. Una de estas tecnologías es la virtualización de sistemas operativos, que consiste básicamente en poder compartir en un mismo Hardware varios sistemas operativos funcionando de forma totalmente independiente. Esto se realiza mediante un Software de virtualización. Dado que en el mercado informático existen varias alternativas pero que nadie ha realizado un análisis a nivel de usuario de cuál de ellas se ajusta mejor a un determinado perfil, en este proyecto se ha realizado un análisis detallado del funcionamiento de esta tecnología. Algunos de los puntos a tratar son las ventajas e inconvenientes de usar esta tecnología, análisis de los mejores virtualizadores del momento para usuarios y empresas con comparativas de funcionalidades, pruebas y test de rendimiento con programas reales en sistemas operativos Windows y Linux, información del proceso del virtualizador a bajo nivel en Linux, etc. La tecnología actual de Hardware ha hecho evolucionar mucho a estos programas desde su aparición y podremos ver en los resultados como la potencia de un sistema operativo virtualizado puede ser igual o incluso superior a la de un sistema operativo real.

Un segundo proyecto corresponde a Universidad de Almería (2010) el cual se refiere a Virtualización de servidores de telefonía IP en GNU/Linux, el principal objetivo de este proyecto es que la virtualización de servicios de telefonía IP (consolidación de servidores centralita VoIP). La telefonía IP (Voice over Internet Protocol), nacida de la necesidad y la utilidad de hacer uso de una sola red para la transmisión de datos en lo que a telefonía se refiere, es la comunicación que viaja



a través de la red permitiendo un ahorro elevado tanto en costes de comunicación como en costes de mantenimiento de las infraestructuras de telefonía, teniendo al mismo tiempo grandes posibilidades de escalabilidad, portabilidad, e integración con las redes locales internas y recursos de las empresas.

La virtualización es sin duda alguna la tecnología con más calado y más demandadas actualmente en la sociedad que está revolucionando el sector de las tecnologías de la información porque está transformando su modelo de trabajo prácticamente a todos los niveles. Virtualizar, en una frase, es lograr la ejecución de máquinas –con sistemas operativos, aplicaciones y servicios propios- dentro de otras máquinas. Es aquí donde yace la diferencia con otras técnicas conocidas también con el sobrenombre de virtualización, aunque bien distintas; como por ejemplo el Grid Computing o Server Agregation (Agregación de Servidores), que a grandes rasgos consiste en lograr la percepción de múltiples servidores como uno sólo.

Qemu (<http://www.qemu.org/>) es sin duda una de las soluciones de virtualización más interesantes en la actualidad. Y es que, aunque haya sido incluido en el presente apartado como un emulador, cuyos motivos serán explicados a continuación, permite también virtualización completa de equipos. No en el sentido de virtualizar una infraestructura completa de servidores, ya que no proporciona funcionalidades complejas como puede tratarse de la migración de máquinas virtuales, y sí en el de satisfacer las condiciones necesarias para la ejecución de máquinas virtuales diversas en arquitectura hardware, sistemas operativos, aplicaciones... por lo que en la mayoría de los casos en los que queremos virtualizar de una manera más modesta Qemu será la solución más adecuada, ya que no será necesario llevar a cabo complejas configuraciones –como por ejemplo recompilar el kernel de Linux.

La diferencia principal entre Qemu y otras soluciones es que funciona como un emulador de procesadores, por lo que no ejecuta el código generado por las máquinas virtuales de forma nativa, sino que lo interpreta. La importancia de la emulación de procesadores que lleva a cabo Qemu, incluso usada por otras soluciones de virtualización de plataforma como parte de su proceso de



virtualización (como es el caso de KVM o Kernelbased Virtual Machine), es el motivo principal el cual ha sido incluido como un emulador, aunque volvemos a recalcar que permite implementar también virtualización completa. Esta decisión tan importante en el diseño y funcionamiento interno conlleva la aparición de diferentes ventajas e inconvenientes directamente derivados de ello.

Un tercer proyecto corresponde a Departamento de Ingeniería Telemática Escuela Superior de Ingenieros Universidad de Sevilla (2016) la idea principal de este proyecto nació de crear una pequeña empresa llamada Informática Soluciones S.L. (informática – telecomunicaciones – soluciones) recién creada en 2015, con base en el municipio de Aracena, en la Sierra de Aracena y Picos de Aroche (Huelva). Esta empresa fue formada por un equipo de jóvenes multidisciplinares con el objetivo de atender las necesidades en materia de informática y comunicaciones (y tecnológicas en general) en entorno de la sierra, poniendo el foco principalmente en profesionales y empresas.

### **5.1 Motivación del proyecto**

En este documento se pretende diseñar un sistema de alta fiabilidad, flexible, escalable y de bajo coste que pueda servir como base para una infraestructura de telecomunicaciones de garantías a cualquier organización. A su vez, se estudiarán y experimentará con técnicas de virtualización. Esta tecnología es en gran medida la madre de la famosa “Nube”, será la base de nuestro sistema y es un campo puntero en los avances informáticos de la actualidad.

En este proyecto analizaremos las distintas técnicas y plataformas de virtualización disponibles, desarrollaremos un entorno de trabajo sobre Proxmox VE que nos permita alcanzar los puntos anteriores y se crearán algunos servidores virtuales de ejemplo. marco teórico

### **5.2 Bases teóricas**

A continuación, mostramos las bases teóricas que sustentan la investigación sobre servicios de blogging escalables basados en virtualización con Qemu.

Aquí relacionamos varias teorías que le dan forma y se vincula con el proyecto planteado. Sobre este particular, Arias (1999), afirma que las bases



teóricas Comprenden un conjunto de conceptos y proposiciones que constituyen un punto de vista o enfoque determinado.

Sabiendo esto, las siguientes teorías se consideran adecuadas para cimentar tal instrumentación y sustentar la línea de investigación:

### **5.3 Teoría web de Anggy Gutierrez**

Es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP (Transmission Control Protocol), basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

### **5.4 URL**

Localizador de recursos uniforme sirve para nombrar recursos en Internet. Este nombre tiene un formato estándar y tiene como propósito asignar una dirección única a cada uno de los recursos disponibles en Internet, como por ejemplo textos, imágenes, vídeos, etc.

### **5.5 Hosting o alojamiento web**

La mayoría de los servicios de hosting ofrecen conexiones que para una persona individual resultarían muy costosas. Usar un servicio de hosting permite que muchas compañías compartan el coste de una conexión rápida a Internet para el acceso a los archivos de sus sitios web.

Algunas compañías de hosting describen sus servicios como hosting virtual. Hosting virtual generalmente implica que sus servicios serán transparentes y que cada sitio web tendrá su propio alojamiento de dominio y sus propias direcciones de email. En la mayoría de los casos, el hosting y el hosting virtual son sinónimos. Algunas compañías de hosting permiten a sus usuarios tener su propio servidor virtual, con la apariencia de que el usuario está controlando un servidor dedicado enteramente a su sitio web.



## **5.6 Virtualización de Servidores**

La virtualización de servidores es una tecnología software que a partir de un servidor físico permite dar la apariencia de que se divide en múltiples servidores (virtuales) sobre los que se pueden instalar diferentes sistemas operativos sin que interfieran entre sí.

A partir del uso de los sistemas virtualizados, varios sistemas operativos (pueden ser iguales o diferentes) se pueden ejecutar simultáneamente en un mismo ordenador sin que ninguno de ellos interfiera con los demás.

## **5.7 Bases legales**

En esta sección damos a conocer las principales leyes, reglamentos y decretos que definen las políticas y estrategias para las actividades científicas, tecnológicas y de innovación en nuestro país Nicaragua.

La asamblea nacional de la Republica de Nicaragua en uso de sus facultades: ha dictado la siguiente: ley general de ciencia, tecnología e innovación.

Arto. 1.- Objeto. La presente ley tiene por objeto organizar la política de Ciencia, Tecnología e Innovación, estrategias, implementación de mecanismos institucionales y operativos para la promoción, estímulos y fomento, definir los lineamientos que orientaran el Sistema Nacional de Ciencia, Tecnología e Innovación, la investigación, la apropiación social del conocimientos y la transferencia e innovación tecnológica, a fin de fomentar capacidad para la generación, uso y circulación del conocimiento e impulsar el desarrollo nacional.

Arto. 2.- Ámbito de aplicación. La presente ley es aplicable en todo el territorio nacional para las personas naturales y jurídicas, nacionales y extranjeras que realizan actividades de investigación científica, transferencia y desarrollo tecnológicos y de innovación.

Arto. 3.- Órgano competente. El Consejo Nicaragüense de Ciencia, Tecnología e Innovación, a través de su Secretaría Ejecutiva; es el órgano competente para la aplicación de la presente ley, reglamento; así como coadyugará en razón de su competencia en los procesos para elaborar, aprobar, aplicar, promover, supervisar y evaluar las normas técnicas de la misma, formular políticas, planes, programas, proyectos, manuales e instructivos para su aplicación.



Arto. 4.- Definiciones. Para los efectos de la presente ley se entenderá por:

a. Talento Humano: Es aquel capital que produce o es capaz de producir mediante la operación diaria o la innovación, los más altos beneficios tanto para la persona como para la compañía a la cual ofrece sus servicios. b. CNE: Consejo Nacional de Educación; c. Empresas de Base Tecnológica: Es toda empresa que en sus procesos o actividades utiliza o desarrolla herramientas, y tienen como fin explotar económicamente nuevos productos, servicios a partir de resultados de la investigación científica y tecnológica. d. Fondos: es todo recurso económico utilizado para impulsar el desarrollo de una actividad específica. e. Innovación: Aplicación de nuevos procesos o perfeccionamiento de los ya existentes mediante los cuales se mejoran la calidad de la producción, servicios, educación o administración de una organización, actividad general o específica. f. CONICYT: Consejo Nicaragüense de Ciencia, Tecnología e Innovación; g. SINICYT: Sistema Nacional de Información Científica, Tecnológica e Innovación. h. SECONICYT: Secretaria Ejecutiva del Consejo Nicaragüense de Ciencia, Tecnología e Innovación; i. SINACYT: Sistema Nacional de Ciencia, Tecnología e Innovación; j. Herramientas Tecnológicas: Medios o recursos que permiten generar, acceder, almacenar y transmitir bienes y servicios, que utilizan para su funcionamiento y composición las tecnologías disponibles o sus adaptaciones. k. FONICYT: Fondo Nicaragüense para el desarrollo de la Ciencia, Tecnología e Innovación. l. Descentralizado: es una forma administrativa en el cual se confiere, a través de una ley a un órgano, autonomía técnica y administrativa, se le otorga patrimonio propio y personalidad jurídica, existiendo control y tutela del Presidente de la República o del ministerio al que esté vinculado.

## **5.8 Definición de términos básicos**

### **5.8.1 ¿Qué es QEMU?**

QEMU es un emulador genérico y de código abierto de máquinas virtuales. Cuando se utiliza como un emulador de máquina, QEMU puede correr sistemas operativos y programas hechos para una máquina en particular (por ej. una placa ARM) en una máquina diferente (e.j. tu PC x86). Usando la traducción dinámica, se consigue un rendimiento muy bueno.



QEMU puede usar hipervisores como **Xen** o **KVM** para utilizar las extensiones del procesador para la virtualización. Cuando se utiliza como virtualizador, QEMU alcanza un performance cercano a el rendimiento nativo ejecutando el código de invitado directamente en el CPU host.

### 5.8.2 Virtualización

La virtualización consiste en la creación de un ámbito virtual, en una máquina física, donde de manera independiente podemos ejecutar programas sobre otro sistema operativo. La virtualización es un programa que suministra soporte para instalar otros sistemas operativos en la misma máquina, y que funcionarán de forma encapsulada dentro de ese ámbito virtual.

Habrà un sistema físico que será la máquina real y un sistema virtual o lógico que se ejecutará virtualizado en el sistema físico. Si tenemos suficientes recursos entonces bien podríamos tener varios sistemas virtuales ejecutándose sobre una única maquina real. Por otra parte también, se puede tener en sentido contrario, varios recursos físicos (servidores o dispositivos de almacenamiento) aparecen como un único recurso virtual.

### 5.8.3 Hosting.

El **alojamiento web** (en inglés *web hosting*) es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía web. Es una analogía de “hospedaje o alojamiento en hoteles o habitaciones” donde uno ocupa un lugar específico, en este caso la analogía alojamiento web o alojamiento de páginas web, se refiere al lugar que ocupa una página web, sitio web, sistema, correo electrónico, archivos etc. en internet o más específicamente en un servidor que por lo general hospeda varias aplicaciones o páginas web.

Las compañías que proporcionan espacio de un servidor a sus clientes se suelen denominar con el término en inglés *web host* ó *Hosting*.

Se puede definir como “un lugar para tu página web o correos electrónicos”, aunque esta definición simplifica de manera conceptual el hecho de que el



alojamiento web es en realidad espacio en Internet para prácticamente cualquier tipo de información, sea archivos, sistemas, correos electrónicos, videos etc. Para que un HOSTING pueda utilizar las funciones de acceso web por navegador y e-mails debe tener asociado a un dominio.

#### **5.8.4 Blogging**

La herramienta de inbound marketing blogging nos colabora entre otras cosas a propagar información de interés para nuestros usuarios así también para los potenciales. Entre las maneras que existen de enriquecer un blog, el Guest Blogging es una de las más extendidas.

Proviene del término "web log". Blogging es una manera interactiva de publicación de contenidos en la web. La acción de blogs se remonta a finales del año 1990, y se ha convertido en una forma autoritaria de la auto-publicación en la década de 2000.

#### **5.8.5 Software de blogging wordpress**

Wordpress o CMS (por sus siglas en inglés, Content Management System) es un sistema de gestión de contenidos ajustado a la realización de cualquier tipo de sitio, tomando en cuenta que ha alcanzado una gran relevancia usado para la creación de blogs o páginas web con una estructura cronológica que se actualiza regularmente. Se desarrolló en el lenguaje PHP para entornos que ejecuten **MYSQL Y APACHE**.

#### **5.8.6 Balanceo de carga**

El balanceo de carga permite distribuir las tareas que tendría que sustentar una única máquina, con el propósito de maximizar la amplitud de proceso de información y de ejecución de tareas. Esto permite que ningún equipo sea parte fundamental del servicio que ofreceremos. De esta manera evitaremos sufrir una detención de dicho servicio debido a una parada de uno de los equipos. En caso que un enlace o ruta falle, los procesos confirman que los mensajes pueden redirigirse de forma inmediata a otro enlace que es transparente para los usuarios.



### 5.8.7 Balanceo de carga

Una maquina cliente, conectada a un balanceador que distribuye las tareas entre dos equipos que hacen función de servidor para evitar fallos de servicios.

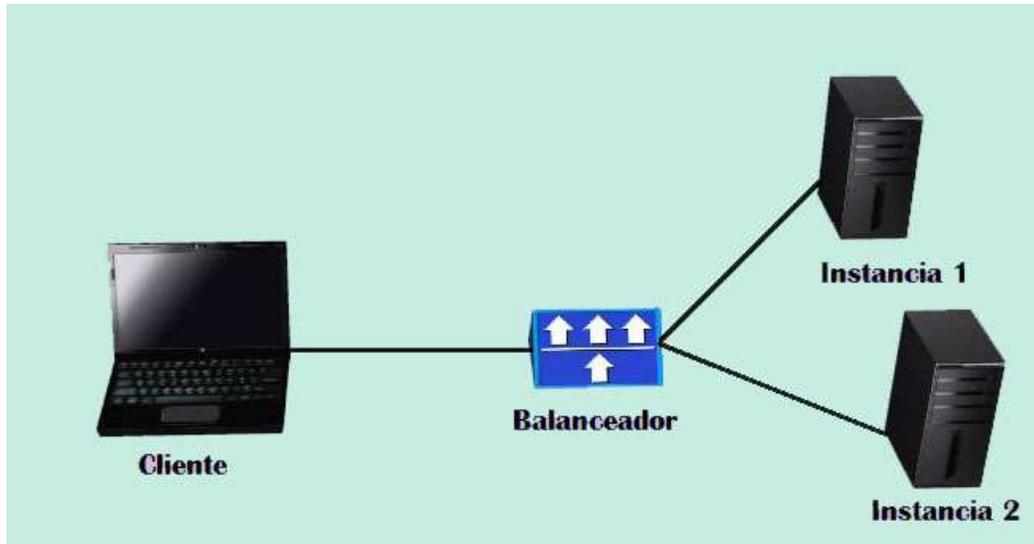


Figura (1). Balanceador de carga



## **VI. MATERIALES SOFTWARE Y HARDWARE UTILIZADOS.**

Para la realización de este proyecto se utilizaron los siguientes materiales Software y hardware.

### **6.1 Materiales Hardware**

- Una laptop Dell: Procesador Intel(R) Core(TM) i3-2350 CPU @ 2.30GHz 2.30 GHz.
- Memoria (RAM): 4,00 GB.
- Un router.
- Un cable de red con terminal RJ45.

### **6.2 Materiales Software**

- Sistema Operativo Debian GNU/Linux 7.4
- Servidor de Virtualización qemu-kvm 1:2.1
- Servidor de base de datos MySQL- 5.5.50
- Servidor web Apache2
- Software de blogging wordpress-4.5.3
- Herramienta de administración remota vía web, webmin\_1.810
- Servidor proxy para web con cache: squid3



## VII. Provisión de servicios de blogging escalable basado en virtualización con QEMU.

Creamos una empresa que brinda servicios de blogging a sus clientes, pero para reducir costes vamos a asignarles máquinas virtuales a cada uno de ellos en lugar de máquinas físicas, también les asignaremos un servidor de base de datos común para todas ellas.

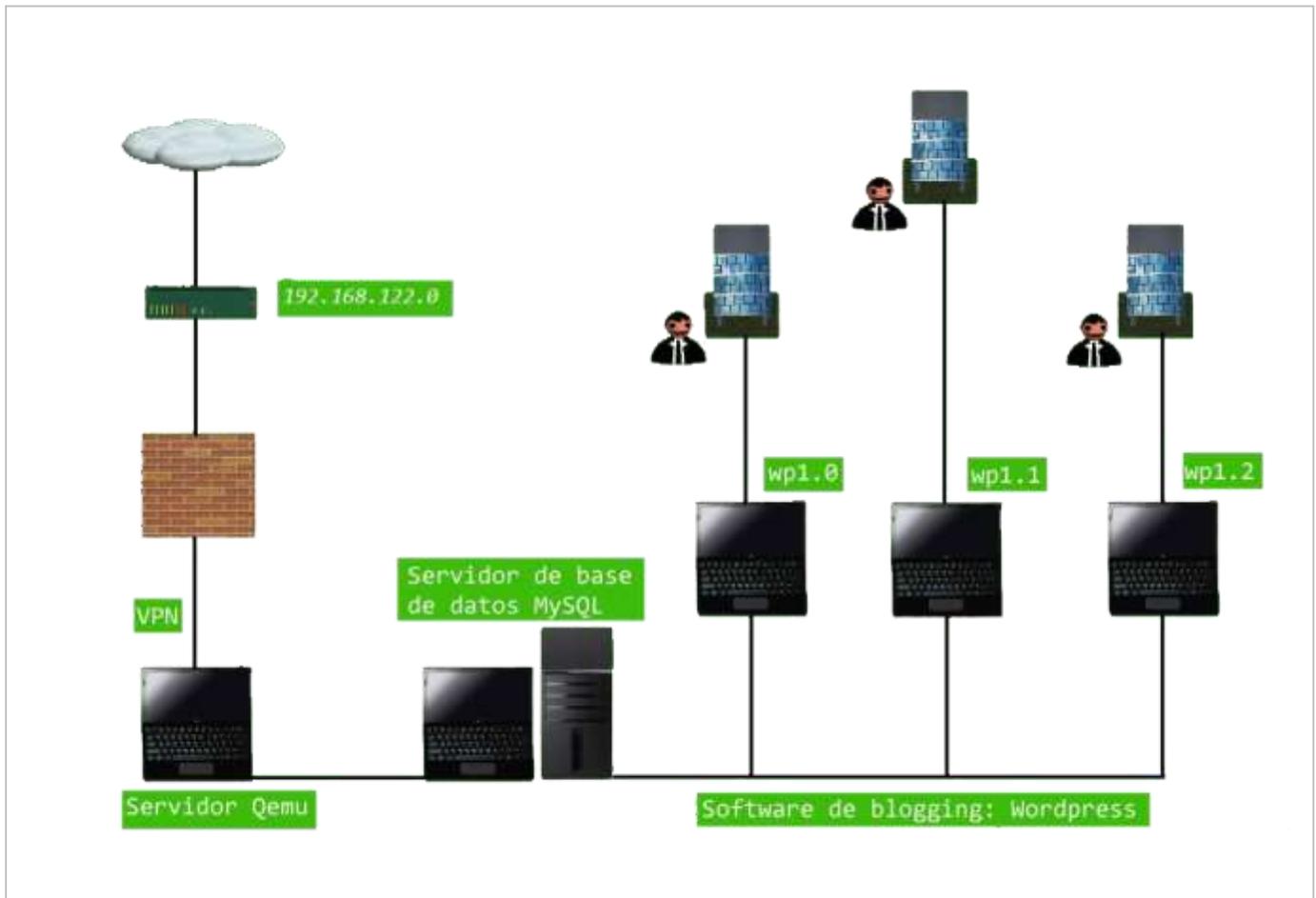


Figura (2). Proveer blogging en máquinas virtuales a clientes.



En el diagrama anterior se muestra a cada máquina virtual con wordpress brindando servicios a sus respectivas empresas, todas ellas tienen un servidor de base de datos común, y todo esto está soportado por el servidor QEMU.

En el siguiente diagrama se muestra los elementos que contienen cada una de las máquinas virtuales antes mencionadas.

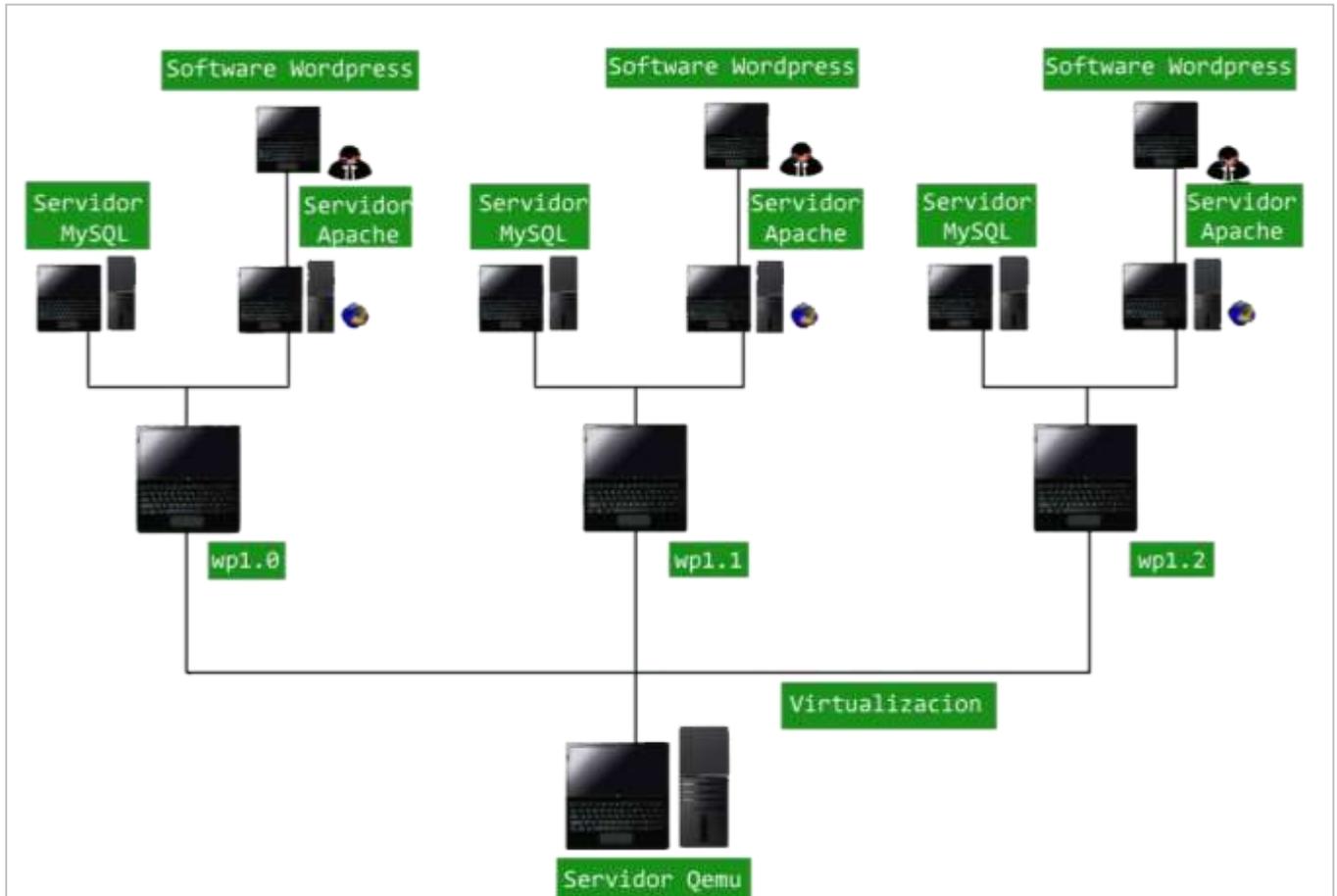


Figura (3). Utilizando virtualización con QEMU para proveer servicios de blogging.

Para llevar a cabo este proyecto se realizó por medio de la aplicación de 11 pasos iniciales fundamentales para cumplir con los objetivos planteados.

Paso 1: Vamos instalar y configurar qemu, conocer su funcionalidad, siendo este nuestro punto de partida ya que qemu nos permitirá crear máquinas virtuales.

Paso 2: Aquí crearemos una máquina virtual y la llamaremos db1, esta máquina será nuestro servidor de base de datos.



Paso 3: Instalaremos MySQL en la máquina virtual db1, pero antes de instalarlo vamos a conocer un poco acerca de MySQL, para comprender su uso.

Paso 4: En este capítulo crearemos otra máquina virtual, la llamaremos wp1.0, esta máquina será nuestro software de blogging.

Paso 5: Aquí vamos a instalar wordpress en la máquina virtual wp1.0, pero antes de instalarlo vamos a conocer que es wordpress y cuál es su utilidad.

Paso 6: En este apartado conoceremos más a fondo sobre la funcionalidad de wordpress y vamos a configurar el servidor de base de datos db1 para permitir acceso remoto desde otro host y también configuraremos wordpress para que haga uso de db1 como servidor de base de datos.

Paso 7: Haremos que la máquina virtual sea accesible desde una herramienta de administración remota vía web, para ello vamos a instalar la herramienta webmin.

Paso 8: En este capítulo se van a crear 2 máquinas virtuales más, para dos nuevos clientes y las llamaremos wp1.1 y wp1.2.

Paso 9: Aquí instalaremos wordpress en las maquinas wp1.1 y wp1.2 para brindarles servicios de blogging a los nuevos clientes y estas máquinas harán uso de db1 como servidor de base de datos.

Paso 10: Ya tenemos tres clones de la misma máquina, ahora en este capítulo lo siguiente que haremos será repartir el tráfico entre las tres máquinas para ello instalaremos squid.

Paso 11: En este capítulo vamos instalar un balanceador de carga apache para que reparta las peticiones entre las tres máquinas, con el propósito de maximizar la amplitud de proceso de información y de ejecución de tareas.



## **7.1 Instalar y configurar QEMU en una máquina de desarrollo.**

Antes de comenzar con la instalación y configuración de QEMU, vamos a conocer un poco que es QEMU, cuál es su utilidad, su objetivo principal y cuáles son sus técnicas de emulación.

QEMU es una aplicación que nos permite crear máquinas virtuales dentro de un sistema operativo, esto quiere decir que podemos ejecutar una gran cantidad de los sistemas operativos que existen, instalándolos en una máquina virtual, estas máquinas virtuales pueden ejecutarse en cualquier tipo de microprocesador o arquitectura, un caso muy común es que estemos utilizando GNU/Linux y queramos también instalar otro sistema operativo, en estas circunstancias QEMU nos será de mucha utilidad porque nos permite instalar varios sistemas operativos en un mismo hardware.

Su principal objetivo es ejecutar un sistema operativo dentro de otro sin tener que reparticionar el disco duro, destinando para su ubicación cualquier directorio dentro de éste.

QEMU actúa como un hypervisor, Un hypervisor o monitor de máquinas virtuales es una plataforma que permite aplicar múltiples métodos de control de virtualización para usar al mismo tiempo diversos sistemas operativos en un mismo computador.

Posee dos técnicas de emulación: Emulación del modo usuario y emulación completo del sistema ordenador.

### **7.1.1 Emulación del modo usuario**

en el cual se pueden ejecutar programas compilados para un tipo de CPU en otro tipo de CPU.



### 7.1.2 Modo de emulación completo de sistema de ordenador

En el cual emula un sistema informático por completo incluyendo procesador y varios periféricos. Este se puede usar para proporcionar hosting virtual a varios ordenadores virtuales en un único ordenador.

### 7.1.3 Instalación y configuración de QEMU.

Para instalar QEMU lo primero que debemos hacer es comprobar si el procesador tiene las extensiones de virtualización de hardware.

Lo comprobamos con el siguiente comando:

### 7.1.4 Línea de código para verificar extensiones de virtualización hardware

```
root@debian:~# egrep -c "(svm|vmx)" /proc/cpuinfo
4
root@debian:~# █
```

*Figura (4).* Código de verificación de extensiones de virtualización hardware

Si el comando nos devuelve un valor 1 o mayor estará indicándonos la cantidad de procesadores que poseen dicha característica, la cual debemos habilitar en caso que no lo este. Pero si el valor devuelto es un 0 entonces el procesador no soporta virtualización de hardware.

En nuestro caso como podemos ver nos devuelve un valor 4 por lo que nos indica que tenemos 4 procesadores que poseen extensiones de virtualización de hardware.

Una vez hecho esto procedemos a instalar QEMU y sus paquetes necesarios, uno de ellos es libvirt una librería escrita en C para interactuar con las recientes capacidades de virtualización de las versiones modernas de Linux y otros sistemas operativos, también debemos instalar virt-manager que es la interfaz gráfica para administrar las máquinas virtuales en nuestro escritorio.

Esto lo hacemos con la siguiente línea.

```
root@debian:~# apt-get install qemu-kvm libvirt-bin bridge-utils virt-manager hal
```



Una vez instalado procedemos a hacer a nuestro usuario miembro del grupo libvirt, porque solo el usuario root y los miembros del grupo libvirt son los únicos que tienen permisos para utilizar las máquinas virtuales.

Para hacer esto vamos a utilizar la siguiente línea de código.

```
root@debian:~# adduser nuria libvirt
```

Donde nuria es el usuario que queremos hacer miembro del grupo libvirt.

Hecho esto ya podemos utilizar el administrador de máquinas virtuales virt-manager

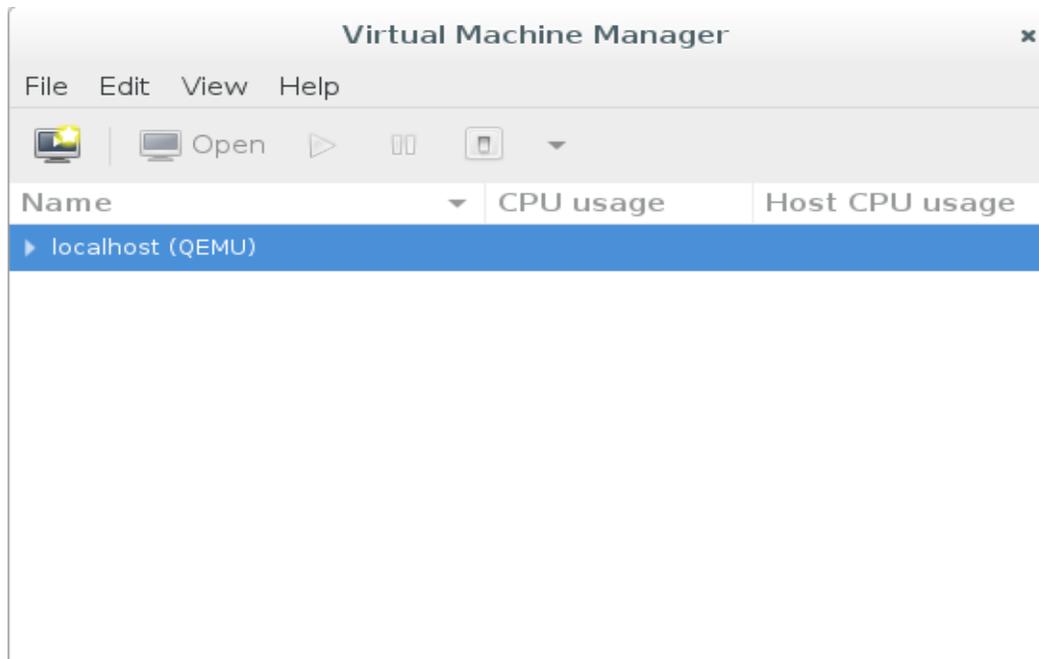


Figura (5). Imagen de la interfaz gráfica para administrar las máquinas virtuales

## 7.2 Crear una máquina virtual llamada DB1.

Para crear una máquina virtual en qemu lo podemos hacer desde el administrador de máquinas virtuales o bien desde línea de comando

En nuestro caso lo haremos desde la línea de comando.

Primero iniciamos el demonio

```
root@debian:~# service libvirtd start
```

Accedemos al hypervisor qemu de la siguiente manera.

```
root@debian:~# virsh -c qemu:///system
```



## 7.2.1 Crearemos la máquina virtual

Para ello primero debemos tener una ISO en nuestro sistema anfitrión, una ISO es un fichero informático donde se guarda una copia o imagen exacta de un sistema de archivo, se rige por el estándar ISO 9660, de ahí su nombre, algunos de los usos más comunes incluyen la distribución de sistemas operativos, por lo que es necesario tenerla para instalar nuestro sistema operativo debían en la máquina virtual que vamos a crear.

Ahora nos aseguramos de que tenemos una ISO en nuestro sistema anfitrión con la siguiente línea de código en la cual ponemos el directorio seguido de la orden ls.

```
root@debian:~# ls /var/lib/libvirt/  
boot  debian-8.5.0-i386-netinst.iso  dnsmasq  images  qemu  sanlock  
root@debian:~# █
```

Figura (6). Línea de código para acceder al directorio libvirt

Como podemos observar nos muestra que tenemos la iso de debían-8.5.0. Hecho esto creamos un directorio donde vamos almacenar nuestras máquinas virtuales, lo hacemos con el siguiente comando.

```
root@debian:~# mkdir /var/lib/libvirt/images
```

Luego de esto lanzamos la línea para crear la máquina.

```
root@debian:~# virt-install --connect qemu:///system --name db1 --memory  
512 disk path=/var/lib/libvirtimages/db1.qcow2.size=10 --vcpus=1 -c  
/var/lib/libvirt/debian-8.5.0-i386-netinst.iso --vnc --os-type Linux --network  
bridge=br1 --noautoconsole --hvm --keymap es
```

Donde conectamos con el hypervisor indicando el nombre de la máquina en este caso le damos el nombre de db1, la memoria RAM en MB, el disco de la nueva máquina, el número de cpus virtuales.

Habilitamos escritorio vnc ya que este es un programa basado en una estructura cliente-servidor que permite tomar el dominio de un ordenador servidor a



través de un ordenador cliente remotamente, también conocido como software de escritorio remoto.

Aquí también vamos a especificar el tipo de sistema operativo que va a ser instalado e indicamos la interfaz de red.

Lo que resta es completar la instalación de debían desde el modo gráfico y seguir los pasos convenientes.

### **7.2.2 Instalacion de MYSQL en la máquina virtual llamada DB1.**

MySQL cuyas siglas en ingles se trasladan a My Structured Query Language es un sistema de administración de base de datos relacional, multihilo y multiusuario, lo que le permite ser utilizado por varias personas al mismo tiempo, e incluso, hacer diversas consultas a la vez, siendo este sumamente versátil. Hoy en día es uno de los más importantes en lo que hace al diseño y programación de base de datos.

Para instalar MySQL procedemos a realizar los siguientes pasos que permitirán una instalación segura y lista para operar con un servidor LAMP.

LAMP es un sistema de infraestructura de internet que usa las siguientes herramientas.

- Linux, el sistema operativo.
- Apache, el servidor web.
- MySQL, el gestor de base de datos.
- Perl, PHP o Python, los lenguajes de programación.

Para llevar a cabo la instalación lo primero que vamos hacer es lanzar la siguiente línea.

```
root@db1:~# apt-get install mysql-server mysql-client
```

Una vez que realizamos la instalación vamos a conectar con el servidor y nos pedirá introducir la contraseña para el usuario root a cómo podemos observar a continuación.



```
root@db1:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 38
Server version: 5.5.50-0+deb8u1 (Debian)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

*Figura (7). Creación del usuario y definición de la contraseña*

Luego crearemos un usuario y definiremos una contraseña para dicho usuario.

```
mysql> CREATE USER 'wordp'@'localhost' IDENTIFIED BY '123QWE';
Query OK, 0 rows affected (0.01 sec)
```

*Figura (8). Creación del usuario Wordp definiendo el host y contraseña*

Aquí mostramos la creación del usuario wordp definiendo el host y la contraseña.

Y también crearemos la base de datos para que el usuario pueda almacenar información

```
mysql> CREATE DATABASE IF NOT EXISTS `wordp-d`;
Query OK, 1 row affected (0.00 sec)

mysql> _
```

*Figura (9). Autorización del servidor de base de datos*

Con esto ya quedara listo nuestro servidor de base de datos.

### **7.3 Crear una máquina virtual llamada WP1.0.**

Vamos a crear una máquina virtual llamada wp1.0, luego instalaremos en ella wordpress para brindar servicios de blogging a nuestro cliente.



Como ya anteriormente habíamos creado una máquina virtual no es necesario repetir los pasos anteriores. En esta ocasión lanzaremos de una vez la línea para crear la máquina wp1.0.

```
root@debian:~# virt-install --connect qemu:///system --name wp1.0 --memory 512 --
disk path=/var/lib/libvirt/images/wp1.0.qcow2,size=4 --vcpus=1 -c /var/lib/libvirt
/debian-7.11.0-i386-netinst.iso --vnc --os-type linux --network bridge=br1 --noaut
oconsole --hvm --keymap es

Empezando la instalación...
Asignando 'wp1.0.qcow2'                | 4.0 GB    00:00
Creando dominio...                    | 0 B      00:00
La instalación del dominio continúa en progreso. Puede reconectarse a
la consola para completar el proceso de instalación.
root@debian:~#
```

Figura (10). Línea de código para crear una máquina virtual llamada wp1.0

Donde conectamos con el hypervisor indicando el nombre de la máquina, la memoria RAM en MB, el disco de la nueva máquina, el número de cpus virtuales, habilitamos escritorio vnc para poder conectar desde un cliente, especificamos también el tipo de sistema operativo que va a ser instalado e indicamos la interfaz de red.

Luego de esto se debe ir al administrador de máquinas virtuales para instalar el sistema operativo en la máquina que acabamos de crear siguiendo los pasos convenientes.

#### 7.4 Instalación de WORDPRESS en la máquina virtual llamada WP1.0

Antes de instalar Wordpress en la máquina virtual vamos a conocer un poco acerca de este software.

Wordpress es un Sistema de gestión de contenido, orientado en las creaciones de cualquier tipo de sitios web, ya que es muy adaptable y es un software de código abierto convirtiéndose en la herramienta más usada y fácil para crear páginas o blog.

Es una plataforma donde podemos crear tiendas virtuales, paginas corporativas sitios web más estáticos o tradicionales, además permite a los usuarios ahorrar tiempo y esfuerzo, ya que hace muy fácil empezar una página web.



WordPress, en principio, es un método de publicación web fundamentado en entradas ordenadas por fecha, las entradas corresponden a una o más categorías o taxonomías. También, cuenta con un administrador de páginas estáticas no cronológicas.

El diseño visual y la estructura del sitio dependen de un sistema de plantillas independiente del contenido, y dependiendo de su autor pueden tener diversas alternativas de personalización.

Lo primero que haremos será instalar los paquetes necesarios, para luego proceder a instalar nuestro sistema wordpress.

Escribimos en la máquina virtual wp1.0 la siguiente línea:

```
root@debian:~# apt-get install apache2 libapache2-mod-php5 php5 php5-curl php5-intl php5-mcrypt php5-mysql php5-sqlite php5-xmlrpc mysql-server mysql-client
```

Una vez finalizada la instalación procedemos a configurar apache,

Crearemos el directorio `/var/www/worpress` para usarlo como directorio de trabajo.

Lo hacemos con la siguiente línea:

```
root@debian:~# mkdir -p /var/www/wordpress
```

Una vez hecho esto vamos a crear el archivo para nuestro sitio wordpress

```
root@debian:~# nano /etc/apache2/sites-available/wordpress
```

El cual tendrá el siguiente contenido.



```
NameVirtualHost 192.168.122.181:80
<VirtualHost 192.168.122.181:80>
  ServerAdmin nuriangelesv@gmail.com
  ServerName debian.debian.qemu.com

  DocumentRoot /var/www/wordpress

  <Directory /var/www/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride All
    Order allow,deny
    allow from all
  </Directory>

  ErrorLog ${APACHE_LOG_DIR}/error.log

  #Possible values include: debug, info, notice, warn, error, crit,
  #alert, emerg,
  LogLevel warn
```

[ 22 líneas leídas ]

Figura (11). Contenido del archivo del sitio wordpress

Primero colocamos la línea NameVirtualHost y ponemos la dirección ip de nuestro host y el puerto 80 que es el puerto por defecto por medio del cual el servidor http escucha la petición hecha por un cliente.

Luego la línea ServerAdmin ponemos nuestro correo del administrador y en ServerName ponemos el nombre que el servidor utiliza para identificarse.

Ahora debemos escribir la línea DocumentRoot el cual indica el directorio donde se almacenan los documentos web.

La directiva DocumentRoot va a compañada de una directiva llamada Directory en la cual se especifican que servicios y características están permitidos o prohibidos en el directorio.

La línea Options Indexes FollowSymLinks MultiViews, nos indica varias posibles opciones de comportamiento del servidor FollowSymLinks este seguirá los enlaces simbólicos y con Multiviews permitirá mostrar contenido en función de diversos valores.

En la siguiente línea AllowOverride All se controla que directivas se pueden



situar en los ficheros `.htaccess`, este es la abreviatura de Hypertext Access, y se trata de un fichero de configuración utilizado por servidores web basados en Apache.

Con la línea `Order allow,deny` se controla el orden en que se evalúan las directivas `Allow` y `Deny`. La directiva `Allow` afecta a los hosts que pueden acceder a una determinada área del servidor, y `Deny` afecta a los hosts que no pueden acceder a una determinada área del servidor.

Pasamos a `ErrorLog $(APACHE_LOG_DIR)/error.log` aquí indica la ubicación del archivo de registro de errores en las consultas.

Y `LogLevel warn` controla el número de mensajes registrados en `error_log`.

Vamos a deshabilitar el sitio configurado por apache en nuestro caso es

`000-default`

```
root@debian:~# a2dissite 000-default
Site 000-default disabled.
To activate the new configuration, you need to run:
  service apache2 reload
root@debian:~# █
```

Figura (12). Deshabilitar el sitio 000-default

Y habilitamos nuestro sitio wordpress.

```
root@debian:~# a2ensite wordpress
Enabling site wordpress.
To activate the new configuration, you need to run:
  service apache2 reload
root@debian:~# █
```

Figura (13). Habilitación del sitio wordpress

También editamos `/etc/hosts` para que quede de la siguiente manera.

Aquí agregamos el nombre del dominio para la dirección IP de nuestro host.

Luego añadimos el módulo `rewrite` de apache



```
root@debian:~# a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
  service apache2 restart
root@debian:~# █
```

*Figura (14).* Ajuste php adaptado a las necesidades de wordpress

Después de esto pasaremos a ajustar php para adaptarlo a las necesidades de wordpress

Esto lo haremos editando el archivo `/etc/php5/apache2/php.ini` en el cual modificaremos las siguientes variables para cambiar la limitación de tamaño en subidas de ficheros a wordpress.

```
max_execution_time = 900
```

Vamos a cambiar el tiempo máximo que se permita que los scripts PHP tarden en ejecutarse y le daremos un valor de 900

```
max_input_time = 900
```

Cambiaremos el valor a 900 el cual será el tiempo máximo en segundos que se permita a un script para que procese todos los datos que reciba de entrada, como ficheros que se están subiendo al servidor, datos que vienen por POST o por GET

```
default_socket_timeout = 900
```

Esta directiva determina la validez de tiempo de espera de un flujo basado en socket en segundos, vamos a cambiar su valor a 900.

```
memory_limit = 512M
```

Con esta directiva se establece el máximo de memoria en bytes que un script puede consumir y ayuda a prevenir que scripts mal programados consuman toda la memoria disponible en el servidor, le daremos el límite de 512M.

```
post_max_size = 64M
```

Aquí colocamos un valor de 64M que será el tamaño máximo del post

```
upload_max_filesize = 64M
```



Se utiliza para asignar el tamaño máximo de los ficheros que deseas que se permita subir, y le damos un valor de 64M

```
max_file_uploads = 32
```

Esta directiva determina el número máximo de ficheros a subir permitidos de forma simultánea, le daremos el valor de 32.

Ahora guardamos, salimos, y reiniciamos apache para que inicie con los valores asignados anteriormente

```
root@debian:~# /etc/init.d/apache2 restart  
[ ok ] Restarting apache2 (via systemctl): apache2.service.
```

Figura (15). Acceso a mysql para creación de usuario y base de datos para wordpress

Vamos a acceder a mysql para crear un usuario y una base de datos para wordpress.

Lo primero al conectarnos a mysql será ingresar la contraseña root.

```
root@debian:~# mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 37  
Server version: 5.5.50-0+deb8u1 (Debian)
```

Figura (16). Acceso a mysql mediante un password

```
mysql> CREATE DATABASE wordpress;  
Query OK, 1 row affected (0.00 sec)  
  
mysql> █
```

Figura (17). Creación de la base de datos wordpress

Aquí se debe crear el usuario para nuestro sistema wordpress y lo hacemos de la siguiente forma.



```
mysql> GRANT ALL ON wordpress.* To wordpress@localhost IDENTIFIED BY '123qwe';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> █
```

Figura (18). Creación de un usuario wordpress en mysql

Hecho esto ya podemos proceder a la descarga de wordpress desde wget pero antes debemos asegurarnos de que wget está instalado esto lo hacemos con el siguiente comando

```
root@debian:~# apt-get install wget
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
wget ya está en su versión más reciente.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 10 no actualizados.
root@debian:~# █
```

Figura (19). Descarga de worpress desde wget

Como vemos ya está instalado por lo tanto procedemos a descargar nuestro sistema wordpress desde wget.

```
root@debian:~# wget --content-disposition http://wordpress.org/latest.tar.gz
```

Ya descargado pasamos a descomprimirlo de la siguiente manera

```
root@debian:~# tar -C /var/www -xf wordpress-4.5.3.tar.gz
```

Asegurándonos de cambiarle los permisos para evitarnos futuros conflictos.

```
root@debian:~# chown -R www-data:www-data /var/www
root@debian:~# find /var/www -type f -exec chmod 0600 {} \;
root@debian:~# find /var/www -type d -exec chmod 0700 {} \;
root@debian:~# █
```

Figura (20). Cambios de permisos para evitar futuros conflictos

Una vez terminado esto pasamos a configurar wordpress desde nuestro navegador web.



Figura (22). Configuración de wordpress desde el navegador web

Colocamos la dirección del host, elegimos el idioma y llenamos los campos. Hecho esto ya tendremos listo nuestro sistema wordpress.

## 7.5 Configurar Wordpress para que haga uso de DB1 como servidor de base de datos.

Antes de configurar Wordpress y nuestro servidor de base de datos vamos profundizar en la funcionalidad de Wordpress.

Uno de los puntos fuertes de Wordpress al igual que otros gestores de contenidos es precisamente eso, la gestión de contenido y su facilidad para insertar, modificar o eliminar contenidos de nuestro sitio web.

Estos contenidos se pueden clasificar en varios tipos. Entradas, Comentarios y Páginas.

### 7.5.1 Entradas

Son contenidos incluidos al WordPress, en ellas es importante la fecha de publicación. Es decir, añadiremos contenido al blog con formato de tipo entrada si deseamos que éste aparezca en orden cronológico inverso, el cual sería los más modernos primeros y los más antiguos al final.

Las Entrada son los elementos característicos de un blog.

Para acceder a ellas, suele haber un menú tipo calendario en uno de los laterales del blog, con enlaces a las entradas ordenadas por fechas.



### **7.5.2 Comentarios**

En el caso de las entradas y las páginas en los blogs tipo WordPress, los usuarios o visitantes pueden dejar sus comentarios bajo la publicación. Esto es opcional, pero normalmente es lo acostumbrado en un blog. No obstante, se pueden activar o desactivar e incluso requerir registro del usuario antes de poder publicar un comentario bajo una entrada o una página.

### **7.5.3 Páginas**

Si la fecha de publicación no es relevante ni existe un orden relacionado con la fecha u orden de publicación entonces vamos añadir el contenido en forma de página. Las páginas efectivamente no son elementos de un blog como tal, sin embargo, se nos permite también este elemento para añadir información estática al blog.

Si queremos acceder a las distintas páginas de un blog, se pueden colocar de forma automática enlaces y menús de enlaces con el orden que deseemos.

### **7.5.4 Páginas estáticas**

Las páginas estáticas nos ofrecen una posibilidad para visualizar la información del blog de forma diferente a los artículos. Estos artículos deben tener categorías o etiquetas para que luego se puedan recuperar y consultar la información.

Las páginas no llevan este sistema de organización y la administración del blog las sitúa en un espacio diferente al de los artículos. Las páginas creadas son accesibles mediante un menú de acceso que suele estar bien en el menú lateral de navegación o en una pestaña de la plantilla con un menú desplegable.

## **7.6 Configuración de Wordpress y el servidor de base de datos**

Para empezar, nos aseguramos de crear un usuario en MYSQL con el nombre de wordp-d en la maquina wp1.0 en la cual ha sido configurado wordpress.



```
mysql> CREATE USER 'wordp-d'@'192.168.122.181' IDENTIFIED BY '123qwe';  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> █
```

*Figura (23).* Creación del usuario en MYSQL en la maquina wp1.0

Ahora lo que haremos será configurar db1 para permitir acceso remoto desde otro host en este caso wp1.0.

Para realizar esto haremos lo siguiente.

Vamos a editar la configuración del servidor MYSQL, por lo cual vamos acceder al archivo `/etc/mysql/my.cnf`.

```
# localhost which is more compatible and is not less secure.  
bind-address          = 192.168.122.158  
#
```

*Figura (24).* Configuración del servidor MYSQL

Localizaremos la línea donde se encuentre la variable `bind-address` para definir en qué dirección IP atiende el servidor MYSQL en nuestro caso es `192.168.122.158`, después de esto vamos a guardar y reiniciamos el servidor MYSQL

```
root@db1:~# /etc/init.d/mysql restart  
[ ok ] Restarting mysql (via systemctl): mysql.service.  
root@db1:~# _
```

*Figura (25).* Reiniciar el servidor MYSQL

Ahora nos conectaremos al servidor MYSQL, nos pedirá la contraseña root.

```
root@db1:~# mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

*Figura (25).* Acceder al servidor MYSQL mediante un password

Y vamos a darle privilegios a nuestro usuario de `wordp-d` que se encuentre en wp1.0 para que pueda conectarse remotamente a la base de datos `wordp-d` que se encuentra en nuestro servidor de bases de datos db1.



```
mysql> GRANT ALL ON `wordpress`.* TO 'wordpress'@'192.168.122.181' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.00 sec)

mysql> _
```

*Figura (26).* Línea de código para conectarse remotamente a la base de datos

Hecho esto salimos del servidor MYSQL y vamos a abrir el puerto 3306 en el firewall del servidor de base de datos, además vamos a permitir el acceso al host 192.168.122.181 en el cual hemos configurado wordpress.

```
root@db1:~# iptables -A INPUT -i eth1 -s 192.168.122.181 -p tcp --destination-port 3306 -j ACCEPT
root@db1:~# _
```

*Figura (27).* Abrir puerto 3306 en el firewall para permitir acceso al host de wordpress

Ahora solo nos queda verificar el acceso remoto al servidor MYSQL desde el wp1.0 con el siguiente comando.

```
root@debian:~# mysql -u wordpress -h 192.168.122.158 -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 38
Server version: 5.5.50-0+deb8u1 (Debian)
```

*Figura (28).* Verificación del acceso remoto al servidor MYSQL desde el wp1.0

Como podemos observar nos pide la contraseña del usuario wordpress y listo, ya tenemos acceso remoto a nuestro servidor de base de datos MySQL.

Pero aún nos falta configurar wordpress para que ahora haga uso de WordPress como su base de datos esto lo haremos desde el archivo de configuración de wordpress wp-config.php.



```
GNU nano 2.2.6          Fichero: wp-config.php
// ** MySQL settings - You can get this info from your web host ** //
/* The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/* MySQL database username */
define('DB_USER', 'wordpress');

/* MySQL database password */
define('DB_PASSWORD', '123qwe');

/* MySQL hostname */
define('DB_HOST', 'db1');

/* Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8mb4');

/* The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');
```

Figura (29). Configuración de wordpress para que haga uso de Word-p

Donde DB\_NAME tendrá el nombre de la base de datos que se encuentra en db1, DB\_USER será el nombre del usuario, DB\_PASSWORD será la contraseña de dicho usuario Y DB\_HOST tendrá el nombre del host donde se encuentra la base de datos en nuestro caso db1.

Hecho esto accedemos desde el navegador

Accedemos a terminar de configurar wordpress y nuestro sitio, por ello vamos introducir los datos que se nos pide



Figura (30). Configuración final de nuestro sitio en wordpress

Nos pedirá nuestro usuario y contraseña para poder acceder a nuestro sitio, después de introducir los datos ya podemos acceder sin ningún problema, como podemos observar a continuación.

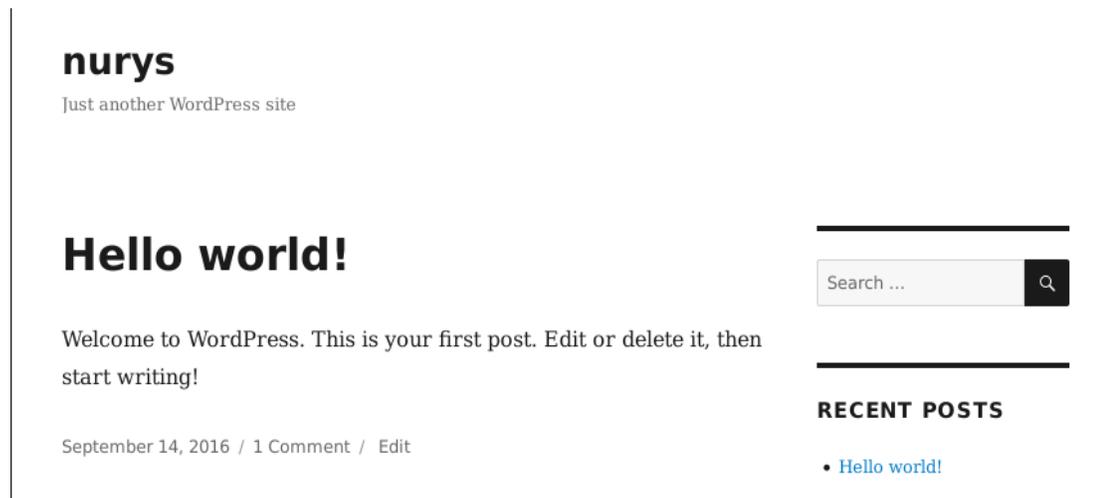


Figura (31). Panel de acceso a nuestro sitio

## 7.7 Hacer que la maquina sea accesible desde una herramienta de administración remota vía web.

Para esto vamos a descargar la herramienta webmin.

### 7.7.1 ¿Qué es Webmin?

Webmin es una herramienta vía web en la cual se pueden configurar sistemas operativos al igual que sus aspectos internos, tales como usuarios, cuotas



de espacios, apagado de equipo, archivos de configuración etc. También controlar y modificar aplicaciones libres.

Webmin permite la administración de sistemas Linux desde cualquier navegador, esta herramienta está escrita en Perl versión 5, y se comunica mediante TCP por el puerto 10000.

Vamos a descargar e instalar esta herramienta en la maquina wp1.0

Primero debemos descargar el programa, para eso vamos a la sección de descargas de la página oficial de webmin, y en nuestro caso vamos a elegir debían y derivados.

RPM suitable for Redhat, Fedora, CentOS, SuSE or Mandrake Linux

<http://prdownloads.sourceforge.net/webadmin/webmin-1.810-1.noarch.rpm> 16.3 MB

**Debian package suitable for Debian, Ubuntu or other derived Linux**

**[http://prdownloads.sourceforge.net/webadmin/webmin\\_1.810\\_all.deb](http://prdownloads.sourceforge.net/webadmin/webmin_1.810_all.deb) 14.8 MB**

Source RPM suitable for Redhat, Fedora, CentOS, SuSE or Mandrake Linux

<http://prdownloads.sourceforge.net/webadmin/webmin-1.810-1.src.rpm> 15.1 MB

*Figura (32).* Descarga del programa desde la página oficial de webmin

Una vez descargado pasamos a la consola para instalarlo con la siguiente orden.

```
root@debian:~# apt-get install -f
```

Ahora ya tendremos el administrador web instalado, nos dirigimos al navegador local y hacemos lo siguiente.

Escribimos la dirección del host con el puerto 10000.



Figura (33). Confirmación de la excepción de seguridad  
Le damos confirmar excepción de seguridad y entramos.



Figura (34). Acceso al webmin  
Y colocamos nuestro nombre de usuario root y contraseña.



## Provisión de servicios de blogging escalable basado en virtualización con QEMU.

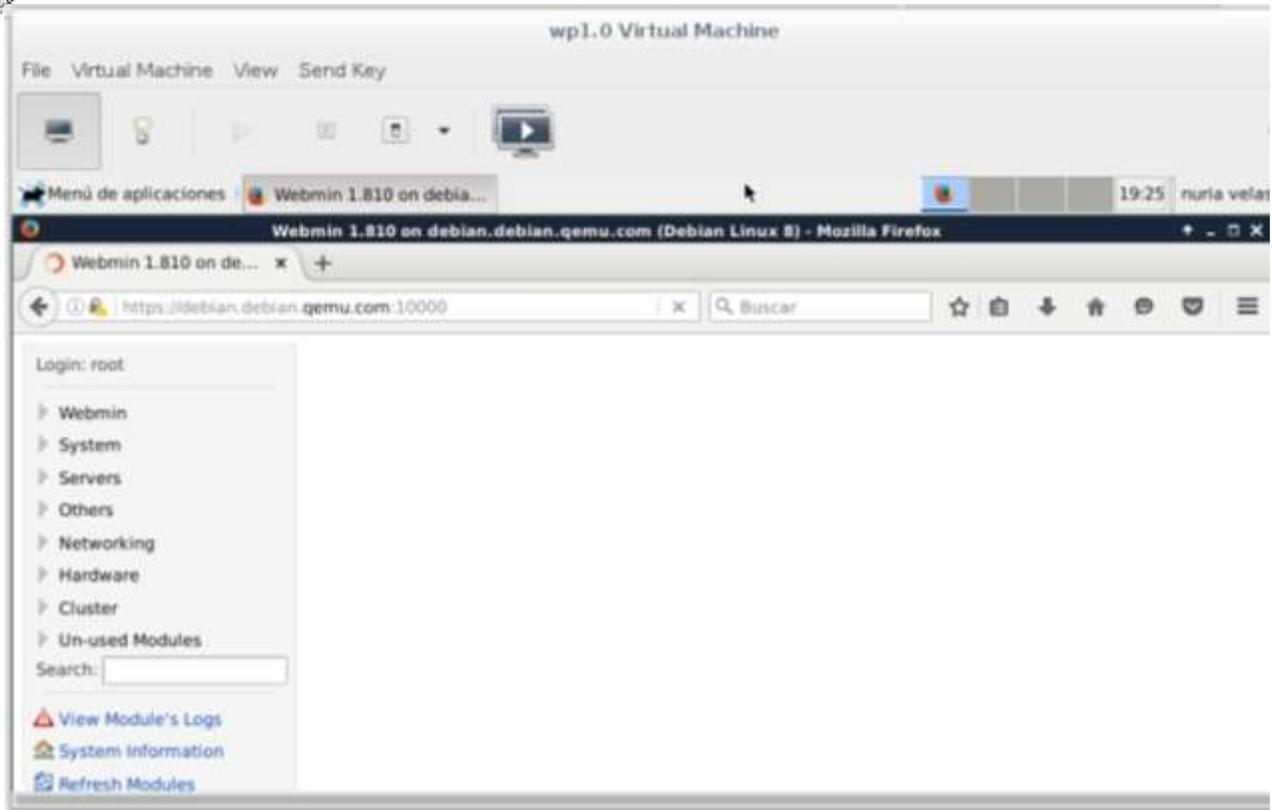


Figura (35). Ventana del acceso del webmin

Y a cómo se puede observar, accedemos perfectamente.

Ahora nos conectamos desde otro host remotamente y a través del navegador podemos acceder a nuestra máquina virtual wp1.0 y administrarla. Introducimos la dirección ip del host wp1.0 y el puerto 10000.



Figura (36). Ventana de acceso desde otro host remoto para acceder y administrar nuestra máquina virtual wp1.0

Introducimos los datos y accedemos sin ningún problema.



Figura (37). Ventana del acceso de la maquina wp1.0

Una vez conectados no ubicamos en Others, y luego entramos en Text Login para acceder a la consola de nuestra máquina virtual wp1.0

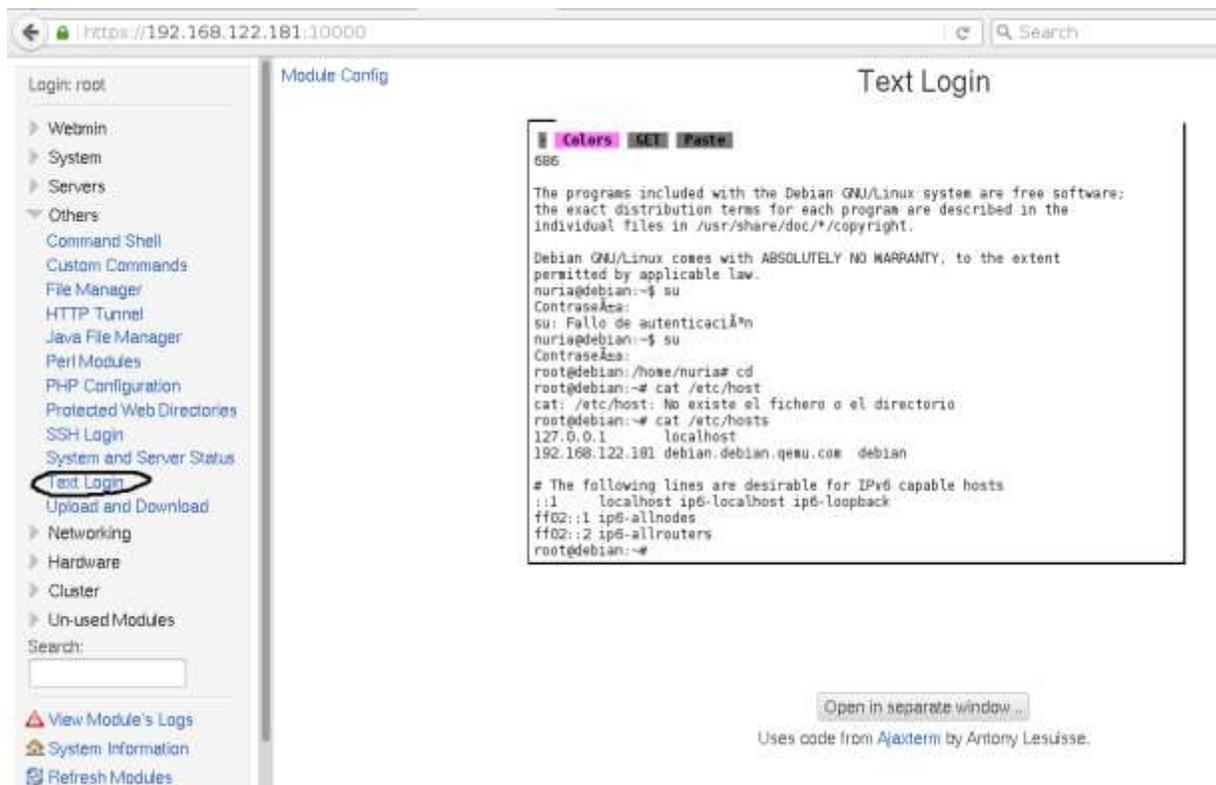


Figura (38). Configuración de la consola de la máquina virtual wp1.0



En la siguiente imagen observamos que podemos administrar la máquina.

```
Colors GET Paste
root@debian:~# cd /var/www/wordpress
root@debian:/var/www/wordpress# ls -l
total 184
-rw----- 1 www-data www-data 418 sep 24 2013 index.php
-rw----- 1 www-data www-data 19935 mar 5 2016 license.txt
-rw-r--r-- 1 www-data www-data 7360 sep 14 09:02 readme.html
-rw----- 1 www-data www-data 5032 ene 27 2016 wp-activate.php
drwx----- 9 www-data www-data 4096 jun 21 10:17 wp-admin
-rw----- 1 www-data www-data 364 dic 19 2015 wp-blog-header.php
-rw----- 1 www-data www-data 1476 ene 30 2016 wp-comments-post.php
-rw-rw-rw- 1 www-data www-data 3129 sep 13 18:22 wp-config.php
-rw----- 1 www-data www-data 2853 dic 16 2015 wp-config-sample.php
drwx----- 6 www-data www-data 4096 sep 13 18:23 wp-content
-rw----- 1 www-data www-data 3286 may 24 2015 wp-cron.php
drwx----- 16 www-data www-data 12288 jun 21 10:17 wp-includes
-rw----- 1 www-data www-data 2380 oct 24 2013 wp-links-opml.php
-rw----- 1 www-data www-data 3316 nov 5 2015 wp-load.php
-rw----- 1 www-data www-data 33837 mar 5 2016 wp-login.php
-rw----- 1 www-data www-data 7887 oct 6 2015 wp-mail.php
-rw----- 1 www-data www-data 13106 feb 17 2016 wp-settings.php
-rw----- 1 www-data www-data 28624 ene 27 2016 wp-signup.php
-rw----- 1 www-data www-data 4035 nov 30 2014 wp-trackback.php
-rw----- 1 www-data www-data 3061 oct 2 2015 xmlrpc.php
root@debian:/var/www/wordpress#
```

Figura (39). Panel de administración de la máquina virtual

## 7.8 Crear dos máquinas virtuales

Vamos a crear 2 máquinas virtuales llamadas wp1.1 y wp1.2 en las cuales instalaremos wordpress para brindar servicios a 2 nuevos clientes y utilizaremos el mismo servidor de base de datos que la maquina wp1.0 y con esto ahorrar costos.

Lo primero será lanzar la línea para la creación de la maquina wp1.1

```
root@debian:~# virt-install --connect qemu:///system --name wp1.1 --memory 512 -
-disk path=/var/lib/libvirt/images/wp1.1.qcow2,size=10 --vcpus=1 -c /var/lib/lib
virt/debian-8.5.0-i386-netinst.iso --vnc --os-type linux --network bridge=br1 --
noautoconsole --hvm --keymap es

Empezando la instalación...
Asignando 'wp1.1.qcow2' | 10 GB 00:00
Creando dominio... | 0 B 00:00
La instalación del dominio continúa en progreso. Puede reconectarse a
la consola para completar el proceso de instalación.
root@debian:~# █
```

Figura (40). Creación de la máquina virtual wp1.1



Lanzamos la línea para la creación de la maquina wp1.2

```
root@debian:~# virt-install --connect qemu:///system --name wp1.2 --memory 512 -
-disk path=/var/lib/libvirt/images/wp1.2.qcow2,size=10 --vcpus=1 -c /var/lib/lib
virt/debian-8.5.0-i386-netinst.iso --vnc --os-type linux --network bridge=br1 --
noautoconsole --hvm --keymap es

Empezando la instalación...
Asignando 'wp1.2.qcow2'                | 10 GB      00:00
Creando dominio...                    |  0 B      00:00
La instalación del dominio continúa en progreso. Puede reconectarse a
la consola para completar el proceso de instalación.
root@debian:~# █
```

Figura (41). Creación de la máquina virtual wp1.2

## 7.9 Instalar wordpress en las maquinas WP1.1 Y WP1.2 para que hagan uso de DB1 como servidor de base de datos.

Instalaremos wordpress en ambas máquinas.

Los mismos pasos que hicimos en la instalación de wordpress en la maquina wp1.0, los haremos para instalar wordpress en las maquinas wp1.1 y wp1.2.

Lo primero que haremos para ambas máquinas virtuales será instalar los paquetes necesarios, para luego proceder a instalar nuestro sistema wordpress.

Cuando finalice la instalación procedemos a configurar apache, crearemos el directorio `/var/www/worpress` para usarlo como directorio de trabajo.

Una vez hecho esto vamos a creamos el archivo para nuestro sitio wordpress en cada una de las máquinas.

```
root@debian:~# nano /etc/apache2/sites-available/wordpress
```

El cual tendrá el siguiente contenido en:



```
nuria@wp1: ~
GNU nano 2.2.6 Fichero: /etc/apache2/sites-available/wordpress
NameVirtualHost 192.168.122.81:80
<VirtualHost 192.168.122.81:80>
  ServerAdmin nuriangelesv@gmail.com
  serverName wp1.1

  DocumentRoot /var/www/wordpress

  <Directory /var/www/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride All
    Order allow,deny
    allow from all
  </Directory>

  ErrorLog ${APACHE_LOG_DIR}/error.log

  # Possible values include: debug, info, notice, warn, error, crit,
  # alert, emerg.
  LogLevel warn
[ 21 líneas leídas ]
```

Figura (42). Creación de archivos para la maquina wp1,1

```
nuria@wp1: ~
GNU nano 2.2.6 Fichero: ../apache2/sites-available/wordpress.conf Modificado
NameVirtualHost 192.168.122.39:80
<VirtualHost 192.168.122.39:80>
  ServerAdmin nurianglesv@gmail.com
  ServerName wp1.2

  DocumentRoot /var/www/wordpress

  <Directory /var/www/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride All
    Order allow,deny
    allow from all
  </Directory>

  ErrorLog ${APACHE_LOG_DIR}/error.log

  # Possible values include: debug, info, notice, war, error, crit,
  # alert, emerg.
  LogLevel warn
```

Figura (43). Creación de archivos para la maquina wp1.2

Hecho esto vamos a deshabilitar el sitio configurado por apache en nuestro



caso es:

000-default y habilitamos nuestro sitio wordpress.

También vamos a editar /etc/host y luego añadimos el módulo rewrite de apache

Después de todo esto pasaremos a ajustar php para adaptarlo a las necesidades de wordpress en las dos máquinas de la misma manera que lo hicimos con la maquina wp1.0.

Esto lo haremos editando el archivo /etc/php5/apache2/php.ini en el cual modificaremos las siguientes variables para cambiar la limitación de tamaño en subidas de ficheros a wordpress:

```
max_execution_time = 900
```

```
max_input_time = 900
```

```
default_socket_timeout = 900
```

```
memory_limit = 512M
```

```
post_max_size = 64M
```

```
upload_max_filesize = 64M
```

```
max_file_uploads = 32
```

Guardamos, salimos y reiniciamos apache para que inicie con los datos configurados.

Luego accedemos a mysql para crear un usuario y una base de datos para wordpress.

Lo mismo para las dos maquinas

Creamos la base de datos y un usuario local y le damos privilegios al usuario para que pueda guardar información en la base de datos.

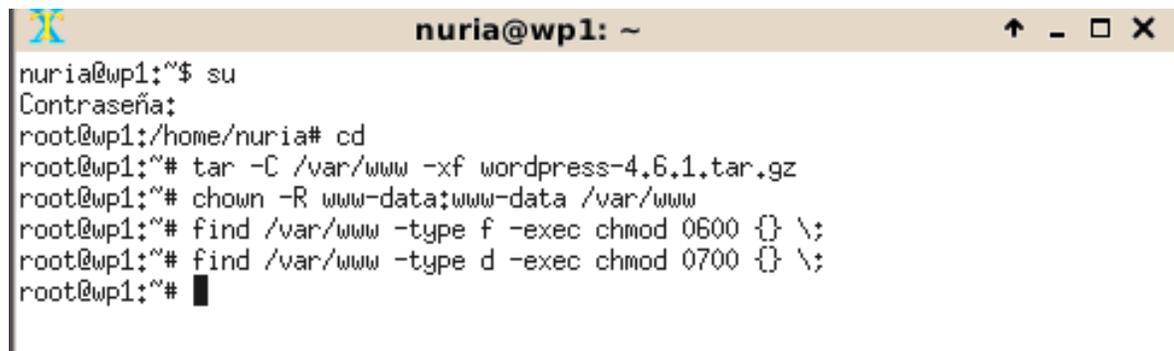


Y procedemos a la descarga de wordpress desde wget.

```
root@wp1:~# wget --content-disposition http://wordpress.org/latest.tar.gz
--2016-09-09 22:15:46-- http://wordpress.org/latest.tar.gz
Resolviendo wordpress.org (wordpress.org)... █
```

Figura (44). Descarga de wordpress desde wget

Ahora descargado pasamos a descomprimirlo, asegurándonos de cambiarle los permisos para evitarnos futuros conflictos.



```
nuria@wp1: ~
nuria@wp1:~$ su
Contraseña:
root@wp1:/home/nuria# cd
root@wp1:~# tar -C /var/www -xf wordpress-4.6.1.tar.gz
root@wp1:~# chown -R www-data:www-data /var/www
root@wp1:~# find /var/www -type f -exec chmod 0600 {} \;
root@wp1:~# find /var/www -type d -exec chmod 0700 {} \;
root@wp1:~# █
```

Figura (45). Descompresión de wordpress y cambios en los permisos

Ya solo nos queda configurar wordpress desde el navegador.

Introducimos la dirección del host, seleccionamos el idioma y llenamos los campos.

Un vez instalado wordpress en las dos maquinas wp1.1 y wp1.2 ya solo nos queda configurarlas para que hagan uso de db1 como servidor de base de datos

Como ya hemos configurado db1 para permitir acceso remoto desde otro host, vamos a conectarnos directamente al servidor MYSQL, pero antes crearemos un usuario en MYSQL en cada una de las máquinas. Y después vamos a crear una base de datos para cada máquina en el servidor de base de datos db1 y les damos permiso a los usuarios de cada una de ellas para que pueda hacer uso de sus respectivas bases de datos.

Ahora solo nos queda verificar el acceso remoto al servidor MYSQL desde el wp1.1



```
root@wp1:~# mysql -u nuria -h 192.168.122.158 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 60
Server version: 5.5.50-0+deb8u1 (Debian)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

Ahora vamos a comprobar el acceso remoto desde wp1.2

```
root@wp1:~# mysql -u wordpress -h 192.168.122.158 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 76
Server version: 5.5.50-0+deb8u1 (Debian)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

---

*Figura (46).* Verificación del acceso remoto al servidor MYSQL desde wp1.2

Hecho esto pasamos a configurar wordpress en ambas máquinas para que ahora hagan uso de la base de datos que creamos en db1, esto lo haremos desde el archivo de configuración de wordpress wp-config.php



En wp1.1

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'nuria');

/** MySQL database username */
define('DB_USER', 'nuria');

/** MySQL database password */
define('DB_PASSWORD', '123qwe');

/** MySQL hostname */
define('DB_HOST', 'db1');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8mb4');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');
```

Figura (47). Configuración de wordpress en wp1.1

En wp1.2

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'wordpress');

/** MySQL database password */
define('DB_PASSWORD', '123qwe');

/** MySQL hostname */
define('DB_HOST', 'db1');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8mb4');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');
```

Figura (48). Configuración de wordpress en wp1.2

Donde DB\_NAME tendrá el nombre de la base de datos que se encuentra en db1,

Y DB\_HOST tendrá el nombre de dicho host.



## 7.10 Ahora hay tres clones del mismo blog lo siguiente es conseguir que se reparta el tráfico entre las tres.

Vamos a repartir el tráfico de ancho de banda entre las máquinas para ello vamos a instalar squid.

Squid es un servidor proxy para web con cache. Se conoce como una de las aplicaciones más populares. Una de sus utilidades es mejorar el rendimiento de todas las conexiones de empresas y particulares a internet guardando las peticiones recurrentes a servidores web y dns en cache.

Lanzamos la línea para instalar squid.

```
root@debian:~# apt-get -y install squid3
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  libcap2 libnetfilter-conntrack3 squid-langpack squid3-common
Paquetes sugeridos:
  squidclient squid-cgi squid-purge resolvconf smbclient ufw winbindd
Se instalarán los siguientes paquetes NUEVOS:
  libcap2 libnetfilter-conntrack3 squid-langpack squid3 squid3-common
0 actualizados, 5 nuevos se instalarán, 0 para eliminar y 50 no actualizados.
Se necesita descargar 2,561 kB de archivos.
Se utilizarán 9,392 kB de espacio de disco adicional después de esta operación.
Des:1 http://http.us.debian.org/debian/ jessie/main libnetfilter-conntrack3 i386
1.0.4-1 [48,2 kB]
Des:2 http://http.us.debian.org/debian/ jessie/main squid-langpack all 20140506-
1 [150 kB]
2% [2 squid-langpack 0 B/150 kB 0%]
```

Figura (49). Línea de instalación de squid

Ya instalado vamos a entrar al archivo de configuración de squid.

```
root@debian:~# nano /etc/squid3/squid.conf
root@debian:~# █
```

Figura (50). Configuración de squid

Vamos a destinar una cantidad específica de ancho de banda para cada máquina dentro de un rango.

En nuestro caso haremos que cada una de las máquinas pueda consumir 512kb/s utilizando algunas reglas establecidas de squid.



```
delay_pools 1
delay_class 1 2
delay_parameters 1 131072/8192 65536/8192
ac lista src 192.168.122.10-192.168.122.199/32
delay_access 1 allow lista
```

Figura (50). Ajuste de las máquinas según reglas establecidas por squid

En la cual definimos el tipo de clase para la regla el cual será clase dos que es el que nos permite dividir cantidades de ancho de banda para cada máquina, y colocamos en parámetros la cantidad que le daremos en números enteros y definimos el rango de las ip en las que será asignada dicha cantidad.

Y por último vamos a configurar un proxy transparente para que todas las peticiones que se le hagan a un puerto 80 las reenvíe hacia el puerto 3128 que utiliza squid.

Como el equipo con squid está en la misma pasarela que el resto de equipos configuraremos el firewall de la siguiente manera.

```
root@debian:~# iptables -t nat A PREROUTING -i eth1 -p tcp --dport 80 -j
REDIRECT --to-port 3128
```

Ya tenemos un proxy para que haga de intermediario entre los pcs de la red y el router de conexión a internet y reparta el tráfico entre el rango de ips asignadas.

### **7.11 Instalar un balanceador de carga que reparta las peticiones entre las tres máquinas WP1.0, WP1.1, Y WP1.2.**

Vamos a utilizar un balanceador de carga apache + mod\_proxy, ya que es uno de los más populares, que nos permite básicamente tener acceso a recursos que se encuentran en otros servidores a través de el mismo apache, y lo realiza enrutando todo el tráfico y canalizándolo al servidor correspondiente.

El balanceo de carga permite distribuir las tareas que tendría que sustentar una única máquina, con el propósito de maximizar la amplitud de proceso de información y de ejecución de tareas. Esto permite que ningún equipo sea parte fundamental del servicio que ofreceremos. De esta manera evitaremos sufrir una detención de dicho servicio debido a una parada de uno de los equipos.



En caso que un enlace o ruta falle, los procesos confirman que los mensajes pueden redirigirse de forma inmediata a otro enlace que es transparente para los usuarios, Como en las tres máquinas tenemos instalado apache, vamos hacer las siguientes configuraciones.

Vamos a utilizar wp1.0 como la maquina apache\_balancer, para ello debemos activar los siguientes módulos.

```
root@debian:~# a2enmod proxy_balancer
```

```
root@debian:~# a2enmod proxy_http
```

Luego preparamos el archivo /etc/hosts, donde colocaremos las ip de las máquinas.

```
127.0.0.1      localhost
192.168.122.181 debian.debian.qemu.com  debian
192.168.122.81  nodo1.mydomain.com
192.168.122.39  nodo2.mydomain.com

# The following lines are desirable for IPv6 capable hosts
::1          localhost ip6-localhost ip6-loopback
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters

BalancerMember http://nodo2.mydomain.com:80 route=lb2

ProxyPass /balancer-manager !
ProxyPass / balancer://mycluster/ lbmethod=byrequests
stickysession=BALANCEID
ProxyPassReverse / http://nodo1.mydomain.com/
```

```
^G Ver ayuda  ^O Guardar   ^R Leer Fich ^Y Pág Ant   ^K CortarTxt ^C Pos ac
^X Salir     ^J Justificar^W Buscar    ^V Pág Sig   ^U PegarTxt  ^T Ortogr
```

Figura (51). Preparación del archivo /etc/host para colocación del ip de las máquinas



.stickysession=BALANCEID con esta opción lo que conseguimos es pasar la variable de sesión al servidor al que le indique el balanceador y así permanecer en el mismo durante toda la sesión aunque actualicemos.

```
BalancerMember http://nodo1.mydomain.com:80 route=lb1

#cluster member 2
BalancerMember http://nodo2.mydomain.com:80 route=lb2

ProxyPass /balancer-manager !
ProxyPass / balancer://mycluster/ lbmethod=byrequests
stickysession=BALANCEID
ProxyPassReverse / http://nodo1.mydomain.com/
ProxyPassReverse / http://nodo2.mydomain.com/

</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

*Figura (52).* pasar la variable de sesión a los nodos

Guardamos el fichero y luego reiniciamos apache

```
root@debian:~# /etc/init.d/apache2 restart
[ ok ] Restarting apache2 (via systemctl): apache2.service.
root@debian:~#
```

*Figura (52).* Guardar y reiniciar apache

Ahora deberemos configurar los nodos para que reciban la variable de sesión.



### En wp1.1

```
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf

RewriteEngine On
RewriteRules
[CO=BALANCEID;balancer.lb1;.mydomain.com]
</VirtualHost>
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Figura (53). Configuración de los nodos para que reciban la variable de sesión en wp1.1

### Guardamos y reiniciamos apache

```
root@wp1:~# /etc/init.d/apache2 restart
[ ok ] Restarting apache2 (via systemctl): apache2.service.
root@wp1:~# █
```

Figura (54). Guardado de los datos y reinicio de apache en wp1.1

### En wp1.2

```
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf

RewriteEngine On
RewriteRule
[CO=BALANCEID;balancer.lb2;.mydomain.com]

</VirtualHost>
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Figura (55). Configuración de los nodos para que reciban la variable de sesión en wp1.2



Guardamos y reiniciamos apache.

```
root@wp1:~# /etc/init.d/apache2 restart
[ ok ] Restarting apache2 (via systemctl): apache2.service.
root@wp1:~# █
```

*Figura (56).* Guardado de los datos y reinicio de apache en wp1.2

Hecho esto ya tenemos configurado nuestro balanceador.



## VIII. CONCLUSIONES

- A través del desarrollo de este proyecto se facilitó proveer los servicios de hostings para blogs escalables mediante el uso de máquinas virtuales brindando servicios y soluciones a las problemáticas de nuestros clientes, utilizando la herramienta de virtualización QEMU.
- Se proporcionó un software de bloggins completamente operativo y funcional, proveyendo el acceso remoto al servidor de base de datos mediante el uso de una máquina virtual.
- Se brindó seguridad a la información garantizando la tolerancia a fallos por medio de balanceo de carga, la cual ejecuta las tareas dentro del servidor de base de datos entre las máquinas.



## IX. REFERENCIAS

- Aoliva. (2008). Balanceo de Carga. Recuperado de [https://aoliva.wordpress.com/2008/01/13/balanceo-de-carga-con-apache-2-mod\\_proxy/](https://aoliva.wordpress.com/2008/01/13/balanceo-de-carga-con-apache-2-mod_proxy/)
- Arias, F. G. (1999). El proyecto de investigación. (3ra ed.). Caracas; Oriol Ediciones.
- Dabidochobits. (2015). Instalación de wordpress en una máquina virtual. Recuperado de <https://www.ochobitshacenunbyte.com/2015/05/19/instalar-wordpress-debian-jessie/>
- Leal, L. J. (2010). Asesoría de tesis. recuperado de <https://asesoriatesis1960.blogspot.com/2010/12/antecedentes-de-la-investigacion.html>
- Linuxito. (2013). Una Maquina accesible desde una herramienta de administración remota. recuperado de <https://www.linuxito.com/gnu-linux/nivel-medio/170-como-habilitar-el-acceso-remoto-a-un-servidor-de-bases-de-datos-mysql>
- Meza, M. O. (2010). El marco teórico en un proyecto de investigación. recuperado de <https://es.scribd.com/doc/38574618/>