

IMPLEMENTACIÓN DE UNA PLATAFORMA BASADA EN CONTENEDORES
PARA DESARROLLO Y HOSTING DE SITIOS WEB CON ALTA
DISPONIBILIDAD

NESTOR IVAN INFANTE MONSALVE
SERGIO ANDRES DULCEY LIZARAZO

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICOMECHANICAS
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2022

IMPLEMENTACIÓN DE UNA PLATAFORMA BASADA EN CONTENEDORES
PARA DESARROLLO Y HOSTING DE SITIOS WEB CON ALTA
DISPONIBILIDAD

NESTOR IVAN INFANTE MONSALVE
SERGIO ANDRES DULCEY LIZARAZO

Trabajo de grado para optar al título de Ingeniero de Sistemas

Director

Manuel Guillermo Florez Becerra, MSc.

Magíster en Informática

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICOMECAICAS
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA

2022

AGRADECIMIENTOS

Antes de iniciar mi vida universitaria en la UIS, siempre he sido aconsejado por familiares egresados de esta grandiosa alma mater y desde allí empezó mi motivación de entrar, para luego conocer a lo largo de mi formación personas maravillosas que compartieron directa e indirectamente conmigo, estas son mis palabras de agradecimiento.

A mi padre y abuela por ser mi apoyo condicional en cualquier cosa que hacía, siempre estuvieron allí y son parte fundamental en mi formación.

A mis tíos Daniel y Alexandra, los primeros ingenieros de la familia, siempre me apoyaron y me guiaron en el camino correcto.

Al director de proyecto y mi compañero de proyecto, que siempre tuvimos un trabajo armónico y un enriquecimiento de conocimiento importante.

A los docentes que me inculcaron su enseñanza tanto de conocimiento como de vida.

Finalmente, a todas las personas que han estado siempre y aportaron de alguna manera en este proceso.

Sergio Andrés Dulcey Lizarazo

AGRADECIMIENTOS

Primeramente, agradecer a la Universidad Industrial de Santander por ofrecerme educación de calidad y permitirme vivir esta etapa tan hermosa, de tantas historias, alegrías, tristezas, frustraciones, pero sobre todo, tanto aprendizaje.

A mi abuelo Juan y mi abuela Geny por siempre apoyarme, educarme con los mejores valores, ofrecerme lo mejor de ellos con todo el amor del mundo.

A mi mamá que siempre está dándome sus consejos, siendo un ejemplo de valentía, berraquera y alentándome siempre a luchar por lo que quiero.

A Daniel, mi hermano que llegó a mi vida a alegrarla siempre con sus risas, sus abrazos, sus juegos y se convirtió en una motivación de siempre ir por lo mejor.

A Majo por motivarme a cada día ser mi mejor versión, brindándome todo su amor, sus consejos y por caminar el sendero de la vida junto a mí.

A Mintoná, Osmelida y Fernando por estar siempre pendientes de mí, cuidarme como su hijo y buscar siempre mi bienestar.

A mi profesor y director de proyecto MSc. Manuel Guillermo por estar al tanto del desarrollo de este proyecto, teniendo la disposición y paciencia para guiarnos cuando no encontrábamos el rumbo.

A Sergio, mi compañero de proyecto, por tener en todo momento una actitud positiva y siempre buscarle una salida a los problemas que se nos presentaron en el desarrollo del proyecto.

Y por último, pero no menos importante, a mis familiares, amigos, profesores, compañeros de clases, a cada persona que ha pasado por mi vida y ha dejado una huella, un mensaje, un consejo que me ha hecho la persona que soy hoy

A todos ustedes, mil gracias.

Nestor Ivan Infante Monsalve

CONTENIDO

	pág.
INTRODUCCIÓN	15
1. SOBRE ESTE TRABAJO DE INVESTIGACIÓN.....	17
1.1. PLANTEAMIENTO DEL PROBLEMA.....	17
1.2. JUSTIFICACIÓN.....	18
1.3. OBJETIVOS.....	19
1.3.1. Objetivo general.....	19
1.3.2. Objetivos específicos	19
2. MARCO DE REFERENCIA.....	20
2.1. ESTADO DEL ARTE.....	20
2.2. MARCO TECNOLÓGICO	22
2.2.1. Computación en la nube	23
2.2.2. Modelos de servicios	23
2.2.2.1. PaaS (Plataforma-as-a-Service)	23
2.2.2.2. IaaS (Infrastructure-as-a-Service).....	23
2.2.2.3. SaaS (Software-as-a-Service)	24
2.2.3. Virtualización.....	25
2.2.3.1. Tecnología virtualización.....	25
2.2.3.2. Máquina virtual.....	25
2.2.3.3. Ubuntu server	26
2.2.3.4. Contenedores	26
2.2.4. Docker.....	27
2.2.4.1. Docker hub.....	28
2.2.4.2. Docker compose	28
2.2.4.3. Dockerfile	28

2.2.5. Web service	28
2.2.5.1. Apache HTTP Server	29
2.2.5.2. Nginx.....	29
2.2.5.3. Protocolo de transferencia	29
2.2.6. Shell.....	30
2.2.7. Filezilla	30
2.2.8. Cliente SSH	31
2.2.9. Fail2ban	31
2.2.10. Framework	32
2.2.10.1. Estructura MERN	32
2.2.10.2. Estructura MEAN	32
2.2.11. Pila de desarrollo web.....	32
2.2.11.1. Pila LAMP	33
2.2.11.2. Pila LEMP	33
2.2.12. Docker swarm	33
2.2.12.1. HAProxy.....	34
2.2.13. Alta disponibilidad	35
3. METODOLOGIA	36
4. DESARROLLO DEL PROYECTO	38
4.1. SITUACIÓN ACTUAL	38
4.1.1. Actualización de los servidores.....	40
4.2. ESQUEMA GENERAL DE LA INFRAESTRUCTURA	44
5. IMPLEMENTACIÓN.....	45
5.1. MÁQUINA VIRTUAL CON USUARIOS ENJAULADOS:.....	45
5.2. USUARIOS ENJAULADOS CON CONTENEDORES DOCKER	48
5.3. ALTA DISPONIBILIDAD USANDO DOCKER SWARM.....	51

5.3.1. Comprobación de prerequisites	53
5.3.2. Crear cluster	54
5.3.2.1. Crear el primer nodo	54
5.4. ACCESO A LA APLICACIÓN DESDE EL EXTERIOR	58
5.5. MANUALES DE USO	59
6. RESULTADOS	60
6.1. PROYECTOS DE ESTUDIANTES	60
6.2. PRUEBA CONTENEDORES DOCKER	62
6.3. ALTA DISPONIBILIDAD A MEIWEB	63
6.4. PRUEBAS	65
7. CONCLUSIONES	68
8. RECOMENDACIONES	69
BIBLIOGRAFIA	70
ANEXOS	74

LISTA DE FIGURAS

	pág.
Figura 1. Modelos de servicios	24
Figura 2. Ilustración de un contenedor Docker y una Máquina Virtual.....	27
Figura 3. Diferencia entre HTTP vs HTTPS.....	30
Figura 4. Framework MERN STACK.	32
Figura 5. Framework MEAN STACK.....	32
Figura 6. Pila de desarrollo LAMP STACK.	33
Figura 7. Pila de desarrollo LEMP STACK.	33
Figura 8. Arquitectura de Docker Swarm.	34
Figura 9. Setup del grupo GID-CONUSS.....	39
Figura 10. Particiones dinámicas del servidor Larusso y Smith.....	41
Figura 11. Ubicación de la carpeta netplan.....	41
Figura 12. Configuración una IP privada dinámica.	42
Figura 13. Interfaz de administración VMM	42
Figura 14. Esquema actual	44
Figura 15. Versión PHP.	46
Figura 16. Menú del administrador.	47
Figura 17. FileZilla	47
Figura 18. phpMyAdmin.....	48
Figura 19. Arquitectura LAMP.....	48

Figura 20. Menú administrativo para contenedores	50
Figura 21. Arquitectura con Docker.	51
Figura 22. Arquitectura Docker Swarm (Cluster)	52
Figura 23. Nodos	55
Figura 24. Stack de Meiweb en Portainer	57
Figura 25. Arquitectura deseada.....	58
Figura 26. Términos y condiciones	59
Figura 27. Grupo B1 Subgrupo 01 - 20212B101.....	60
Figura 28. Grupo B1 Subgrupo 02 - 20212B102.....	60
Figura 29. Grupo B1 Subgrupo 03 - 20212B103.....	61
Figura 30. Grupo B1 Subgrupo 04 - 20212B104.....	61
Figura 31. Contenedores ejecutados para MERN STACK.	62
Figura 32. Página de club unión.	62
Figura 33. Contenedores del aplicativo MEIWEB usando LEMP.....	63
Figura 34. Nodos listados	63
Figura 35. Pagina Meiweb	64
Figura 36. Visualizador de nodos.....	64

LISTA DE TABLAS

	Pág.
Tabla 1. Especificaciones del servidor Larusso.....	39
Tabla 2. Especificaciones del servidor Smith.....	39
Tabla 3. Especificaciones del servidor Johnson	40
Tabla 4. Servidor Smith	43
Tabla 5. Servidor Larusso.....	43
Tabla 6. Servidor Johnson	43
Tabla 7. Especificaciones de la máquina virtual.	46
Tabla 8. Máquina virtual con Docker y Docker Compose.	50
Tabla 9. Especificaciones MV utilizadas en alta disponibilidad.....	52
Tabla 10. Pruebas al primer servicio usuarios enjaulados.....	65
Tabla 11. Pruebas al segundo servicio usuarios enjaulados con contenedores Docker	66
Tabla 12. Pruebas a MEIWEB con Docker Swarm.....	67

LISTA DE ANEXOS

	Pág.
Anexo A. SLA – Terminos y condiciones de uso	74
Anexo B. Manual Administrativo usuarios enjaulados	78
Anexo C. Manual administrativo usuarios enjaulados usando contenedores	91

RESUMEN

TÍTULO: IMPLEMENTACIÓN DE UNA PLATAFORMA BASADA EN CONTENEDORES PARA DESARROLLO Y HOSTING DE SITIOS WEB CON ALTA DISPONIBILIDAD*

AUTOR: NESTOR IVAN INFANTE MONSALVE, SERGIO ANDRES DULCEY LIZARAZO**

PALABRAS CLAVE: CONTENEDORES, ALTA DISPONIBILIDAD, SERVICIO HOSTING, PLATAFORMA, SERVICIOS.

DESCRIPCIÓN: El constante cambio en las tecnologías que se usan para el desarrollo e implementación de páginas web, requiere que las plataformas que brindan el servicio hosting estén también a la vanguardia de las nuevas tecnologías. Uno de los aspectos a tener en cuenta al momento de implementar una plataforma, es la adaptabilidad y escalabilidad, ya que las organizaciones necesitan la flexibilidad de utilizar las aplicaciones en cualquier entorno, sea desarrollo o producción.

Como alternativa a las opciones de virtualización con máquinas virtuales, en este proyecto se realiza el desarrollo de una plataforma con la cual se pueda realizar un encapsulamiento de la aplicación haciendo uso de la tecnología de contenedores mediante la herramienta orquestadora Docker que nos permite tener las librerías necesarias para que una aplicación funcione independientemente de las librerías que tenga el sistema operativo en una maquina anfitriona, lo que facilita la migración, actualización, mantenimiento de los aplicativos y también significa un ahorro en recursos como lo es el almacenamiento.

Aprovechando lo que estas tecnologías ofrecen, se puede profundizar un poco más en la implementación de herramientas que mantengan los servicios web sin interrupciones, es por eso que en este proyecto se busca reactivar el servicio de alta disponibilidad para la plataforma Meiweb haciendo uso de contenedores.

*Trabajo de grado

**Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Manuel Guillermo Florez Becerra. Magister en informática.

ABSTRACT

TITLE: IMPLEMENTATION OF A PLATFORM BASED ON CONTAINERS FOR THE DEVELOPMENT AND HOSTING OF WEBSITES WITH HIGH AVAILABILITY*

AUTHOR: NESTOR IVAN INFANTE MONSALVE, SERGIO ANDRES DULCEY LIZARAZO**

KEY WORDS: CONTAINERS, HIGH AVAILABILITY, HOSTING SERVICE, PLATFORM, SERVICES.

DESCRIPTION: The constant change in the technologies used for the development and implementation of web pages requires that the platforms that provide the hosting service are also at the forefront of new technologies.

One of the aspects to take into account when implementing a platform is adaptability and scalability, since organizations need the flexibility to use applications in any environment, be it development or production.

As an alternative to the virtualization options with virtual machines, in this project the development of a platform is carried out with which an application encapsulation can be carried out using container technology through the Docker orchestrator tool that allows us to have the libraries necessary for an application to work independently of the libraries that the operating system has on a host machine, which facilitates the migration, updating, maintenance of the applications and also means savings in resources such as storage.

Taking advantage of what these technologies offer, it is possible to delve a little deeper into the implementation of tools that maintain web services without interruptions, which is why this project seeks to reactivate the high availability service for the Meiweb platform using containers.

*Degree work

**Faculty of Physical-Mechanical Engineering. School of Systems and Computer Engineering. Director: Manuel Guillermo Florez Becerra. Master's in Computing.

INTRODUCCIÓN

El internet apareció en los años 60, sin embargo, ha sido hasta los años 90 que se masificó su uso debido a la aparición de la web, y desde ese entonces la web y las tecnologías que son usadas en ese ámbito, han estado en un constante y acelerado cambio.

Una de las grandes ramas de la Ingeniería de Sistemas es el desarrollo de software, y dentro de éste encontramos el desarrollo web que en los últimos años ha tenido una gran importancia ya que cada vez es más necesario para una organización ofrecer a sus clientes y/o empleados un sistema eficiente y seguro según sean las necesidades y requerimientos de dicha organización.

Unas de las preguntas a las que se enfrentan los desarrolladores web en sus proyectos son, ¿Qué tecnologías voy a usar? ¿Cuáles de las tecnologías existentes se ajustan más a lo que quiere y necesita mi cliente o mi organización? ¿Por cuánto tiempo necesito que mi plataforma esté en ejecución?, es por esto que en la Universidad nos enseñan a siempre estar a la vanguardia de las nuevas tecnologías, a actualizarse siempre que la tecnología se actualice.

Es ahí donde radica la necesidad de la universidad de proveer a los estudiantes de herramientas que faciliten al estudiante realizar sus trabajos con las mejores tecnologías posibles, el actual proyecto pretende brindar al estudiante de la asignatura de Programación en la web, la flexibilidad para adaptar su proyecto de clase a la tecnología que cada estudiante decida y crea que es conveniente utilizando una plataforma de servicio hosting para aplicativos web, que también beneficiará a los estudiantes de proyecto de grado para la implementación de su proyecto con tecnologías de empaquetamiento como los contenedores que

permiten distintos lenguajes de programación, frameworks y entornos de ejecución multiplataforma.

Varias organizaciones requieren que sus operaciones sean continuas y no se interrumpan en ninguna circunstancia, es por eso que recurren a soluciones para cuando alguno de los servidores falle, se garantice que el sistema continúe en funcionamiento. El trabajo de grado busca también realizar reingeniería al servicio de alta disponibilidad en el aula virtual MEIWEB, ya que es en esta donde los estudiantes realizan sus actividades académicas remota y presencialmente. La Alta disponibilidad ya se ha implementado en proyectos anteriores, pero debido a la actualización constante de la tecnología, el servicio de alta disponibilidad dejó de prestarse por lo que se hace necesario reactivar y adecuar este servicio para que funcione a modo de prueba de fallos.

1. SOBRE ESTE TRABAJO DE INVESTIGACIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

La escuela de ingeniería de sistemas de la Universidad Industrial de Santander, en donde se encuentra el grupo de investigación GID-CONUSS (Grupo de Investigación y Desarrollo en Computación en la Nube, Seguridad, Servidores y servicios), implementa un servicio hosting para la asignatura programación en la web (PGWEB), que permite al estudiante el desarrollo de sus proyectos de semestre.

Debido a las limitaciones del servicio mencionado anteriormente, los estudiantes poseen pocas alternativas de arquitecturas web para el avance en su formación académica, que imposibilita el libre manejo de los aplicativos, por tal motivo, las nuevas tecnologías no se podrían emplear para el proyecto de trabajo semestral, no obstante, usando un tipo de empaquetamiento lógico, como los contenedores, que puede alojar y ejecutar las aplicaciones en cualquier lugar con cualquier tipo de entorno, podrían los estudiantes beneficiarse de estas últimas tecnologías. Esas mismas facilidades también se pueden utilizar para darle hosting a los proyectos de grado.

Otro servicio ofrecido por el grupo GID-CONUSS, independiente del servicio hosting, es la alta disponibilidad (HA) para sitio web. En la actualidad está desactivado. Para volver reactivar este servicio, se requiere investigar y realizar una reingeniería que permita volver a dejar en funcionamiento en HA estos servicios.

Acorde a la problemática mencionada, es la base de la temática del trabajo de grado que conduce a plantear y mejorar dichos servicios, lo que genera el planteamiento de dos preguntas:

¿Los contenedores son más apropiados en el servicio hosting que las máquinas virtuales?

¿Qué modelos se pueden usar para reactivar el servicio de alta disponibilidad?

1.2. JUSTIFICACIÓN

El manejo de servicio hosting permite a los usuarios obtener una infraestructura para el funcionamiento y almacenamiento de datos y archivos para el sitio web. Los alumnos que cursan la asignatura PGWEB necesitan aplicar los conocimientos y fundamentos teóricos impartidos en clase para realizar la implementación en un sitio web; Un alto porcentaje de estudiantes de proyecto de grado también necesitan realizar su proyecto mediante el desarrollo web; Sin embargo, el servicio actual no permite el uso de nuevas tecnologías, entonces, es necesario implementar un servicio hosting que permita a los estudiantes de la comunidad EISI desarrollar sus proyectos en estas nuevas plataformas.

La alta disponibilidad permite el funcionamiento continuo de los sitios web cuando se presentan fallas en algunos de los servidores, la reactivación del servicio es indispensable para aumentar la confiabilidad del sistema.

1.3. OBJETIVOS

1.3.1. Objetivo general

- Implementar el servicio de alta disponibilidad en los servidores del grupo GIDCONUSS y una plataforma basada en contenedores aplicada a la asignatura de programación en la web (PGWEB) y proyectos de grado.

1.3.2. Objetivos específicos

- Actualizar la plataforma de la máquina virtual hosting para la asignatura PGWEB desde Ubuntu 14.04 LTS a una versión actual LTS compatible.
- Diseñar e implementar una arquitectura de servicio Hosting basado en contenedores y máquinas virtuales para
 - Implementar contenedores para la asignatura PGWEB.
 - Un modelo basado en contenedores para implementar proyectos de grado.
 - Implementar una interfaz orientada al administrador.
- Aplicar reingeniería al modelo de alta disponibilidad en los servidores del grupo GID-CONUSS para la plataforma MEIWEB.
- Documentar y actualizar los manuales de administración y las políticas de seguridad para estos servicios.

2. MARCO DE REFERENCIA

2.1. ESTADO DEL ARTE

El servicio de web hosting y el sistema HA son muy importantes en la actualidad para las empresas y personas independientes, donde a través del tiempo, se han encontrado nuevas formas y modelos para su desarrollo. En la Escuela de Ingeniería de Sistemas e Informática (EISI) en la Universidad Industrial de Santander (UIS) mediante el trabajo de grado con los estudiantes y profesores albergaron estos temas para el implemento de servicios, algunos ejemplos son evidenciados en el grupo GID-CONUSS y las PaaS.

Las plataformas PaaS se basan en el desarrollo de aplicaciones que permiten recursos externos para crear y hospedar esas aplicaciones web en diferentes lenguajes, las cuales se encuentran CloudBees, Engine Yard, Google App Engine, Heroku, Microsoft Windows Azure, Amazon Web Services, OpenShift, etc.

Actualmente existe un servicio con máquina virtual hosting que se utiliza en la asignatura PGWEB con usuarios enjaulados, usando la aplicación FileZilla, la máquina virtual con Ubuntu server 14.04 LTS y un usuario generado en MySQL, donde usa scripts y una dirección del servidor suministrada por la universidad para alojar proyectos de clases, específicamente aplicaciones web. Las funciones de los scripts es agrupar usuarios y asignarles hosting en la web donde solo ellos puedan modificar, además establece un procedimiento de cierre al finalizar el tiempo establecido para el desarrollo de los proyectos y se tiene el superusuario que es el administrador.

En el proyecto Prototipo De Servidor De Alta Disponibilidad Para La Escuela De Ingeniería De Sistemas E Informática¹ se implementó un prototipo de servidor de alta disponibilidad con el sistema operativo GNU / LINUX DEBIAN 5 y la herramienta Heartbeat. Este prototipo permite conectar los servidores web para que interactúen como una sola máquina frente al usuario final, donde si uno falla, el otro servidor asume la carga de trabajo principal, evitando considerablemente las caídas de servidores y el tiempo de restauración.

En el proyecto Instalación, Administración, Configuración E Implementación De Servidores Linux Con Énfasis En El Desarrollo De Un Modelo Administrativo Y La Creación De Un Prototipo De Clúster De Alta Disponibilidad² se utilizó un software para la alta disponibilidad con las herramientas como el sistema operativo GNU / LINUX DEBIAN 6 de 64 bits, los servidores web Apache 2 y Apache 2 Tomcat6, el lenguaje de programación PHP, una máquina virtual de java, un gestor de bases de datos MySQL y gestores de correos; Creando un cluster de alta disponibilidad con dos servidores , varios software libres y una infraestructura de cluster pasivo/activo para el montaje, con políticas de roles se utiliza un proceso administrativo organizado y controlado ofreciendo un servicio en los servidores de la EISI para mitigar los problemas.

En el proyecto Administración, Mantenimiento, Configuración Y Monitoreo De Los Equipos Servidores Del Grupo GID-Conuss Con Énfasis En El Análisis Y Reestructuración De Los Modelos De Alta Disponibilidad Y Computación En La

¹ GRAZT MEDINA,GISELLE y SANTA ARCINIEGAS,ALEJANDRO. Prototipo De Servidor De Alta Disponibilidad Para La Escuela De Ingeniería De Sistemas E Informática. UNIVERSIDAD INDUSTRIAL DE SANTANDER, 2010.

² BARBOSA AYALA,ALEXANDER y MUÑOZ DUARTE,ELKIN DARÍO. Instalación, Administración, Configuración E Implementación De Servidores Linux Con Énfasis En El Desarrollo De Un Modelo Administrativo Y La Creación De Un Prototipo De Clúster De Alta Disponibilidad. . UNIVERSIDAD INDUSTRIAL DE SANTANDER, 2012.

Nube³ se usó las herramientas como el sistema operativo LINUX, Cluster Resource Manager (CRM), DRBD, servidor Apache y MySQL, donde este sistema funciona mediante una política de roles clásicamente conocida como Maestro/Esclavo, es decir que solo uno de los pares está habilitado para correr servicios y atender a las peticiones de los usuarios, y el otro se mantiene en espera, copiando los datos y tomando el control como rol Maestro cuando se detecta una falla en el sistema principal. Se diferencia del servicio anterior en la capa, ya que está directamente sobre la capa de hardware de los equipos miembros del cluster.

En el proyecto Administración De Prototipo Infraestructura De Computación En La Nube Del Grupo Gid-Conuss Con Énfasis En La Implementación De Nuevos Módulos De Openstack⁴, que es el último trabajo implementado dentro de la escuela EISI sobre alta disponibilidad y en los servidores del grupo GID-CONUSS, en la que existe un modelo de computación de la nube utilizando KVM e implementado openstack con módulos principales y Docker, que consta de 4 máquinas virtuales, dos de las cuales tendrán la función de balancear la carga entrante en MEIWEB.

2.2. MARCO TECNOLÓGICO

En esta parte mediante un análisis investigativo sobre los temas a tratar, se definen los términos claves que permitirán el desarrollo de este proyecto:

³ LIZARAZO TORRES,JOHN,EDINSON. ADMINISTRACION, MANTENIMIENTO, CONFIGURACION Y MONITOREO DE LOS EQUIPOS SERVIDORES DEL GRUPO GID-CONUSS CON ENFASIS EN EL ANALISIS Y REESTRUCTURACION DE LOS MODELOS DE ALTA DISPONIBILIDAD Y COMPUTACION EN LA NUBE [Recurso Electronico]. Bucaramanga: Tesis (Ingeniero de Sistemas) - UIS. Escuela de Ingenieria de Sistemas e Informatica, 2013, 2013.

⁴ MARTÍNEZ RANGEL,JULIÁN ANDRÉS y VÁSQUEZ ROMERO,CESAR DAVID. Administración De Prototipo Infraestructura De Computación En La Nube Del Grupo Gid-Conuss Con Énfasis En La Implementación De Nuevos Módulos De Openstack. UNIVERSIDAD INDUSTRIAL DE SANTANDER, 2017.

2.2.1. Computación en la nube Es una tecnología que permite acceder remotamente, de cualquier lugar del mundo y en cualquier momento, a softwares, almacenamiento de archivos y procesamiento de datos a través de Internet, sin la necesidad de conectarse a un ordenador personal o servidor local.⁵

2.2.2. Modelos de servicios

2.2.2.1. PaaS (Plataform-as-a-Service) La plataforma como servicio por sus siglas en inglés es una categoría de computación en la nube en la cual un tercero brinda el hardware y software de soporte de aplicaciones. Esta modalidad de Cloud Computing permite que el usuario gestione sus propias aplicaciones despreocupándose de la infraestructura y la plataforma, ya que el proveedor del servicio aloja en sus instalaciones el hardware y software y ofrece la plataforma como una solución integrada a semejanza de una caja negra.⁶

2.2.2.2. IaaS (Infrastructure-as-a-Service) A diferencia de PaaS, la IaaS ofrece el servicio de hardware a través de internet, esto quiere decir que el usuario administra el sistema operativo, datos, middleware, tiempos de ejecución, todo esto sin preocuparse por la actualización o mantenimiento físico de los servidores.⁷

⁵ GRAPSAS,Tatiana. ¿Qué es cloud computing o computación en la nube? Conoce sobre el término a continuación. Sep 16,. Disponible en: <https://rockcontent.com/es/blog/computacion-en-la-nube/>

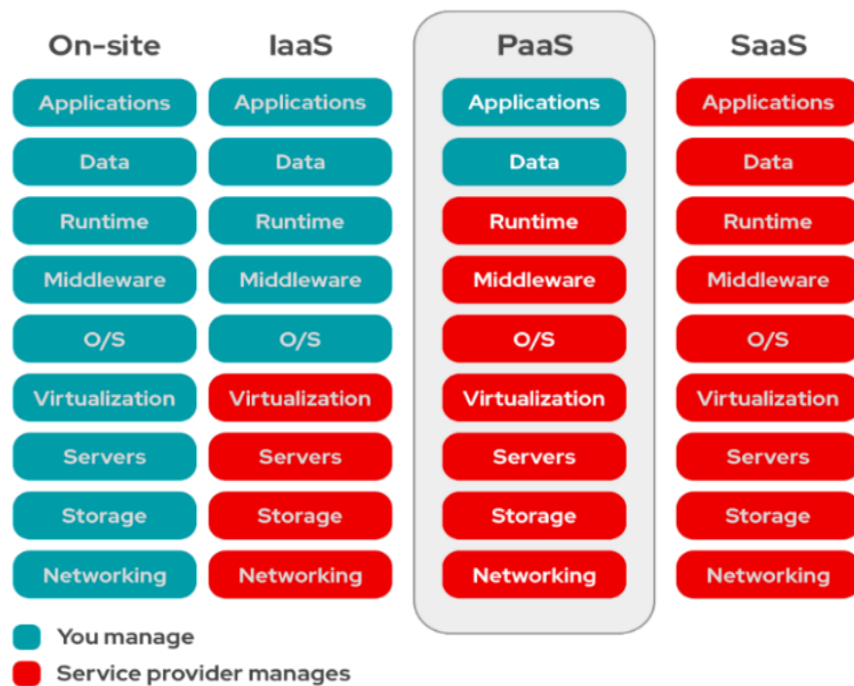
⁶ [Anónimo]¿Qué es una PaaS?. Disponible en: <https://www.redhat.com/es/topics/cloud-computing/what-is-paas>

⁷ BHARDWAJ,Sushil; JAIN,Leena y JAIN,Sandeep. Cloud Computing: A Study of Infrastructure as a Service (IaaS): 2010. p. 60

2.2.2.3. SaaS (Software-as-a-Service) En esta categoría de Cloud Computing, el usuario recibe una aplicación en la nube y junto con ésta, la infraestructura de TI y las plataformas necesarias para el correcto funcionamiento de la aplicación, logrando de esta manera despreocuparse de mantenimientos de infraestructura de TI, plataformas, software, teniendo control únicamente de una personalización mínima.⁸

En la figura 1. se describe gráficamente las diferencias entre PaaS, IaaS y SaaS, y lo que tendría que manejar si se tratara de software convencional sin hacer uso de herramientas que nos ofrece cloud computing.

Figura 1. Modelos de servicios



Fuente: [Anónimo]¿Qué es una PaaS?. Disponible en: <https://www.redhat.com/es/topics/cloud-computing/what-is-paas>

⁸ CUSUMANO, Michael. Technology Strategy and Management : Cloud Computing and SaaS as New Computing Platforms: New York] : Association for Computing Machinery, p. 27-290001-0782

2.2.3. Virtualización Es una tecnología útil para los servicios TI que usa recursos ligados al hardware, también distribuye sus funcionalidades a los usuarios o entornos, lo que permite utilizar toda la capacidad de una máquina física. Es decir, crea un entorno informático virtual generado por un equipo, ayudando a las empresas en particionar un servidor físico en varias máquinas virtuales, esas máquinas virtuales pueden interactuar de forma independiente y ejecutar sistemas operativos mientras comparten recursos.

La virtualización tiene varias ventajas, una de las principales es la mejora de escalabilidad y las cargas de trabajo, mejorando varias condiciones como reducir el consumo de energía, usar menos servidores, costo de infraestructuras, el mantenimiento y la recuperación rápida en caso de fallos. Aunque tiene inconvenientes que pueden afectar al rendimiento, por lo que puede afectar las funcionalidades.⁹

2.2.3.1. Tecnología virtualización KVM Es una máquina virtual basada en el kernel, y es una tecnología de virtualización de código abierto integrada a Linux, su principal característica es ejecutar múltiples máquinas virtuales ejecutando imágenes de Linux o Windows sin modificar. Estas máquinas tienen su hardware virtualizado con distintos parámetros.¹⁰

2.2.3.2. Máquina virtual Una máquina virtual es un software que tiene en su interior un sistema operativo donde imita un PC, existen dos tipos de máquinas virtuales, las de sistemas y las de procesos. El primer tipo es aquel donde emula un

⁹ [Anónimo]Virtualización. Disponible en: <https://www.redhat.com/es/topics/virtualization>

¹⁰ [Anónimo]¿Qué es KVM?. Disponible en: <https://www.redhat.com/es/topics/virtualization/what-is-kvm>

ordenador completo, de tal modo que se puede ejecutar otro sistema operativo en su interior.¹¹

2.2.3.3. Ubuntu server Es un sistema operativo sin entorno gráfico, viable para los computadores con pocas características, todas las acciones se realizan con la consola, y desde una conexión remota. Su manejo es similar a cualquier sistema Linux, pero con otras características particulares como el sudo.

Para usar Ubuntu Server, se utilizará la aplicación VirtualBox que sirve para crear una máquina virtual con este sistema operativo.¹²

2.2.3.4. Contenedores Los contenedores son una herramienta que ofrecen un mecanismo de empaquetado lógico en el que las aplicaciones son independientes del entorno en el que se ejecutan. Esta separación permite que las aplicaciones basadas en contenedores se implementen de manera que puedan ser ejecutadas en cualquier sistema operativo sin importar si se cambia el entorno. La creación de contenedores ofrece una separación entre desarrolladores y administradores de TI, ya que los desarrolladores se enfocan en la lógica y las dependencias de su aplicación, mientras que los administradores de TI se ocupan en la implementación y la administración sin preocuparse por los detalles específicos de cada aplicación, como las versiones del software y las configuraciones de la app.

Usualmente los contenedores son comparados con las máquinas virtuales sin embargo su funcionamiento es muy diferente ya que cada máquina virtual requiere de un sistema operativo junto con el software necesitado, mientras que los contenedores sólo necesitan del software, bibliotecas y dependencias necesarias y

¹¹ RAMÍREZ, Iván. Máquinas virtuales: qué son, cómo funcionan y cómo utilizarlas. Disponible en: <https://www.xataka.com/especiales/maquinas-virtuales-que-son-como-funcionan-y-como-utilizarlas>

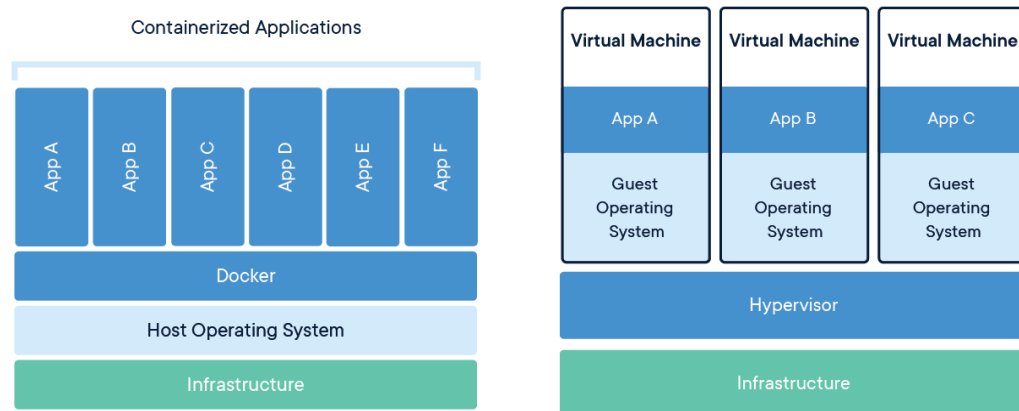
¹² [Anónimo]Ubuntu Server | Ubuntu Fácil.

se pueden llevar al sistema operativo que más convenga. Esto en la práctica significa que hacer funcionar una aplicación empaquetada dentro de un contenedor requiere menos recursos que si se tratase de una máquina virtual.¹³

2.2.4. Docker Docker es un proyecto de código abierto que permite crear contenedores ligeros y portables de manera que aplicaciones se puedan empaquetar junto con las librerías y dependencias y ejecutar en cualquier entorno independiente del sistema operativo.

En la Figura 2. se aprecian las diferencias entre contenedores haciendo uso de Docker y una máquina virtual.

Figura 2. Ilustración de un contenedor Docker y una Máquina Virtual



Fuente: [Anónimo]What is a Container? - Docker. Disponible en: <https://www.docker.com/resources/what-container/>

¹³ [Anónimo]What is a Container? - Docker. Disponible en: <https://www.docker.com/resources/what-container/>

2.2.4.1. Docker hub Es un espacio de nube donde los usuarios pueden subir las imágenes Docker y tenerlas disponibles a usar, esta cuenta con 10.000 aplicaciones gratuitas tanto de imágenes oficiales, como de los mismos usuarios.

Se usa en los archivos yml de docker compose con “image”.¹⁴

2.2.4.2. Docker compose Es una herramienta para definir y ejecutar aplicaciones Docker de contenedores múltiples, que utiliza un archivo YAML para su configuración.

Se necesita el Dockerfile para que pueda reproducirse en cualquier lugar.¹⁵

2.2.4.3. Dockerfile Es un archivo de texto plano que contiene una serie de instrucciones necesarias para crear una imagen que será utilizado para determinado propósito, en algunas ocasiones, este archivo puede ser añadido en las imágenes encontradas en Docker Hub.¹⁶

2.2.5. Web service Es un software con un formato basado en texto que funciona con Internet. Este sistema se encarga de permitir la transmisión de solicitudes y respuestas entre diferentes servidores o aplicaciones, sin importar las diferencias que existan entre los lenguajes de programación en el que fueron desarrolladas o la plataforma en la que se ejecutan.¹⁷

Como ejemplo son Apache y Nginx.

¹⁴ [Anónimo]¿Qué es Docker Hub? – El Camino del Sysadmin. Junio 4,. Disponible en: https://blog.carreralinux.com.ar/2020/06/docker_hub/

¹⁵ [Anónimo]Serie Docker: ¿Qué es Docker Compose? – I.E.S. Celia Viñas – Ciberseguridad. Disponible en: <https://iescelia.org/ciberseguridad/serie-docker-que-es-docker-compose/>

¹⁶ [Anónimo]Qué es DockerFile. Disponible en: <https://openwebinars.net/blog/que-es-dockerfile/>

¹⁷ [Anónimo] ¿Qué es un web service y cómo funciona?. Disponible en: <https://www.crehana.com/blog/desarrollo-web/que-es-web-service/>

2.2.5.1. Apache HTTP Server Es un servidor web, que mediante esa herramienta todas las páginas web y material multimedia forman parte de la infraestructura de las aplicaciones móviles y servicios cloud.

De esta manera, cuando en un navegador se escribe una dirección web, genera una petición HTTP a un servidor remoto, ese servidor web responde a esa petición con la página web a visualizar. Apache trabaja con módulos, pero tiene un pobre rendimiento a otras alternativas como Nginx.¹⁸

Se puede usar en Linux, Windows y CentOS.

2.2.5.2. Nginx Un servidor web de código abierto lanzado en 2004, y es uno de los más usados por las personas desarrolladoras incluso en compañías de alto perfil, desde su éxito ahora también es usado como proxy inverso, cache de HTTP y balanceador de carga.¹⁹

2.2.5.3. Protocolo de transferencia Ambos son protocolos de comunicación que se evidencian cuando se realiza búsquedas por internet, HTTP (Hypertext Transfer Protocol) es un protocolo de transferencia de información de internet y HTTPS (Hypertext Transfer Protocol Secure) surge por la necesidad de proteger la información, la seguridad.

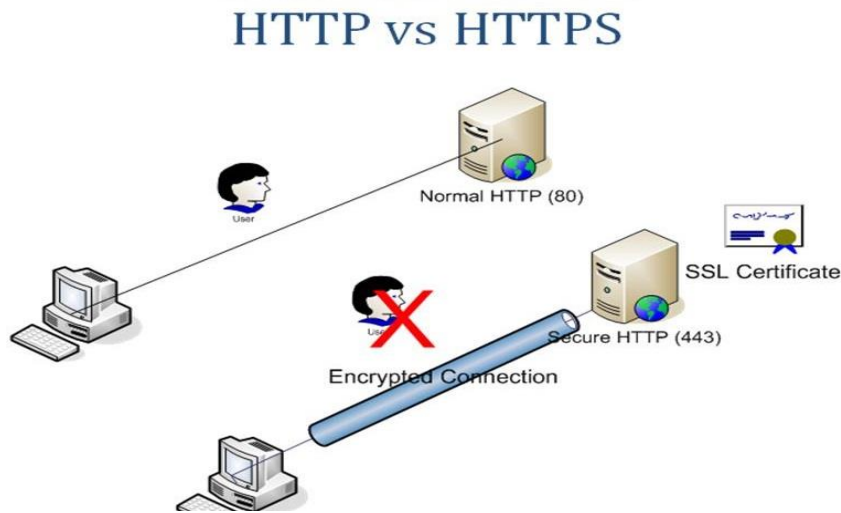
Para tener HTTPS se necesita tener instalado Certificado SSL en el servidor web.²⁰

¹⁸ LEÓN,Álvaro De. ¿Qué es el Servidor Apache? ¿Para qué sirve? Guía de Instalación fácil. Disponible en: <https://blog.infranetworking.com/que-es-apache-servidor/>

¹⁹ [Anónimo]¿Qué Es Nginx y Cómo Funciona?. Disponible en: <https://kinsta.com/es/base-de-conocimiento/que-es-nginx/>

²⁰ ACIBEIRO,María. ¿Cuál es la diferencia entre HTTP y HTTPS?. Disponible en: <https://es.godaddy.com/blog/diferencia-entre-http-y-https/>

Figura 3. Diferencia entre HTTP vs HTTPS



Fuente: RODRIGUEZ,Victoria. Diferencias entre HTTP y HTTPS. Julio 26,. Disponible en: <https://www.segurisoft.es/enciptacion/diferencia-http-https/>

2.2.6. Shell Este término se refiere a aquellos programas que proveen una interfaz de usuario para acceder a los servicios del sistema operativo, que pueden ser gráficos o de texto simple. Los Shells cumplen su función de facilitar la forma en que se ejecutan los distintos programas disponibles en un computador.

El método que usa es la comunicación con el sistema operativo que es de forma interactiva o como un script. El script de Shell es una secuencia de mandatos del Shell y del sistema operativo que se almacena en el archivo.²¹

2.2.7. Filezilla Es una aplicación FTP libre y de código abierto que consta de un cliente FTP y un servidor FTP, convirtiéndolo en uno de los más populares para gestionar y acceder a archivos.

²¹ [Anónimo]IBM Documentation. Disponible en: <https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/es/aix/7.2?topic=administration-operating-system-shells>

Los protocolos que maneja esta aplicación es el FTP (File Transfer Protocol) para compartir archivos y SFTP (SSH File Transfer Protocol) que permite operaciones sobre archivos remotos. El uso más común que usan en FileZilla es para conectarse al hosting de una página web y subir archivos, pero se necesitan datos como la dirección del servidor, el usuario y la contraseña.²²

2.2.8. Cliente SSH PuTTY Es una terminal de simulación open source que fue desarrollado para actuar como cliente de conexiones seguras a través de protocolos raw TCP, Telnet, rlogin y portal serial.

Por lo tanto, este software se indica para establecer conexiones seguras de acceso remoto a servidores a través de Shell Seguro (SSH) para construir canales encriptados entre servidores.

De esta manera, PuTTY fue desarrollado para el uso, principalmente de programadores y administradores de red, ya que su interfaz es prácticamente configurable y cuenta con numerosas opciones de ajuste de conexiones.²³

2.2.9. Fail2ban Es una aplicación en Linux que permite evitar accesos no autorizados al servidor, funciona con el bloqueo o baneando las ip, eso ocurre cuando hacen vario intentos de acceso incorrecto a un servidor, es una estrategia ideal de seguridad.²⁴

²² RAMÍREZ,Iván. Filezilla: qué es, para que sirve y primeros pasos con este cliente de FTP. Sep 19,. Disponible en: <https://www.xataka.com/basics/filezilla-que-sirve-primeros-pasos-este-cliente-ftp>

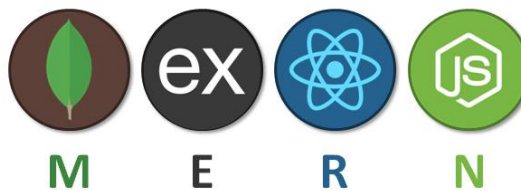
²³ [Anónimo]PuTTY en programación, aprende qué es y cómo utilizarlo. Disponible en: <https://www.hostgator.mx/blog/putty-en-programacion-aprende-que-es/>

²⁴ LEÓN,Manuel. Fail2ban: instalar y configurar para prevenir ataques | Arsys. Disponible en: <https://www.arsys.es/blog/instalar-fail2ban>

2.2.10. Framework Es una estructura que puede ser usada para un proyecto, es requerido por un desarrollador web que produzca el software o la aplicación, un ejemplo de Framework son MERN STACK y MEAN STACK, también conocida como pila tecnológica.²⁵

2.2.10.1. Estructura MERN MERN (MongoDB, Express, React, Node.js)

Figura 4. Framework MERN STACK.



Fuente: [Anónimo]How to learn MERN Stack in 2022? Books, Courses, and Projects. Disponible en: <https://medium.com/javarevisited/how-to-learn-mern-stack-from-scratch-1784618eaffa>

2.2.10.2. Estructura MEAN MEAN (MongoDB, Express, Angular, Node.js).

Figura 5. Framework MEAN STACK.



Fuente: MISHRA,Bipin. MEAN Stack Vs MERN Stack: Which is a Better Choice to Build Your Project?. Disponible en: <https://www.mindinventory.com/blog/mean-stack-vs-mern-stack/>

2.2.11. Pila de desarrollo web Una pila de software incorpora solo los componentes de software necesarios para ejecutar una aplicación, algunos de los

²⁵ [Anónimo]Framework: ¿qué es y cuál es su función en Internet? [1]. Disponible en: <https://rockcontent.com/es/blog/framework/>

cuales podrían ser virtualización o recursos de hardware físico-abstractos. Una pila de tecnología incluye componentes de software y hardware de la infraestructura de TI, como ejemplos se tiene LAMP y LEMP.²⁶

2.2.11.1. Pila LAMP LAMP (Linux, Apache, MySQL, PHP).

Figura 6. Pila de desarrollo LAMP STACK.



Fuente: [Anónimo]LAMP Stack - Web Development & Technology Resources. Disponible en: <https://codecondo.com/wp-content/uploads/2017/09/LAMP-Stack.jpg>

2.2.11.2. Pila LEMP LEMP (Linux, Nginx, MySQL, PHP)

Figura 7. Pila de desarrollo LEMP STACK.



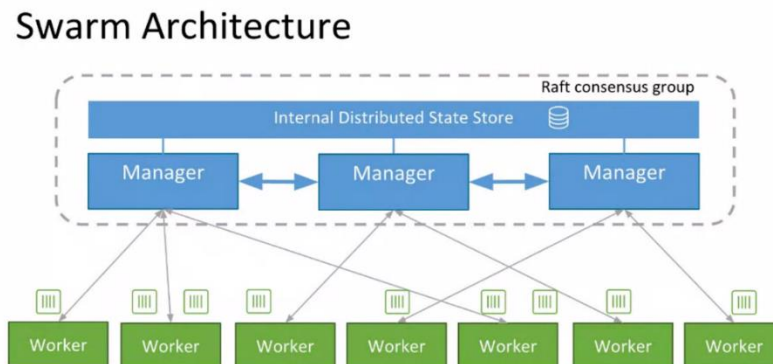
Fuente: [Anónimo]How to Install Nginx, Mysql, PHP (LEMP) Server on Ubuntu 16.04. Disponible en: <https://www.cloudjojo.com/how-to-install-nginx-mysql-php-lemp-server-on-ubuntu-16-04/>

2.2.12. Docker swarm Swarm es un software creado por los programadores de Docker que permite agrupar una serie de hosts de Docker en un clúster y gestionar

²⁶ [Anónimo] ¿Qué es una pila de software? - Krypton Solid. Disponible en: <https://kryptonsolid.com/que-es-una-pila-de-software/>

los clústeres de forma centralizada, así como orquestar los contenedores. Aunque hasta la versión 1.11 de Docker había que implementar Swarm como una herramienta aparte, las versiones posteriores de la plataforma de contenedores soportan un modo nativo swarm (también conocido como modo enjambre), por lo que cuando los usuarios instalan el Docker engine, también tienen a su disposición este gestor de clústeres, se le conoce como “orquestador de contenedores”.²⁷ Swarm implementa el algoritmo de consenso RAFT en los nodos manager.

Figura 8. Arquitectura de Docker Swarm.



Fuente: [Anónimo]Kubernetes vs Docker Swarm:Cúal es el mejor orquestador de containers?. Disponible en: <https://www.nubersia.com/es/blog/kubernetes-vs-docker-swarm/>

2.2.12.1. HAProxy Es un equilibrador de carga TCP y HTTP de código abierto de alta disponibilidad, es utilizada en gran medida en todo tipo de servidores Linux con buena seguridad.

Es considerado una herramienta de bajo coste para el balanceo de cargas en los servidores de datos.²⁸

²⁷ [Anónimo]Docker Compose y Docker Swarm: orquestación de contenedores Docker. Disponible en: <https://www.ionos.es/digitalguide/servidores/know-how/docker-compose-y-swarm-gestion-multicontenedor/>

²⁸ VELASCO,Rubén. HAProxy, un balanceador de cargas TCP/HTTP gratuito. Disponible en: <https://www.redeszone.net/2014/10/25/haproxy-un-balanceador-de-cargas-tcphttp-gratuito/>

2.2.13. Alta disponibilidad Para hablar de alta disponibilidad tenemos primero que definir disponibilidad, la cual es la capacidad del sistema de permitir el ingreso de usuarios y realizar cambios, consultas, gestionar trabajos, etc. Si un sistema no permite lo anterior, se dice que no está disponible.

Sabiendo lo anterior podemos definir alta disponibilidad como la capacidad de asegurar cierto grado de continuidad de la operación de un sistema durante un periodo de medición.²⁹

La implementación del sistema de alta disponibilidad necesita de un sistema de archivos, base de datos que puede ser distribuido como MariaDB, Un servidor web como Apache y otras aplicaciones dependiendo del modelo que se desee aplicar.³⁰

²⁹ [Anónimo]Alta disponibilidad - ¿Qué es y cómo funciona? Disponible en: <https://ciberseguridad.com/guias/alta-disponibilidad/>

³⁰ [Anónimo]IBM Documentation. Disponible en: <https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/es/cdfsp/7.6.1.2?topic=systems-high-availability-models>

3. METODOLOGIA

En el desarrollo del proyecto, se propone distintas fases en la metodología para obtener el cumplimiento de los objetivos específicos.

Fase 1:

En la primera fase se llevará a cabo una investigación sobre la información del proyecto para comprender su funcionamiento, las actividades a realizar son:

Actividad 1: Analizar alternativas de herramientas a utilizar en el proyecto haciendo uso de trabajos de grado previos sobre el tema, artículos, documentación, etc.

Actividad 2: Decidir las herramientas a utilizar en la fase de implementación del proyecto

Fase 2:

En esta fase se comenzará con la implementación del proyecto haciendo una actualización de la plataforma actual de la máquina virtual hosting para la asignatura PGWEB, que se encuentra en Ubuntu Server 14.04 LTS a una versión reciente LTS compatible. Las actividades por realizar son:

Actividad 1: Hacer una copia de seguridad de la máquina virtual que presta el servicio actualmente en caso de cualquier eventualidad.

Actividad 2: Actualizar los scripts de la máquina virtual a una versión reciente LTS compatible.

Actividad 3: Comprobar el correcto funcionamiento de la plataforma en el nuevo sistema operativo.

Actividad 4: Actualizar las políticas de seguridad SLA.

Fase 3:

Después de analizar la información e identificar las herramientas a usar, se realizará la etapa del desarrollo de la plataforma de servicio hosting usando contenedores con las siguientes actividades a realizar:

Actividad 1: Hacer el análisis de la anterior plataforma evidenciada en la fase 2 y de los scripts usados en ese prototipo funcional del grupo GID-CONUSS.

Actividad 2: Diseñar la arquitectura de la plataforma.

Actividad 3: Crear prototipo a nivel local (en los computadores personales) para posterior implementación en los servidores.

Actividad 4: Implementar el prototipo en las máquinas servidores del grupo GID-CONUSS.

Actividad 5: Comprobar el funcionamiento del servicio hosting con los proyectos de la asignatura PGWEB y aplicativos webs de proyectos de grado.

Actividad 6: Se desarrollará la interfaz del servicio hosting orientada al administrador para su sencilla utilización.

Actividad 7: Actualizar los manuales de administración y políticas de seguridad.

Fase 4:

Realización del prototipo del modelo de alta disponibilidad para implementación en la plataforma MEIWEB usando Docker Swarm, las actividades a realizar son:

Actividad 1: Analizar detalladamente el modelo anterior implementado en la plataforma MEIWEB.

Actividad 2: Aplicar reingeniería al servicio de alta disponibilidad.

Actividad 3: Comprobar el correcto funcionamiento simulando fallos en una máquina.

Actividad 4: Actualizar y documentar administración remota para el mantenimiento de los servidores CONUSS

Fase 5:

Se realizará la digitación del libro de proyecto de grado.

4. DESARROLLO DEL PROYECTO

Se lleva a cabo un análisis a la infraestructura del grupo GID-CONUSS y una investigación sobre la implementación de la plataforma PAAS y el servicio de alta disponibilidad, donde se obtiene que:

1. El grupo GID-CONUSS cuenta con tres servidores DELL POWEREDGE cada uno con Ubuntu 14.04 LTS
2. La máquina virtual de la plataforma hosting de PGWEB se encuentra desactualizada y sin un menú para desplegar las funciones.
3. Los antiguos proyectos de alta disponibilidad del grupo GID-CONUSS están desactualizados y complejos en su desarrollo.
4. La falta de manuales de proyectos anteriores y la implementación de estos en el proyecto.
5. La necesidad de implementar contenedores con Docker, y dejar un trabajo a futuro para utilizar Kubernetes.

4.1. SITUACIÓN ACTUAL

Los requerimientos para usar en los servicios de acuerdo con lo que se tiene de infraestructura en el grupo GID-CONUSS son limitados y se tuvo que hacer sus respectivos cambios para lograr los objetivos del proyecto.

Figura 9. Setup del grupo GID-CONUSS



Tabla 1. Especificaciones del servidor Larusso.

Marca	Dell
Referencia	PowerEdge R720
Almacenamiento	2 HDD de 1TB Non-RAID
Procesador	INTEL XEON Six-core at 2.00GHz
RAM	32GB

Tabla 2. Especificaciones del servidor Smith.

Marca	Dell
Referencia	PowerEdge R710
Almacenamiento	2 HDD de 1TB en configuración RAID-1
Procesador	INTEL XEON Quad-core at 2.66GHz
RAM	24GB

Tabla 3. Especificaciones del servidor Johnson

Marca	Dell
Referencia	PowerEdge R710
Almacenamiento	2 HDD de 1TB en configuración RAID-1
Procesador	INTEL XEON Quad-core at 2.66GHz
RAM	24GB

Esos cambios se efectuaron de la siguiente manera:

4.1.1. Actualización de los servidores En el grupo GID-CONUSS se tiene tres (3) servidores donde se han trabajado la mayoría de los proyectos y se encuentra alojado MEIWEB,

- Se verifica los archivos que se encuentra dentro de los tres servidores (Larusso, Smith y Johnson), donde se evidencia archivos de proyectos de grados anteriores y una versión antigua de Ubuntu Server LTS.

Se actualiza a 22.04 LTS la versión de los servidores, lamentablemente al ser una versión reciente no se tiene una documentación amplia para desarrollar adecuadamente el proyecto, es allí donde se utiliza 20.04 LTS.

Con el objetivo de lograr esa actualización, se formatea los servidores y eso conlleva a la escogencia de archivos a los cuales se realiza una copia de seguridad para guardarlos, y es allí donde se decide solo formatear dos servidores (a uno de esos se le instala interfaz gráfica de Ubuntu) y en el tercer servidor utilizar máquinas virtuales con Ubuntu Server 20.04 LTS, aunque se tenga el servidor con Ubuntu Server 14.04 LTS usando Virtual Machine Manager (VMM) desde otra computadora.

Los servidores que se formatearon fueron Larusso y Smith, con la diferencia que el servidor Smith es con interfaz gráfica Gnome para usar VMM y Teamviewer.

- El almacenamiento que tendrá los dos servidores que son formateados con Ubuntu Server 20.04 LTS debe ser fundamental para abarcar cualquier proyecto o herramienta sin ningún problema.

Figura 10. Particiones dinámicas del servidor Larusso y Smith.

```

conusse@lactoria:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO  TYPE MOUNTPOINT
loop0       7:0      0   48M  1  loop /snap/snapd/16778
loop1       7:1      0   47M  1  loop /snap/snapd/16292
loop2       7:2      0  63,2M 1  loop /snap/core20/1623
loop3       7:3      0  67,2M 1  loop /snap/lxd/21835
loop4       7:4      0  67,8M 1  loop /snap/lxd/22753
loop5       7:5      0   62M  1  loop /snap/core20/1611
sda         8:0      0  931G  0  disk
├── sda1     8:1      0   1,1G  0  part /boot/efi
├── sda2     8:2      0    1G  0  part
├── ┌vg0-boot 253:7    0 1020M  0  lvm  /boot
├── └sda3     8:3      0   600G  0  part
├── ┌vg1-libvirt 253:4    0 600G  0  lvm  /var/lib/libvirt
├── └sda4     8:4      0   200G  0  part
├── ┌vg2-backup 253:0    0 200G  0  lvm  /backup
├── └sda5     8:5      0   129G  0  part
├── ┌vg3-usr   253:1    0   15G  0  lvm  /usr
├── ┌vg3-tmp   253:2    0   10G  0  lvm  /tmp
├── ┌vg3-var   253:3    0   40G  0  lvm  /var
├── ┌vg3-swap  253:5    0   10G  0  lvm  [SWAP]
├── └vg3-raiz  253:6    0    54G  0  lvm  /
sr0         11:0     1 1024M  0  rom
sr1         11:1     1 1024M  0  rom

```

- Se configura Netplan para colocarle la IP privada a cada servidor y de esta manera quede configurada su red dentro de la universidad.

Figura 11. Ubicación de la carpeta netplan

```

root@lactoria:/etc/netplan# ls
00-installer-config.yaml

```

Figura 12. Configuración una IP privada dinámica.

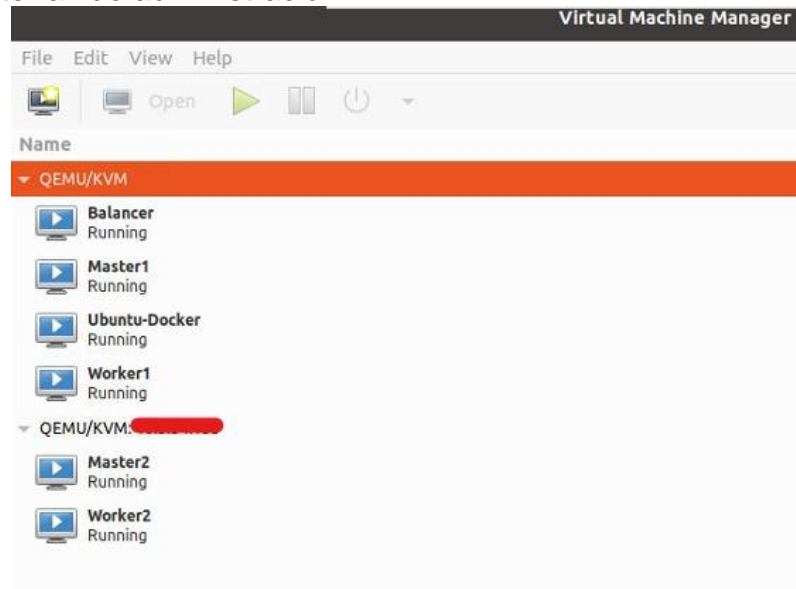
```
GNU nano 4.8                                00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  renderer: networkd
  ethernets:
    eno1:
      dhcp4: no

    eno2:
      dhcp4: no
    eno3:
      dhcp4: no
    eno4:
      dhcp4: true

  bridges:
    br0:
      interfaces: [eno4]
      dhcp4: true
  version :2
```

- Se coloca una interfaz gráfica Gnome en el servidor Smith para usar VMM tanto en los dos servidores.

Figura 13. Interfaz de administración VMM



- Se crea las máquinas virtuales con Ubuntu server 20.04 LTS necesarias para cada servicio y se administra con VMM en los siguientes servidores:

Tabla 4. Servidor Smith

Servidor: Smith
Máquinas Virtuales:
➤ Máquina virtual Master1: Nodo maestro 1.
➤ Máquina virtual Worker1: Nodo Trabajador 1.
➤ Máquina virtual Balancer: Balanceador de cargas.
➤ Máquina virtual Docker_Pruebas: Pruebas Docker.

Tabla 5. Servidor Larusso

Servidor: Larusso
Máquinas Virtuales:
➤ Máquina virtual Master2: Nodo maestro 2
➤ Máquina virtual Worker2: Nodo trabajador 2.

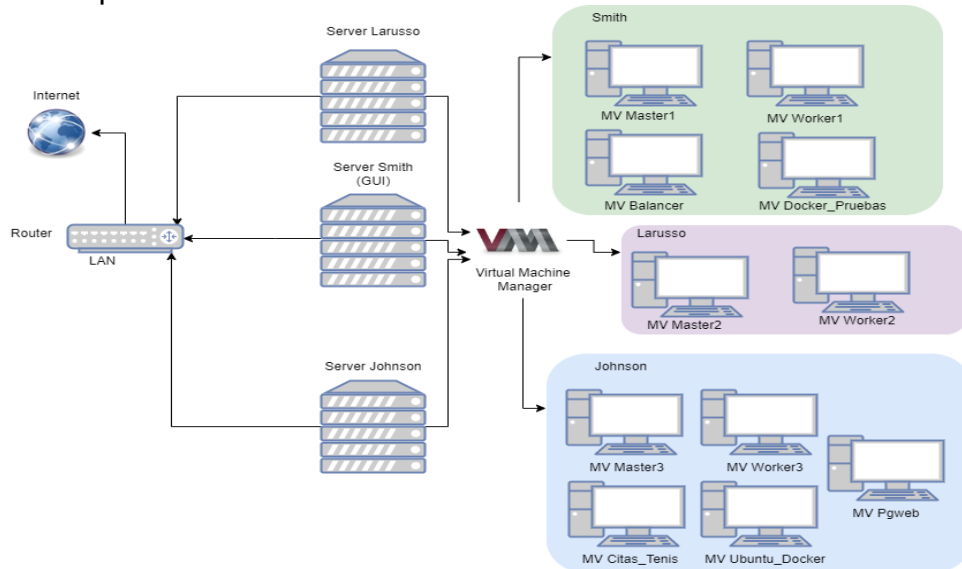
Tabla 6. Servidor Johnson

Servidor: Johnson
Máquinas virtuales:
➤ Máquina virtual Master3: Nodo maestro 3.
➤ Máquina virtual Worker3: Nodo trabajador 3.
➤ Máquina virtual Citas_Tenis: MERN
➤ Máquina virtual Ubuntu_Docker: Servicio hosting con contenedores Docker
➤ Máquina virtual Pgweb: Usuarios enjaulados LAMP

- Se guarda lo que se realizó en un manual que explica con detalles la información para el administrador.

4.2. ESQUEMA GENERAL DE LA INFRAESTRUCTURA

Figura 14. Esquema actual



Fuente: Elaboración propia

En la figura 14. se observa el esquema de la infraestructura a usar para realizar el proyecto, en el que usaremos las máquinas virtuales con Ubuntu Server 20.04 LTS donde se implementará Docker y Docker Swarm.

Esas MV tienen herramientas instaladas que se adecuan al funcionamiento del proyecto con los servicios implementados en el grupo GID-CONUSS.

5. IMPLEMENTACIÓN

Para el proyecto se analiza tres definiciones importantes como usuarios enjaulados, Docker y Docker Swarm, dando como resultado el desarrollo de estos tres servicios para el grupo GID-CONUSS y sus respectivos manuales.

5.1. MÁQUINA VIRTUAL CON USUARIOS ENJAULADOS:

De antecedentes se tiene el modelo PAAS implementado por el proyecto de grado titulado Prototipo De Un Servicio De Computación En La Nube PAAS (Plataforma Como Servicio) Y Web Service Orientado Al Desarrollo De Programación En La Web³¹, donde realizo una plataforma de servicio hosting funcional en la asignatura “programación en la web” en el desarrollo del programa académico por varios años, que permitió a los estudiantes subir archivos y modificar la página web a realizar como proyecto final usando herramientas como:

- LAMP (LINUX, APACHE, MYSQL/MARIADB, PHP5.6, PHPMYADMIN).
- FileZilla.
- PuTTY (SSH).
- Fail2ban, usado para bloquear conexiones remotas que deseen entrar sin permiso por un límite de intentos usando SSH.
- Sconly, usado como guía para usuarios enjaulados.

Este servicio fue implementado en Ubuntu Server 14.04 LTS, por lo que se realiza una actualización y modificación a este software, analizando que ya existe Ubuntu Server 20.04 LTS para manejar este nuevo procedimiento.

³¹ GARRIDO CRUZ, PAOLA, ANDREA. PROTOTIPO DE UN SERVICIO DE COMPUTACION EN LA NUBE PAAS (PLATAFORMA COMO SERVICIO) Y WEB SERVICE ORIENTADO AL DESARROLLO DE PROGRAMACION WEB [Recurso Electronico]. Bucaramanga: Tesis (Ingeniero de Sistemas) - UIS. Escuela de Ingenieria de Sistemas, 2017, 2017.

La implementación de la nueva plataforma comenzó con definir que especificaciones se coloca a la MV, para que pueda abarcar un peso adecuado en el almacenamiento de los aplicativos creados por los estudiantes de clase.

Tabla 7. Especificaciones de la máquina virtual.

Especificaciones de la MV	<ul style="list-style-type: none"> • RAM: 2048 MB • 20 GB Disco duro • Tamaño Dinámico • Ubuntu Server 20.04 LTS
Primera partición	Boot 512MB
Segunda partición	19.496 GB
Tercera partición	Bios_grup 1MB
LVM (Volúmenes lógicos)	<ul style="list-style-type: none"> • /var: 3.0GB • /usr: 6.0GB • /home: 6.0GB • /: 4,496 GB

Cuando ya se crea la MV, se le instala todas las herramientas mencionadas anteriormente con algunos cambios como PHP, donde se puede usar cualquier versión tanto en el sistema como en apache usando los comandos y un menú que maneje los scripts Shell creados para este servicio.

Figura 15. Versión PHP.

```

Selection  Path                Priority  Status
-----
* 0        /usr/bin/php7.4    74       auto mode
  1        /usr/bin/php7.2    72       manual mode
  2        /usr/bin/php7.4    74       manual mode

Press <enter> to keep the current choice[*], or type selection number:

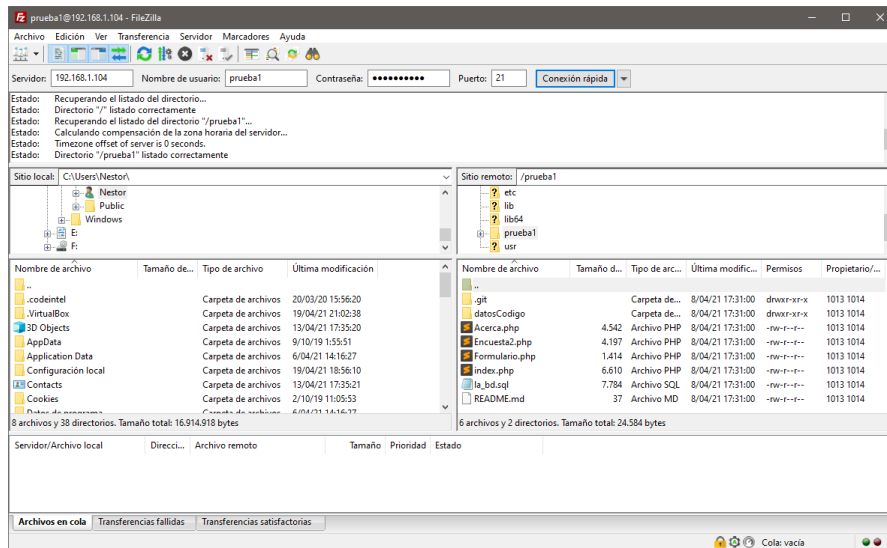
```

Figura 16. Menú del administrador.

```
root@actuweb:/home/administrador# bash /root/setup_chroot/menu.sh
Seleccione la acción que desea realizar:
1) Crear usuario          6) Cerrar materia      11) Backup usuario
2) Mostrar usuarios enjaulados 7) Abrir usuario      12) Backup semestre
3) Generar BD           8) Abrir materia      13) Backup total
4) Enviar correo usuarios 9) Eliminar Usuario   14) Salir
5) Cerrar usuario       10) Eliminar Usuario y archivos
#?
```

Como se observa en la figura 16. el menú solamente será habilitado para el administrador para seleccionar cualquier acción, donde tendrá a su vista el modo de uso, y de esta manera puede crear usuarios que posteriormente pueden ser usados en la herramienta FileZilla, a fin de que los estudiantes puedan editar y subir los archivos.

Figura 17. FileZilla



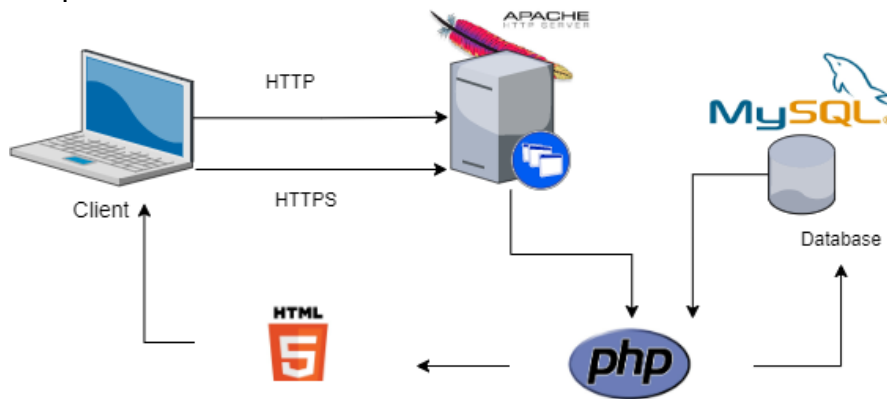
Así mismo una herramienta como phpMyAdmin para evidenciar las bases de datos que maneja la administración MySQL en las páginas web creadas por los estudiantes.

Figura 18. phpMyAdmin



La MV tiene una IP pública y privada donde es agregada por medio de netplan para configurar redes. El servidor para usar es apache que usa http y https en una página como en la figura 19.

Figura 19. Arquitectura LAMP.



Fuente: Elaboración propia

5.2. USUARIOS ENJAULADOS CON CONTENEDORES DOCKER

De la parte anterior lleva al desarrollo de la siguiente que es con contenedores Docker, que facilita el manejo de otros entornos de desarrollo en aplicativos webs tanto para Backend como Frontend, donde lo esencial es la diversidad de opciones que puedan usar los estudiantes de proyecto de grado y de clase.

Este servicio usa herramientas y archivos como:

- FileZilla
- PuTTY (SSH)
- Fail2ban, usado para bloquear conexiones remotas que deseen entrar sin permiso por un límite de intentos usando SSH.
- Sponly, usado como guía para usuarios enjaulados.
- Herramienta Docker
- Herramienta Docker Compose
- Archivo Dockerfile
- Archivo .env, para cambiar los puertos de los contenedores.
- Nginx como servidor web/proxy inverso para los contenedores
- Dialog, usado para crear el menu

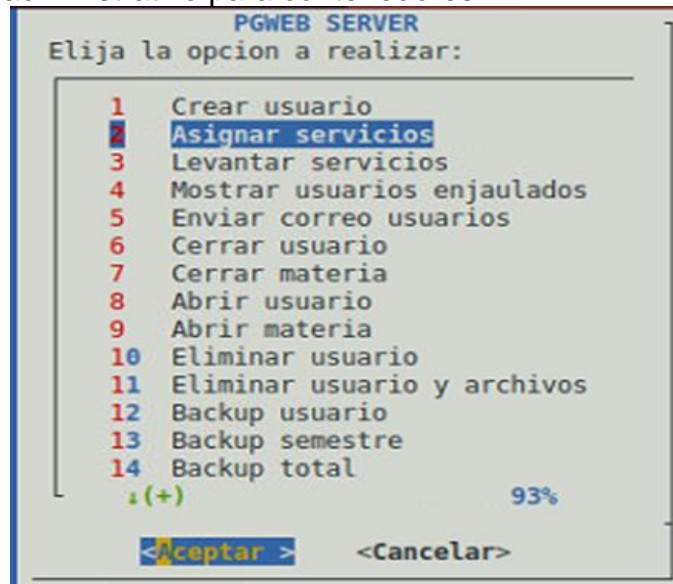
La MV a usar es Ubuntu server 20.04 LTS, donde se instala las herramientas mencionadas anteriormente. Se vuelve a usar nuevamente programas como FileZilla (cliente FTP) y PuTTY SSH, para los estudiantes y administradores respectivamente.

Tabla 8. Máquina virtual con Docker y Docker Compose.

Especificaciones de la MV	<ul style="list-style-type: none"> • RAM: 2048 MB • 128 GB Disco duro • Tamaño Dinámico • Ubuntu Server 20.04 LTS
Primera partición	Boot 512MB
Segunda partición	127.5GB
Tercera partición	Bios_grup 1MB
LVM (Volúmenes lógicos)	<ul style="list-style-type: none"> • /var: 60GB • /usr: 12GB • /home: 48GB • /: 7.5GB

Las especificaciones de la máquina virtual como se evidencia en la tabla 8. se hizo por el gran tamaño que tiene las imágenes del repositorio Docker Hub.

Figura 20. Menú administrativo para contenedores

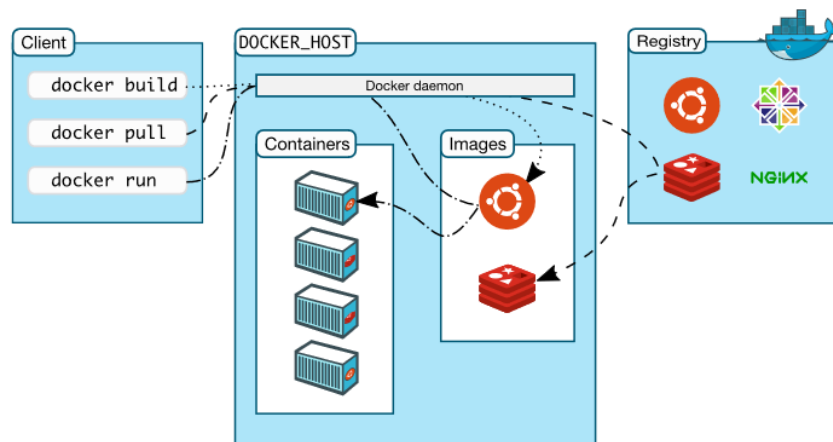


En la figura 20. se observa el menú creado para el administrador del servicio hosting con contenedores, donde se usa las opciones para emplear ese servicio a los

estudiantes, de tal forma que puedan usar contenedores ya creados como MERN STACK, MEAN STACK y LEMP, en dado caso que deseen usar otros programas y herramientas, se le va a proveer de una MV para ser usada en VirtualBox y puedan empezar por ellos mismos a aprender de Docker.

Este servicio es manejado con IP privada dentro de la universidad usando netplan y su esquema se evidencia en la figura 21.

Figura 21. Arquitectura con Docker.



Fuente: [Anónimo]Arquitectura de Docker. Disponible en: https://jsitech1.gitbooks.io/meet-docker/content/arquitectura_de_docker.html

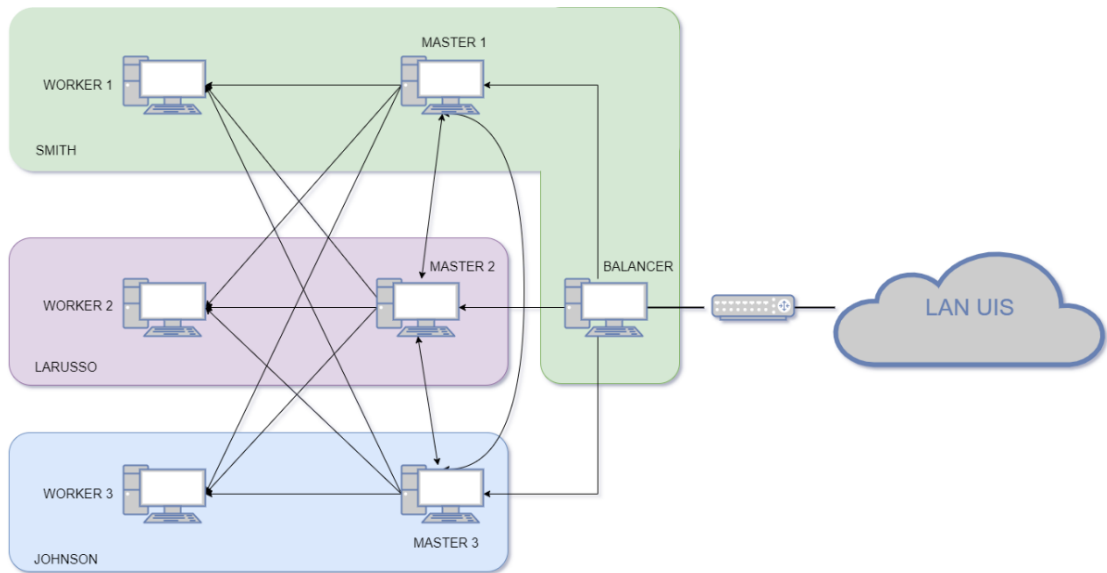
5.3. ALTA DISPONIBILIDAD USANDO DOCKER SWARM

El uso de alta disponibilidad es importante para mantener la conectividad, por lo que se implementó HA en el aplicativo web MEIWEB usando Docker Swarm, que es un software basado en la arquitectura maestro-esclavo.

Tabla 9. Especificaciones MV utilizadas en alta disponibilidad

Especificaciones de la MV	<ul style="list-style-type: none"> • RAM: 2048 MB • 20 GB Disco duro • Tamaño Dinámico • Ubuntu Server 20.04 LTS
Primera partición	Boot 512MB
Segunda partición	19.5GB
Tercera partición	Bios_grup 1MB
LVM (Volúmenes lógicos)	<ul style="list-style-type: none"> • /var: 8.0GB • /home: 4.0GB • /: 7.5GB

Figura 22. Arquitectura Docker Swarm (Cluster)



Fuente: Elaboración propia

En la Figura 22. se describe visualmente el clúster utilizado para el servicio de Meiweb en configuración de Alta Disponibilidad, en el cual se hizo uso de 3 máquinas físicas con 7 máquinas virtuales. La máquina Balancer cuenta con un servidor HAProxy, el cual es el encargado de realizar el balanceo de cargas, el

servidor recibe una solicitud y la entrega al nodo que tenga la función de Leader en ese momento y el nodo Leader se encargará de distribuir la solicitud entre los nodos restantes. En el caso que el nodo Leader se caiga, habrá otros 2 nodos Master esperando para tomar la posición de Leader y de esta manera continuar prestando el servicio sin interrupciones por lo cual el usuario final no notará los cambios que se realicen dentro del clúster.

5.3.1. Comprobación de prerequisites Lo primero antes de iniciar la configuración de los clusters, es comprobar que todas las máquinas virtuales tienen instalado Docker y que están en la misma versión. Para comprobar esto se usa el siguiente comando. (Para ejecutar cualquier comando de docker, se debe estar logueado como *root*)

```
docker --version
```

También se debe comprobar que el servicio de Docker está activo, Para esta comprobación se ejecuta el siguiente comando:

```
service docker status
```

En dado caso que no esté activo, se usa el siguiente comando:

```
service docker start
```

5.3.2. Crear cluster

5.3.2.1. Crear el primer nodo El primer nodo que se creará será Master1, para la creación del primer nodo es necesario conocer la dirección IP de la máquina en la que quedará alojado el nodo, para esto se utiliza el comando

```
ip -a
```

Conociendo la dirección IP, se une este nodo al nuevo cluster con el siguiente comando:

```
docker swarm init --advertise-addr <IP del Master1>
```

En esa respuesta, Docker sugiere los siguientes comandos que se van a ejecutar ya que se debe agregar 2 nodos manager y 3 nodos worker al clúster.

Para agregar los nodos manager:

```
docker swarm join-token manager
```

La respuesta de este comando se tendrá que copiar y pegar directamente en la consola de comandos del Master2 y del Master3, para este caso, no interesa conocer la dirección IP de estos dos nodos ya que Docker Swarm se encarga de identificarlos y agregarlos al clúster.

Ahora se agregan los nodos worker con el otro comando sugerido en la respuesta de la creación de Master1:

```
docker swarm join --token SWMTKN-1-2pqdrf8vky7vs7fzmxq06kaigua75g62  
kp694tk7l4cwi1f0sn-0drgutgbpz4aofpq2h7m5tmzw 10.3.54.36:2377
```

Este comando al igual que con los manager se tendrá que pegar en cada una de las terminales de los nodos a agregar como worker.

Terminado de ejecutar los comandos en esas terminales se procede a verificar que cada nodo fue agregado al cluster con éxito, para esto se ejecuta el siguiente comando en cualquiera de los nodos manager (master1, master2, master3):

docker node ls

El resultado que debemos ver es algo como esto:

Figura 23. Nodos

```
root@master1:/home/conuss# docker node ls
ID                HOSTNAME  STATUS  AVAILABILITY  MANAGER STATUS  ENGINE VERSION
ko4u2j3mzzj6qe9xm4rz2brqj *  master1  Ready   Active        Reachable       20.10.18
zr6krx2yp7r6hq27n53rifx44  master2  Ready   Active        Reachable       20.10.18
9w0ruz64by7tuddcdhcao6wdg  master3  Ready   Active        Leader          20.10.18
n1xt04rozcvskb0pdmqbyzgk4  worker1  Ready   Active        Reachable       20.10.18
yovibta6ox6qxr5b7yu6hpbq9  worker2  Ready   Active        Reachable       20.10.18
0cxhdd3e68xckjmvjyunuqykt  worker3  Ready   Active        Reachable       20.10.18
root@master1:/home/conuss#
```

En el cual se debe comprobar que todas las máquinas se encuentren en estado activo y que la ENGINE VERSION sea la misma en todos los nodos.

Ya que Docker en modo Swarm implementa en los nodos Manager el algoritmo de consenso RAFT³², se recomienda tener un número impar de nodos Manager siendo 3 el mínimo para garantizar la tolerancia a fallos³³ y tantos Worker como desee. Para este proyecto la topología propuesta es 3 Manager y 3 Worker debido a las limitaciones físicas y a que el entorno no es crítico.

³² [Anónimo]Raft consensus in swarm mode. Disponible en: <https://docs.docker.com/engine/swarm/raft/>

³³ [Anónimo]Administer and maintain a swarm of Docker Engines. Disponible en: https://docs.docker.com/engine/swarm/admin_guide/

Ubicados en el contexto de Meiweb, Definidos los puertos a utilizar en el archivo .env, se procede a desplegar el stack de Swarm, con el siguiente comando:

```
docker stack deploy --compose-file docker-compose.yml meiweb
```

Luego se comprueba que el stack fue levantado exitosamente y que todos los servicios convergieron a las réplicas definidas. Esto se puede comprobar con el comando

```
docker service ls
```

Se procede a escalar cada servicio con el comando

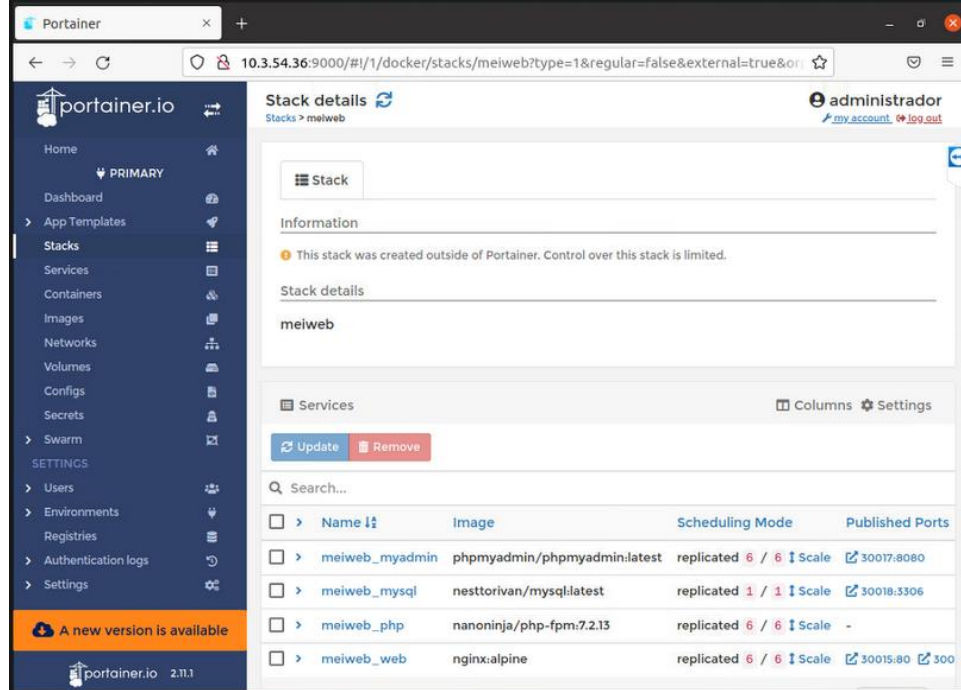
```
docker service scale meiweb_(*)=(**)
```

donde (*) es el servicio para escalar y (**) es el número de replicas que buscamos y finalmente revisamos nuevamente el estado de los servicios.

Por último, se despliega el stack de Portainer³⁴, una herramienta de visualización y administración del clúster Swarm, una aplicación parte de parte código abierto que, aunque sus herramientas son mucho más enfocadas a la creación y mantenimiento del clúster, para efectos de este proyecto se va a utilizar como una herramienta de visualizar gráficamente los contenedores en ejecución y el estado actual del clúster.

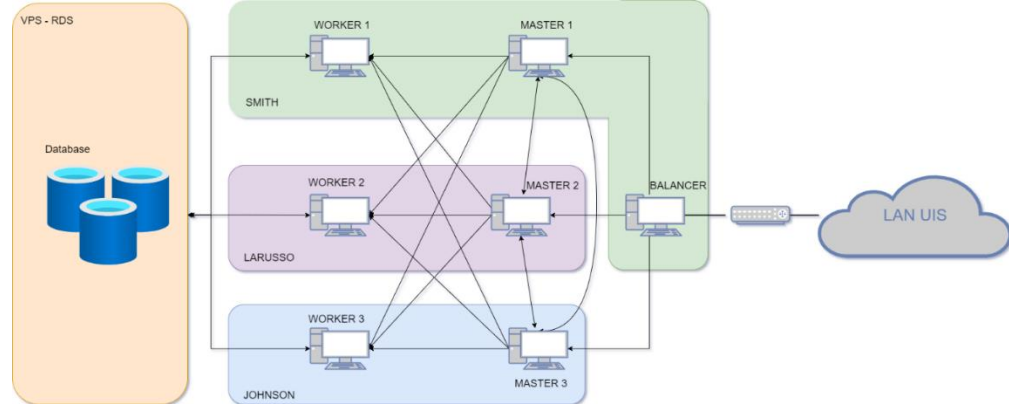
³⁴ [Anónimo] Docker and Kubernetes Management | Portainer. Disponible en: <https://www.portainer.io>

Figura 24. Stack de Meiweb en Portainer



Un problema encontrado en la implementación del servicio tiene que ver con el escalado de los contenedores de las bases de datos, ya que en el orquestador Swarm e inclusive Kubernetes no se permite el escalamiento de bases de datos, es por lo que se recomienda como trabajo futuro implementar la base de datos en una VPS rentada que permita el escalamiento de la base de datos, de manera que el servicio se preste sin interrupciones, de manera que la arquitectura del cluster quede de la siguiente manera:

Figura 25. Arquitectura deseada



Fuente: Elaboración propia

5.4. ACCESO A LA APLICACIÓN DESDE EL EXTERIOR

Para visualizar cualquier aplicación web con contenedores en el exterior la mejor forma es rentar un servicio, aunque se tiene una alternativa llamada “Self-Hosting”, las siguientes formas se pueden usar tanto para Docker y Docker Swarm.

- Usando el archivo YAML de docker compose, esta forma es obteniendo una ip pública, y usando contenedores se puede redireccionar a esa IP, existen varias imágenes que tiene sus herramientas para llevar a cabo ese proceso, pero las que tienen mayor documentación son Nginx y Traefik³⁵.
- Proveedores o Plataformas de alojamiento Docker, Es necesario pagar para los beneficios que ofrecen, y esos mismos servicios CaaS o PaaS, manejan sus servidores por lo que es más sencillo y adecuado. En el caso de Heroku, que también es de alojamiento de aplicaciones, maneja contenedores, pero de una forma distinta como las imágenes, es decir se debe analizar que va a cambiar el formato de uso, por eso es necesario buscar la palabra específica “Alojamiento Docker”.

³⁵ [Anónimo]Traefik Proxy Documentation - Traefik. Disponible en: <https://doc.traefik.io/traefik/>

- Usando Self-Hosting, en este es por medio de servidor propio y solo se puede usar si la persona administradora puede gestionar³⁶

5.5. MANUALES DE USO

Los Manuales creados explican la forma como se crearon los servicios y la manera de usarlos por parte del administrador como soporte, junto con los de uso en dado caso que el estudiante desee tener una guía que le explique. Estos documentos estarán guardados en la base de datos de archivos del grupo GID-CONUSS.

Se plantea un SLA para el uso de los servicios para la comunidad de la Universidad Industrial de Santander, especialmente los estudiantes de proyecto de grado y los que necesiten usar ese servicio por proyectos de clase.

Figura 26. Términos y condiciones

Bienvenido al servidor GID-CONUSS

Términos y condiciones:

Anexo A: Términos y Condiciones de Uso [2021/04/05]

Este acuerdo establece los términos y condiciones de uso aplicables para usted (el "USUARIO") en su uso de los servicios prestados (los "SERVICIOS") por el Grupo de Investigación y Desarrollo en Computación en la Nube, Seguridad, Servidores y Servicios ("GID-CONUSS") para la plataforma de infraestructura como servicio (IaaS) de computación en la nube cloudEISi. Al usar hacer uso de los SERVICIOS, el USUARIO automáticamente acepta estos Términos y Condiciones de Uso y acuerda registrarse por ellos.

GID-CONUSS es una instancia comprometida con los usuarios que además de proveer SERVICIOS, vela por su calidad y fidelidad. Para ello institucionalizó los **TERMINOS Y CONDICIONES DE USO** los cuales determinan y garantizan el debido proceso para utilizar información personal y privada del USUARIO que hace uso del SERVICIO para no infringir de ninguna manera las disposiciones contenidas en la Ley Estatutaria del Habeas Data (Ley 1581 del 2012), reglamentada mediante el decreto 1377 del 2013 y en la Ley de Privacidad o Habeas Data Colombiana (Ley 1266 de 2008).

Es interés de GID-CONUSS la protección de la privacidad de la información personal del USUARIO obtenida a través del SERVICIO, comprometiéndose a adoptar una política de confidencialidad.

1. Usuarios elegibles.

Los servicios están abiertos para el uso de cualquier miembro de la comunidad académica de la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander.

2. Cuenta de usuario, contraseña y seguridad.

Como parte de la creación de una cuenta de usuario se otorgará un nombre de cuenta y una contraseña. El USUARIO es completamente responsable por mantener la confidencialidad de su cuenta o contraseña. Si el USUARIO tiene conocimiento de que la privacidad de su contraseña ha quedado expuesta, debe notificar inmediatamente a GID-CONUSS e informar acerca de cualquier fallo de seguridad.

3. Contenidos del usuario.

El USUARIO acepta usar los SERVICIOS sólo para fines legales y queda prohibida la publicación, distribución y alojamiento en los SERVICIOS de:

(a) cualquier material ilegal, perjudicial, amenazante, abusivo, acosador, difamatorio u obsceno de cualquier tipo, incluyendo, entre otros, cualquier material que fomente conductas que constituirían una ofensa penal, que produjera responsabilidad civil o que de otro modo infrinja cualquier ley aplicable local, estatal, nacional o internacional.

(b) del cual el USUARIO no tenga el derecho de poner a disposición conforme a ninguna ley o relación contractual; o

(c) infrinja cualquier derecho de patente, marca comercial, secreto de mercado, derecho de autor u otros derechos de propiedad de cualquier parte. El USUARIO acepta no transmitir, subir, publicar, enviar por email o poner a disposición de otro modo virus informáticos, publicidad no solicitada o no autorizada, material promocional o de venta, lo que incluye cadenas de correspondencia, correos masivos o cualquier otra forma de "spam".

El USUARIO acepta NO

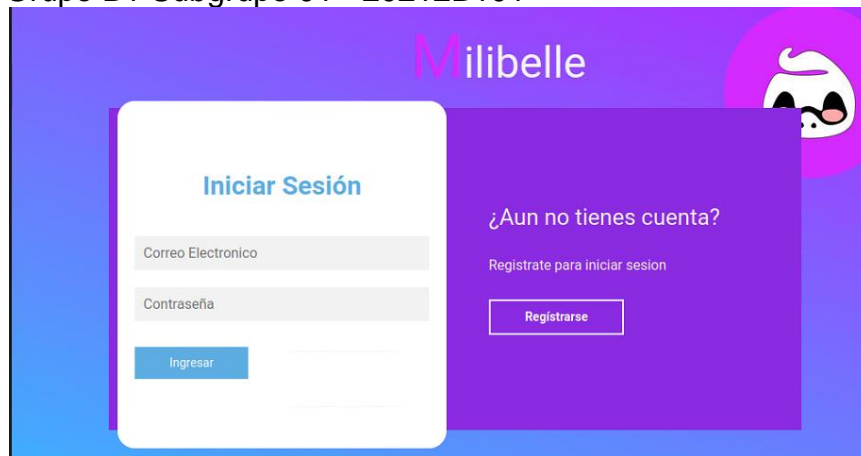
³⁶ ROWDEN, Dan. What is self-hosting? · Gloat. Disponible en: <https://gloat.dev/what-is-self-hosting/>

6. RESULTADOS

6.1. PROYECTOS DE ESTUDIANTES

La máquina virtual que usa usuarios enjaulados con LAMP fue elaborado para que los estudiantes pudieran subir los proyectos, en este caso, la nueva MV fue implementada para las clases de 2021-2 dando evidencia del funcionamiento y ejecución de los proyectos de los estudiantes.

Figura 27. Grupo B1 Subgrupo 01 - 20212B101



Nota: Pagina elaborado por el grupo B1 Subgrupo 01 del semestre 2021-2

Figura 28. Grupo B1 Subgrupo 02 - 20212B102



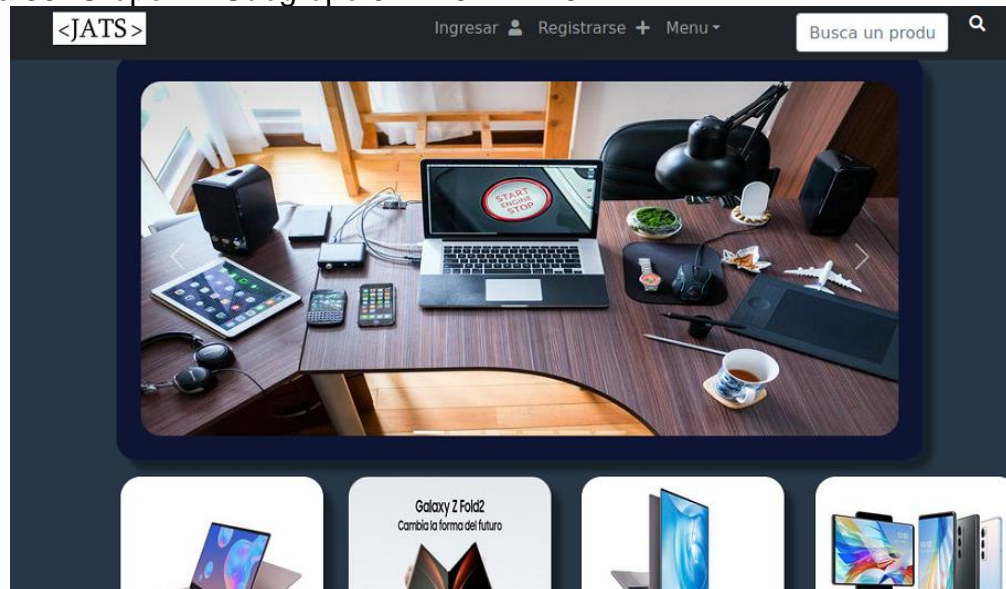
Nota: Pagina elaborado por el grupo B1 Subgrupo 02 del semestre 2021-2

Figura 29. Grupo B1 Subgrupo 03 - 20212B103



Nota: Pagina elaborado por el grupo B1 Subgrupo 03 del semestre 2021-2

Figura 30. Grupo B1 Subgrupo 04 - 20212B104



Nota: Pagina elaborado por el grupo B1 Subgrupo 04 del semestre 2021-2

6.2. PRUEBA CONTENEDORES DOCKER

Las pruebas se hicieron con el desarrollo de los contenedores de proyecto de grado por parte de estudiantes, usando MERN STACK, se dockeriza el aplicativo web que se realiza en el proyecto de grado entregado en 2022-1 titulado "Prototipo de curso para la asignatura de "Programación en la web" y la implementación de un aplicativo web PWA para programación de citas y eventos como recurso didáctico".

Figura 31. Contenedores ejecutados para MERN STACK.

```
root@container-server:/home/administrador# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED
STATUS        PORTS
NAMES
605ce55dd3a3  nginx:alpine                        "/docker-entrypoint..." 3 weeks ag
o Up 2 days   0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443-
>443/tcp      web
e5df7192aae4  citagendaf:prod                      "npm start"              3 weeks ag
o Up 2 days   0.0.0.0:3000->3000/tcp, :::3000->3000/tcp
citagenda_front_1
10d94269885e  citagendab:prod                      "pm2-runtime ./proce..." 3 weeks ag
o Up 2 days   0.0.0.0:4000->4000/tcp, :::4000->4000/tcp
citagenda_back_1
5b089e20ee0b  mongo:4.4.13                         "docker-entrypoint.s..." 3 weeks ag
o Up 2 days   0.0.0.0:27017->27017/tcp, :::27017->27017/tcp
citagenda_database_1
ac78c48163c8  mongo-express:1.0.0-alpha.4         "tini -- /docker-ent..." 3 weeks ag
o Up 2 days   0.0.0.0:8081->8081/tcp, :::8081->8081/tcp
citagenda_mongo-ui_1
```

Figura 32. Página de club unión.



De esta manera, los estudiantes que vayan a usar una estructura MERN, usaran el mismo contenedor para su fácil uso.

6.3. ALTA DISPONIBILIDAD A MEIWEB

Se hizo las pruebas con Docker para ver el funcionamiento de la página MEIWEB, como se evidencia en la figura 33.

Figura 33. Contenedores del aplicativo MEIWEB usando LEMP.

```
root@container-server:/home/administrador# docker ps
CONTAINER ID   IMAGE                                COMMAND                                CREATED
STATUS        PORTS
NAMES
747f635b5717   phpmyadmin/phpmyadmin              "/docker-entrypoint..."            7 days ago
Up 2 days     0.0.0.0:8080->80/tcp, :::8080->80/tcp
meiwebestudiantes_myadmin_1
af33af56d4a4   nginx:alpine                       "/docker-entrypoint..."            7 days ago
Up 2 days     0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443
/tcp
meiwebestudiantes_web_1
3659ba2c00b2   nanoninja/php-fpm:7.2.13          "docker-php-entrypoi..."           7 days ago
Up 2 days     9000/tcp
meiwebestudiantes_php_1
394838d0f3de   mysql:5.7                           "docker-entrypoint.s..."           7 days ago
Up 2 days     0.0.0.0:3306->3306/tcp
meiwebestudiantes_mysql_1
```

Luego se apaga los contenedores Docker y se activa el servicio de Docker Swarm, donde se escala los nodos como se observa en la figura 34.

Figura 34. Nodos listados

```
root@master3:/home/conuss# docker node ls
ID                HOSTNAME        STATUS        AVAILABILITY        MANAGER STATUS        ENGINE VERSION
ko4u2j3mzzj6qe9xm4rz2brqj   master1        Ready        Active              Leader                20.10.18
zr6krx2yp7r6hq27n53rifx44   master2        Ready        Active              Reachable             20.10.18
9w0ruz64by7tuddodhcao6wdg *   master3        Ready        Active              Reachable             20.10.18
nlxt04rozcvskb0pdmqbyzgzk4   worker1        Ready        Active              Reachable             20.10.18
yovibta6ox6qxr5b7yu6hpgqb9   worker2        Ready        Active              Reachable             20.10.18
0cxhdd3e68xckjmvjyynuqykt    worker3        Ready        Active              Reachable             20.10.18
```

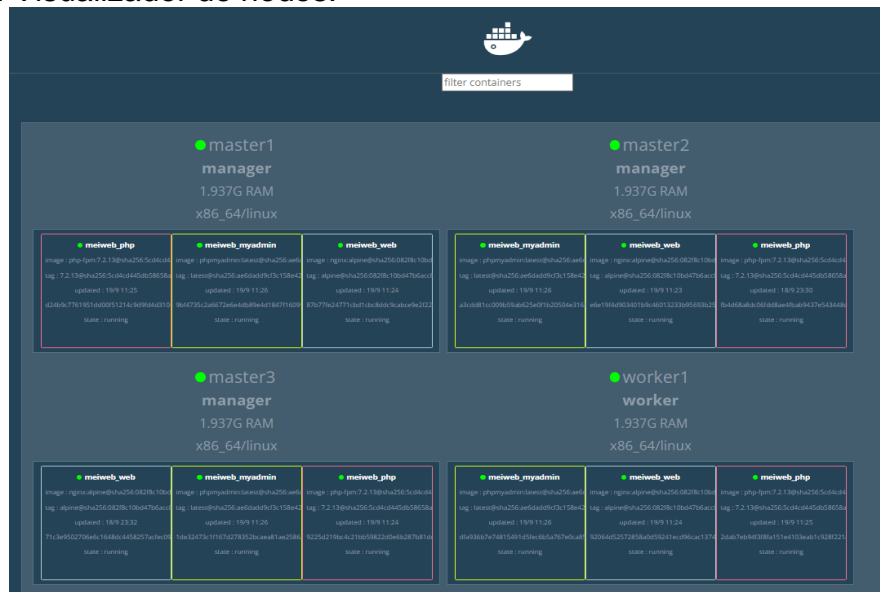
Finalmente se evidencia la página Meiweb en funcionamiento, usando tanto Docker como Docker Swarm.

Figura 35. Pagina Meiweb



Con un visualizador de contenedores, se puede evidenciar gráficamente los nodos y los contenedores en cada uno.

Figura 36. Visualizador de nodos.



6.4. PRUEBAS

Se realiza unas pruebas de funcionalidad a los servicios creados:

Tabla 10. Pruebas al primer servicio usuarios enjaulados

Prueba	Resultado
Iniciar sesión desde PuTTY por parte del administrador	Exitoso
Iniciar sesión desde FileZilla como cliente FTP	Exitoso
Cambiar de versión php a apache	Exitoso
Visualizar el menú y comprobar su funcionalidad	Exitoso
Crear usuarios con sus bases de datos	Exitoso
Administrar la gestión de usuarios con el menú	Exitoso
Subir archivos desde FileZilla	Exitoso
Comprobar la ip pública del servicio	Exitoso
Visualizar la política de seguridad SLA	Exitoso
Visualizar HTTP y HTTPS con apache	Exitoso
Visualizar páginas web LAMP	Exitoso
Visualizar PHPMYADMIN para administrar la base de datos MYSQL	Exitoso
Hacer backups de las páginas web y sus bases de datos	Exitoso

Tabla 11. Pruebas al segundo servicio usuarios enjaulados con contenedores Docker

Prueba	Resultado
Iniciar sesión desde PuTTY por parte del administrador	Exitoso
Iniciar sesión desde FileZilla como cliente FTP	Exitoso
Visualizar el menú y comprobar su funcionalidad	Exitoso
Comprobar la funcionalidad de Docker Compose	Exitoso
Comprobar la funcionalidad de Dockerfile	Exitoso
Comprobar la funcionalidad de un archivo (.env)	Exitoso
Subir archivos desde FileZilla	Exitoso
Visualizar la política de seguridad SLA	Exitoso
Visualizar HTTP y HTTPS con NGINX	Exitoso
Visualizar páginas web MERN STACK y MEAN STACK usando Docker	Exitoso
Visualizar la interfaz de MongoDB	Exitoso
Hacer backups de las páginas web	Exitoso

Tabla 12. Pruebas a MEIWEB con Docker Swarm.

Prueba	Resultado
Iniciar sesión desde PuTTY por parte del administrador	Exitoso
Usar Docker como herramienta para MEIWEB	Exitoso
Desplegar contenedores para un aplicativo LEMP	Exitoso
Cambiar el formato del archivo de Docker Compose para usar Docker Swarm	Exitoso
Crear cluster de docker swarm	Exitoso
Unir Managers al clúster	Exitoso
Unir Workers al clúster	Exitoso
Desplegar el servicio	Exitoso
Escalar el servicio usando la totalidad de los nodos y con redundancia para evitar fallos	Exitoso
Configurar el servidor HA Proxy agregando las direcciones IP de los Managers	Exitoso
Enviar una solicitud al servidor HA Proxy esperando una respuesta afirmativa de la aplicación	Exitoso
Simular fallos apagando las máquinas	Exitoso
Revisar que después del fallo las maquinas se vuelvan a unir al clúster.	Exitoso

7. CONCLUSIONES

Los usuarios enjaulados permiten el manejo de los estudiantes en un servidor FTP para el servicio hosting, dando uso a la gestión y acceso de archivos de aplicaciones web, en este caso con el primer prototipo usando la pila de desarrollo LAMP o con el segundo prototipo basado en contenedores usando la pila tecnológica MERN STACK, MEAN STACK.y LEMP.

El uso de contenedores con la plataforma Docker facilita el desarrollo de distintas arquitecturas de software flexibilizando los entornos de desarrollo, motivando a los estudiantes de la materia Programación en la web a ampliar sus áreas de estudio y a buscar alternativas en el uso de herramientas para la solución de problemas.

Los menús de los servicios hosting y manuales creados para el administrador ayudan como guía en la ejecución del servicio, presentando una gran variedad de documentación en dado caso de un fallo o trabajo futuro.

La implementación de un servicio de alta disponibilidad requiere de una arquitectura compleja de manera que se garantice servicio ininterrumpido, el principal problema de esta arquitectura se encuentra con la base de datos ya que el escalamiento de este servicio no se encuentra disponible en Docker Swarm ni en Kubernetes.

El desarrollo de proyectos enfocado a brindar servicios, administración y mantenimiento de servidores da un valor agregado en la vida de un profesional relacionado con el área TI ya que conoce el porqué de cada situación, de cada restricción y también adquiere habilidades en la resolución de problemas relacionados con las tecnologías que usa.

Linux y en general el software libre brinda un sinnúmero de alternativas y soluciones a problemas que se presentarán en el ámbito laboral, haciendo un buen uso se puede llegar a un gran nivel de personalización de tal manera que sean satisfechas las estrictas demandas que en la actualidad se requiere.

8. RECOMENDACIONES

El proyecto se usó en la red LAN del campus con Docker y Docker Swarm, para usarlo por fuera de esta red, se necesita rentar una instancia EC2* en AWS** o cualquier plataforma que venga con Linux de preferencia. Otra forma es rentar dominios públicos y modificar los Docker compose usando nginx, apache o traefik. Por último, se tiene el uso de Self-Hosting que es el servidor propio.

Los contenedores usados hacen parte de MERN STACK, MEAN STACK, LEMP y LAMP, si el estudiante desea usar otro framework, debe consultar.

Para futuros proyectos, si otros estudiantes desean continuar con este trabajo, necesitan saber que la complejidad aumenta al usar la siguiente etapa llamada Kubernetes, ya que el uso de esta herramienta requiere conocimientos de Docker y Docker Swarm.

Kubernetes aunque sea complicado, favorece mayormente con HA, escalabilidad y balanceo de cargas, junto con la mejora de optimización de recursos y aplicaciones que trabajan en esto conjuntamente.

Para que el servicio de alta disponibilidad sea completamente a prueba de fallos, se recomienda tener la base de datos en un VPS (Virtual Private Server) o RDS (Amazon RDS) que son autoescalable, ya que, Docker en modo Swarm e inclusive Kubernetes no tiene una herramienta que facilite la escalabilidad de las bases de datos

* Elastic Compute Cloud

** Amazon Web Service

BIBLIOGRAFIA

¿Qué es una pila de software? - Krypton Solid. Disponible en: <https://kryptonsolid.com/que-es-una-pila-de-software/>

¿Qué es Docker Hub? – El Camino del Sysadmin. (Junio 4,). Disponible en: https://blog.carreralinux.com.ar/2020/06/docker_hub/

¿Qué es KVM? redhat. (Mar 21,). Disponible en: <https://www.redhat.com/es/topics/virtualization/what-is-KVM>

¿Qué Es Nginx y Cómo Funciona? Kinsta. (Febrero 21,). Disponible en: <https://kinsta.com/es/base-de-conocimiento/que-es-nginx/>

¿Qué es una PaaS? redhat. (Jul 25,). Disponible en: <https://www.redhat.com/es/topics/cloud-computing/what-is-paas>

¿Qué es un web service y cómo funciona?. <https://www.crehana.com>. (Noviembre 11,). Disponible en: <https://www.crehana.com/blog/desarrollo-web/que-es-web-service/>

ACIBEIRO, María. ¿Cuál es la diferencia entre HTTP y HTTPS? Blog. Disponible en: <https://es.godaddy.com/blog/diferencia-entre-http-y-https/>

Administer and maintain a swarm of Docker Engines. Docker Documentation. Disponible en: https://docs.docker.com/engine/swarm/admin_guide/

Alta disponibilidad - ¿Qué es y cómo funciona? Ciberseguridad. Disponible en: <https://ciberseguridad.com/guias/alta-disponibilidad/>

Arquitectura de Docker. Disponible en: https://jsitech1.gitbooks.io/meet-docker/content/arquitectura_de_docker.html

BARBOSA AYALA, ALEXANDER y MUÑOZ DUARTE, ELKIN DARÍO. INSTALACIÓN, ADMINISTRACIÓN, CONFIGURACIÓN E IMPLEMENTACIÓN DE SERVIDORES LINUX CON ÉNFASIS EN EL DESARROLLO DE UN MODELO ADMINISTRATIVO Y LA CREACIÓN DE UN PROTOTIPO DE CLÚSTER DE ALTA DISPONIBILIDAD. UNIVERSIDAD INDUSTRIAL DE SANTANDER, 2012.

BHARDWAJ, Sushil; JAIN, Leena y JAIN, Sandeep. Cloud Computing: A Study of Infrastructure as a Service (IaaS). Cloud Computing: A Study of Infrastructure as a Service (IaaS). 2010. pp. 60

CUSUMANO, Michael. Technology Strategy and Management : Cloud Computing and SaaS as New Computing Platforms. Technology Strategy and Management : Cloud Computing and SaaS as New Computing Platforms. New York] : Association for Computing Machinery, pp. 27-29

Docker and Kubernetes Management | Portainer. Disponible en: <https://www.portainer.io>

Docker Compose y Docker Swarm: orquestación de contenedores Docker. IONOS Digital Guide. Disponible en: <https://www.ionos.es/digitalguide/servidores/know-how/docker-compose-y-swarm-gestion-multicontenedor/>

Framework: ¿qué es y cuál es su función en Internet? Rock Content - ES. Disponible en: <https://rockcontent.com/es/blog/framework/>

GARRIDO CRUZ, PAOLA,ANDREA. PROTOTIPO DE UN SERVICIO DE COMPUTACION EN LA NUBE PAAS (PLATAFORMA COMO SERVICIO) Y WEB SERVICE ORIENTADO AL DESARROLLO DE PROGRAMACION WEB [recurso electronico] Bucaramanga: Tesis (Ingeniero de Sistemas) - UIS. Escuela de Ingenieria de Sistemas, 2017, 2017.

GRAPSAS, Tatiana. ¿Qué es cloud computing o computación en la nube? Conoce sobre el término a continuación. rockcontent. (Sep 16,). Disponible en: <https://rockcontent.com/es/blog/computacion-en-la-nube/>

GRAZT MEDINA, GISELLE y SANTA ARCINIEGAS, ALEJANDRO. PROTOTIPO DE SERVIDOR DE ALTA DISPONIBILIDAD PARA LA ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA. UNIVERSIDAD INDUSTRIAL DE SANTANDER, 2010.

How to Install Nginx, Mysql, PHP (LEMP) Server on Ubuntu 16.04. Cloud JoJo. Disponible en: <https://www.cloudjojo.com/how-to-install-nginx-mysql-php-lemp-server-on-ubuntu-16-04/>

How to learn MERN Stack in 2022? Books, Courses, and Projects. Medium. Disponible en: <https://medium.com/javarevisited/how-to-learn-mern-stack-from-scratch-1784618eaffa>

IBM Documentation. (-03-03). Disponible en: <https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/es/cdfsp/7.6.1.2?topic=systems-high-availability-models>

IBM Documentation. (-05-16). Disponible en: <https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/es/aix/7.2?topic=administration-operating-system-shells>

Kubernetes vs Docker Swarm: Cúal es el mejor orquestador de containers? Nubersia. Disponible en: <https://www.nubersia.com/es/blog/kubernetes-vs-docker-swarm/>

LAMP Stack - Web Development & Technology Resources. (Septiembre 5,). Disponible en: <https://codecondo.com/wp-content/uploads/2017/09/LAMP-Stack.jpg>

LEÓN, Álvaro De. ¿Qué es el Servidor Apache? ¿Para qué sirve? Guía de Instalación fácil. Infranetworking. Disponible en: <https://blog.infranetworking.com/que-es-apache-servidor/>

LEÓN, Manuel. Fail2ban: instalar y configurar para prevenir ataques | Arsys. Blog de arsys.es. Disponible en: <https://www.arsys.es/blog/instalar-fail2ban>

LIZARAZO TORRES, JOHN, EDINSON. ADMINISTRACION, MANTENIMIENTO, CONFIGURACION Y MONITOREO DE LOS EQUIPOS SERVIDORES DEL GRUPO GID-CONUSS CON ENFASIS EN EL ANALISIS Y REESTRUCTURACION DE LOS MODELOS DE ALTA DISPONIBILIDAD Y COMPUTACION EN LA NUBE [recurso electronico] Bucaramanga: Tesis (Ingeniero de Sistemas) - UIS. Escuela de Ingenieria de Sistemas e Informatica, 2013, 2013.

MARTÍNEZ RANGEL, JULIÁN ANDRÉS y VÁSQUEZ ROMERO, CESAR DAVID. ADMINISTRACIÓN DE PROTOTIPO INFRAESTRUCTURA DE COMPUTACIÓN EN LA NUBE DEL GRUPO GID-CONUSS CON ÉNFASIS EN LA IMPLEMENTACIÓN DE NUEVOS MÓDULOS DE OPENSTACK. UNIVERSIDAD INDUSTRIAL DE SANTANDER, 2017.

MISHRA, Bipin. MEAN Stack Vs MERN Stack: Which is a Better Choice to Build Your Project? Disponible en: <https://www.mindinventory.com/blog/mean-stack-vs-mern-stack/>

PuTTY en programación, aprende qué es y cómo utilizarlo. (Abril 20,). Disponible en: <https://www.hostgator.mx/blog/putty-en-programacion-aprende-que-es/>

Qué es DockerFile. OpenWebinars.net. Disponible en: <https://openwebinars.net/blog/que-es-dockerfile/>

Raft consensus in swarm mode. Docker Documentation. Disponible en: <https://docs.docker.com/engine/swarm/raft/>

RAMÍREZ, Iván. Máquinas virtuales: qué son, cómo funcionan y cómo utilizarlas. Xataka. Disponible en: <https://www.xataka.com/especiales/maquinas-virtuales-que-son-como-funcionan-y-como-utilizarlas>

RAMÍREZ, Iván. Filezilla: qué es, para que sirve y primeros pasos con este cliente de FTP. (Sep 19,). Disponible en: <https://www.xataka.com/basics/filezilla-que-sirve-primeros-pasos-este-cliente-ftp>

RODRIGUEZ, Victoria. Diferencias entre HTTP y HTTPS. (Julio 26,). Disponible en: <https://www.segurisoft.es/enciptacion/diferencia-http-https/>

ROWDEN, Dan. What is self-hosting? - Gloat. Disponible en: <https://gloat.dev/what-is-self-hosting/>

Serie Docker: ¿Qué es Docker Compose? – I.E.S. Celia Viñas – Ciberseguridad. Disponible en: <https://iescelia.org/ciberseguridad/serie-docker-que-es-docker-compose/>

Traefik Proxy Documentation - Traefik. Disponible en: <https://doc.traefik.io/traefik/>

Ubuntu Server | Ubuntu Fácil. 2013,

VELASCO, Rubén. HAProxy, un balanceador de cargas TCP/HTTP gratuito. RedesZone. Disponible en: <https://www.redeszone.net/2014/10/25/haproxy-un-balanceador-de-cargas-tcphttp-gratuito/>

Virtualización. redhat. (Mar 19,). Disponible en: <https://www.redhat.com/es/topics/virtualization>

What is a Container? - Docker. Disponible en: <https://www.docker.com/resources/what-container/>

ANEXOS

Anexo A. SLA – Termino y condiciones de uso

Bienvenido al servidor GID-CONUSS

Términos y condiciones:

Anexo A: Términos y Condiciones de Uso [2021/04/05]

Este acuerdo establece los términos y condiciones de uso aplicables para usted (el "**USUARIO**") en su uso de los servicios prestados (los "**SERVICIOS**") por el Grupo de Investigación y Desarrollo en Computación en la Nube, Seguridad, Servidores y Servicios ("**GID-CONUSS**") para la plataforma de infraestructura como servicio (IaaS) de computación en la nube cloudEISI. Al usar hacer uso de los **SERVICIOS**, el **USUARIO** automáticamente acepta estos Términos y Condiciones de Uso y acuerda regirse por ellos. **GID-CONUSS** es una instancia comprometida con los usuarios que además de proveer **SERVICIOS**, vela por su calidad y fidelidad. Para ello institucionalizó los **TERMINOS Y CONDICIONES DE USO**, los cuales determinan y garantizan el debido proceso para utilizar información personal y privada del **USUARIO** que hace uso del **SERVICIO** para no infringir de ninguna manera las disposiciones contenidas en la Ley Estatutaria del Habeas Data (Ley 1581 del 2012), reglamentada mediante el decreto 1377 del 2013 y en la Ley de Privacidad o Habeas Data Colombiana (Ley 1266 de 2008).

Es interés de **GID-CONUSS** la protección de la privacidad de la información personal del **USUARIO** obtenida a través del **SERVICIO**, comprometiéndose a adoptar una política de confidencialidad.

1. Usuarios elegibles.

Los servicios están abiertos para el uso de cualquier miembro de la comunidad académica de la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander.

2. Cuenta de usuario, contraseña y seguridad.

Como parte de la creación de una cuenta de usuario se otorgará un nombre de cuenta y una contraseña. El **USUARIO** es completamente responsable por mantener la confidencialidad de su cuenta o contraseña. Si el **USUARIO** tiene conocimiento de que la

privacidad de su contraseña ha quedado expuesta, debe notificar inmediatamente a **GID-CONUSS** e informar acerca de cualquier fallo de seguridad.

3. Contenidos del usuario.

El **USUARIO** acepta usar los **SERVICIOS** sólo para fines legales y queda prohibida la publicación, distribución y alojamiento en los **SERVICIOS** de:

(a) cualquier material ilegal, perjudicial, amenazante, abusivo, acosador, difamatorio u obsceno de cualquier tipo, incluyendo, entre otros, cualquier material que fomente conductas que constituirían una ofensa penal, que produjera responsabilidad civil o que de otro modo infrinja cualquier ley aplicable local, estatal, nacional o internacional.

(b) del cual el **USUARIO** no tenga el derecho de poner a disposición conforme a ninguna ley o relación contractual; o

(c) infrinja cualquier derecho de patente, marca comercial, secreto de mercado, derecho de autor u otros derechos de propiedad de cualquier parte. El **USUARIO** acepta no transmitir, subir, publicar, enviar por email o poner a disposición de otro modo virus informáticos, publicidad no solicitada o no autorizada, material promocional o de venta, lo que incluye cadenas de correspondencia, correos masivos o cualquier otra forma de "spam".

El **USUARIO** acepta NO

(i) falsificar encabezados o manipular de otro modo los identificadores de archivos con el objeto de ocultar el origen de cualquier contenido del **USUARIO**;

(ii) recopilar o almacenar información de identificación personal acerca de otros usuarios sin la debida autorización y consentimiento de estos;

(iii) utilizar herramientas de interceptación de datos;

(iv) hacer uso de software malicioso o de obstaculización de redes o,

(v) realizar accesos abusivos a los equipos informáticos que permiten la prestación de los **SERVICIOS** o la prestación los **SERVICIOS** de los demás usuarios. El **USUARIO** reconoce que **GID-CONUSS** tiene el derecho (pero no la obligación), de eliminar el contenido del **USUARIO** que viole estos Términos y Condiciones, y nos reservamos el derecho de denegar el servicio y/o cancelar cuentas sin previo aviso para cualquier usuario que viole estos Términos y Condiciones o infrinja los derechos de terceros.

4. Seguridad de los servicios.

La seguridad de las instancias asignadas por parte de los **SERVICIOS** prestados recae bajo la administración del **USUARIO**, así como el manejo de las copias de seguridad de los datos, código fuente, aplicaciones y bases de datos que hagan parte de sus contenidos,

GID-CONUSS no se responsabiliza en casos de falla total o parcial de los servidores y que imposibiliten la recuperación de los contenidos por los sistemas de almacenamiento implementados.

5. Quejas por derecho de autor.

GID-CONUSS respeta la propiedad intelectual de terceros. Si usted cree que su trabajo ha sido copiado en este sitio de manera que constituya una infracción a sus derechos de autor, puede escribir a cloudeisi@gmail.com para notificarnos acerca de la situación.

6. Descargo de responsabilidad.

El **USUARIO** reconoce y acepta que utiliza este servicio sin cargo alguno y que por lo tanto este uso se ofrece “tal como está”, **GID-CONUSS** expresamente niega toda garantía de cualquier tipo, ya sea expresa o implícita. Como **USUARIO** acepta de forma expresa que todo uso de la información disponible en las los **SERVICIOS** asignados son para su uso personal o académico y que dicha información se utilizará bajo su propio riesgo y responsabilidad. **GID-CONUSS** expresamente niega toda responsabilidad por errores u omisiones en la información, los materiales y las funciones que se incluyen en nuestros servidores o en los sitios de terceros enlazados desde o hacia nuestros servicios por parte del **USUARIO**.

7. Indemnización.

El **USUARIO** acepta indemnizar y liberar de toda responsabilidad a **GID-CONUSS** sus directivos, desarrolladores, investigadores, y administradores en cuanto a cualquier reclamación o demanda, lo que incluye los honorarios razonables de representación legal, entablada por terceros debida a, o que resulte de, el uso por parte del **USUARIO** de los **SERVICIOS** prestado.

8. Potestad jurídica.

Los **SERVICIOS** son administrados por **GID-CONUSS** desde sus oficinas dentro de la República de Colombia. Al hacer uso de los **SERVICIOS**, tanto el **USUARIO** como **GID-CONUSS** aceptan que todos los asuntos relacionados con su uso de los **SERVICIOS** estarán regidos e interpretados de acuerdo con la legislación de la República de Colombia, sin considerar los principios de conflictos de leyes, como si se realizaran enteramente dentro de la República de Colombia.

9. Modificación de los términos de uso del servicio.

GID-CONUSS puede cambiar los términos y condiciones de los Términos de Uso del Servicio. **GID-CONUSS** hará las notificaciones de los cambios y publicará la versión más

reciente en la sección Términos de Uso del sitio <http://cloudeisi.uis.edu.co>, si el **USUARIO** no está de acuerdo con las modificaciones puede dejar de usar los **SERVICIOS** informando a **GID-CONUSS**.

10. Confidencialidad de la Información.

GID-CONUSS no compartirá ni revelará la información confidencial con terceros, excepto que tenga expresa autorización de parte del **USUARIO**, o cuando ha sido requerido por orden judicial o legal.

Anexo B. Manual Administrativo usuarios enjaulados

MANUAL ADMINISTRATIVO DE LA INSTANCIA DE PROGRAMACIÓN WEB

Para que los estudiantes puedan subir sus contenidos se crearon usuarios enjaulados con un único directorio con permiso de escritura, el cual posee un vínculo simbólico al directorio /var/www, también se establecieron cuotas de almacenamiento en sus respectivos directorios personales, una cuota de límite 'soft' de 185MB y un límite 'hard' de 200MB.

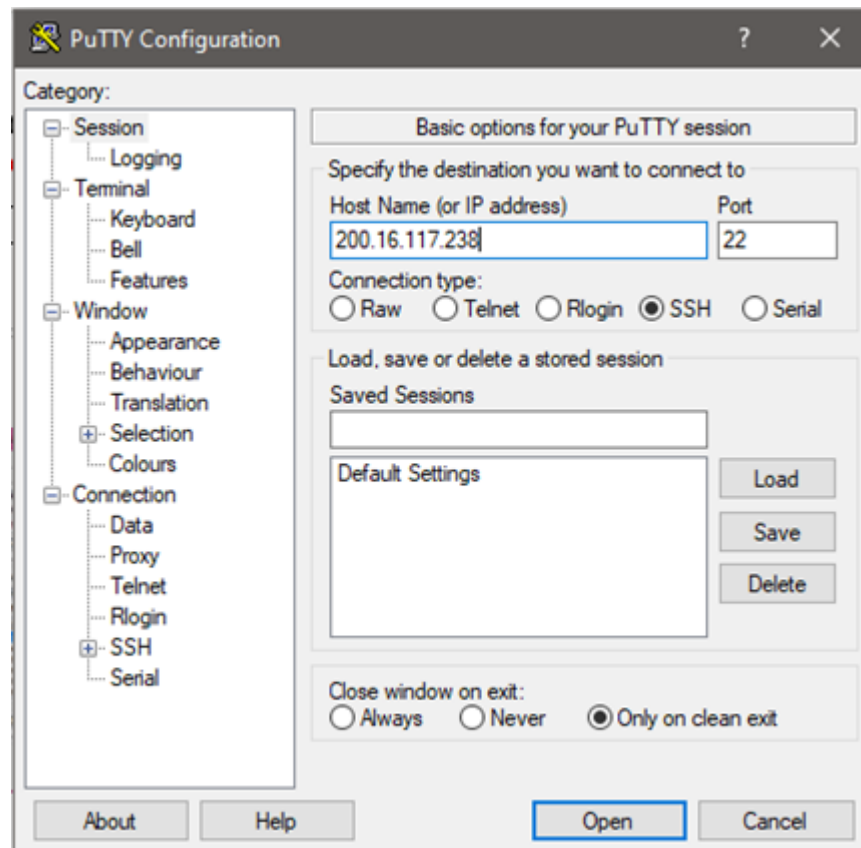
1. GENERACION DE GRUPOS DE PROGRAMACION WEB

1.1. Conexión por ssh

Para el uso de esta instancia se le hace entrega de usuario y password para el ingreso a la respectiva instancia, donde el profesor (en caso de que el profesor sea el administrador de la instancia) o el administrador (según información de usuarios o grupos reportados por el profesor) estará en las condiciones de generar los usuarios enjaulados que considere necesarios, el cual describiremos a continuación:

Debe generar una conexión SSH con la IP asignada puede usar el programa de PuTTY como se observa en la figura 1.

Figura 1. Conexión a PuTTY



Fuente: autor

El usuario administrador realiza su login y su password, no se puede hacer login directamente como root (configuración de ssh.config)

1.2. Ubicación de archivos dentro del el sistema

Posteriormente nos movemos dentro las carpetas del sistema operativo como se observa donde se encuentran los archivos necesarios para iniciar el proceso de generación de cada uno de los usuarios el sistema en la siguiente ruta

```
cd /root/setup_chroot
```

1.3 Uso del menú

El menú cuenta con las funciones completas para administrar el sistema sin necesidad de ejecutar script por script. Para ejecutar el Menú tenemos que ser root y ejecutarlo con el siguiente comando:

```
bash /root/setup_chroot menu.sh
```

En este menú se encuentran las funciones:

Figura 2. Menú administrador

```
root@actuweb:/home/administrador# bash /root/setup_chroot/menu.sh
Seleccione la acción que desea realizar:
1) Crear usuario                6) Cerrar materia              11) Backup usuario
2) Mostrar usuarios enjaulados  7) Abrir usuario               12) Backup semestre
3) Generar BD                  8) Abrir materia              13) Backup total
4) Enviar correo usuarios      9) Eliminar Usuario           14) Salir
5) Cerrar usuario              10) Eliminar Usuario y archivos
#?
```

Fuente: Autor

Seleccionamos el número de la acción que deseamos realizar y cada opción mostrará el modo de uso.

1.4 Creación de los usuarios de programación web dentro del sistema

Para generar los usuarios enjaulados debemos ejecutar el script generación de usuarios el cual se puede ejecutar así:

bash setup_chroot.sh USUARIO //importante hay que realizarlo usuario por usuario y no tiene ningún problema; genera como salida el archivo enjaulados.txt, en forma acumulativa. Después de terminado el semestre se puede limpiar el archivo enjaulados.txt si el administrador lo desea.

Al ejecutar el script se sugiere el siguiente formato con el objeto de tener un control cronológico el cual se observa en la tabla 1.

Tabla 1 Formato de generación de usuarios

USUARIO	FORMATO	20202B101
AÑO	XXXX	2020
SEMESTRES	X	2
GRUPO	XX	B1
SUBBRUPO	XX	01

Este script crea un usuario nuevo dentro del sistema con privilegios para ejecutar sftp, en el archivo enjaulados.txt que se encuentra dentro de la misma carpeta. Se almacena la dupla usuario y contraseña. Las instrucciones para el envío de este archivo al administrador y/o profesor se encuentran especificadas en el literal 1.6

Nota 1: El archivo setup_chroot.sh incluye automáticamente a los usuarios en el grupo "sconply" así quedan enjaulados para FileZilla, si los usuarios creados no se agregan a este grupo, los usuarios pueden navegar entre las diferentes carpetas, es decir no quedan en la jaula.

Para comprobar que los usuarios están en el grupo se ejecuta el siguiente comando:

less /etc/group

y se ubica el grupo sponly donde deben estar los usuarios creados, en caso de que no pertenezcan al grupo sponly, se deberá incluir manualmente con el editor nano o vim.

Nota 2: para que apache pueda correr es necesarios que la carpeta enjaulada tenga permiso 755 es decir para otros que incluye el servidor apache “r-x”

1.5 Proceso cerrar los sitios fin de semestre o después de haber creado los usuarios en /root/scripts:

- abrir_usuario.sh: funciona para un grupo de trabajo y cambia el propietario.
Ejemplo: *bash abrir_usuario.sh 20202B108*
- cerrar_usuario.sh: impide subir o actualizar el sitio; le cambia el propietario a root, root es el propietario.
Ejemplo: *bash cerrar_usuario.sh 20202B108*
- cerrar_usuario_mitadSemestre.sh: cambia a propietario a root; pero no interfiere con las BD porque todavía no existe, además coloca derechos como chmod 755, permitiendo el acceso a los sitios para leer.
Ejemplo: *bash cerrar_usuario_mitadSemestre.sh 20202B108*
- abrir_materia.sh: funciona para abrir un grupo de usuarios que tenga el mismo formato
Ejemplo: *bash abrir_materia.sh 20202B1*
- cerrar_materia.sh: funciona para cerrar un grupo de usuarios que tenga el mismo formato
Ejemplo: *bash cerrar_materia.sh 20202B1*

- eliminar_usuario.sh: funciona para eliminar un usuario, pero conservando los archivos.

Ejemplo: *bash eliminar_usuario.sh 20202B1*

- eliminar_usuario_archivos.sh: funciona para eliminar por completo a un usuario y todos sus archivos.

Ejemplo: *bash eliminar_usuario_archivos.sh 20202B1*

1.6 Creación de la BD y el usuario para los estudiantes de programación web.

El proceso por seguir es la creación de la base de datos y el usuario de la base de datos en el cual el nombre de la base y el usuario son el mismo, con el objetivo de mantener un orden cronológico dentro de la instancia. El password es el mismo del proceso anterior. Para realizar este proceso debemos estar en el directorio:

/root/setup_chroot

Ejecutamos el archivo con la siguiente línea de código:

bash gen_BD_materia.sh enjaulados.txt

El único parámetro suministrado es el archivo de enjaulados.txt del respectivo semestre, el script genera la base de datos, el usuario y la contraseñas a partir de este archivo, con la característica que la BD y el usuario tienen el mismo nombre.

Nota 3: Si se quiere ejecutar uno a uno se debe correr el siguiente script siguiendo el formato presentado en la tabla 2.

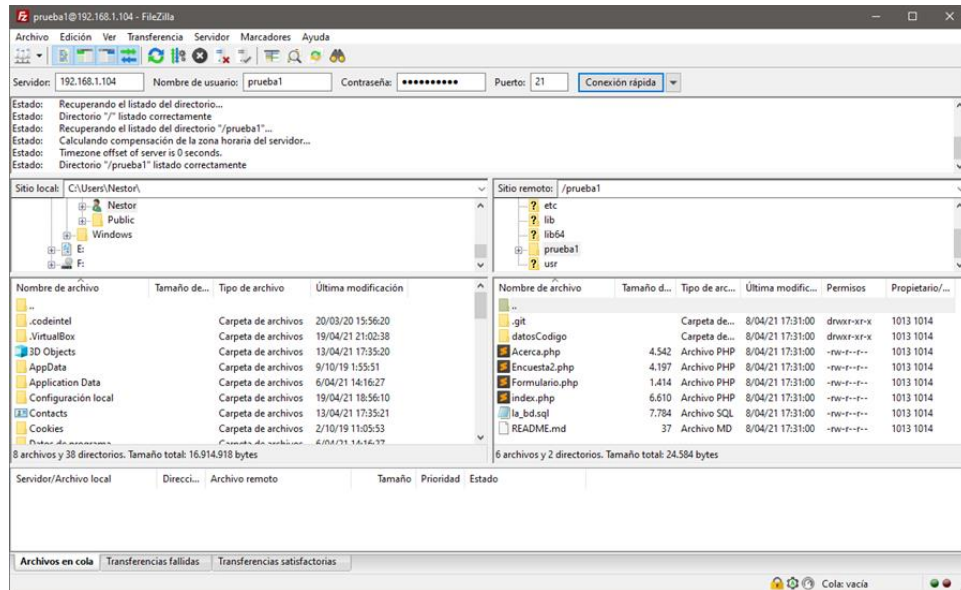
bash gen_BD_grupo.sh base_datos nombre_usuario contraseña

Tabla 2 Generación de bases de datos

SCRIPT	BASE DE DATOS	USUARIO	PASSWORD
gen_BD_grupo.sh	20171B101	20171B101	Y6PggOipwf

Con el proceso anterior damos por finalizado el procedimiento de generación de las instancias para los usuarios de programación web donde los estudiantes pueden acceder y transmitir sus archivos por ftp como se observa en la figura 3. Ofreciéndoles un entorno real de hosting en producción.

Figura 3. Entorno FileZilla



Fuente: Autor

1.7 Envío de claves al profesor de programación web

Se generaron contraseñas pseudoaleatorias por el script "setup_chroot.sh" y están organizadas en duplas 'usuario' 'contraseña'. Como ejemplo:

20171B101 Y6PjQOipqw

20171B102 wX_BkayZol

Como resultado de la ejecución se obtuvo el archivo enjaulados.txt, este archivo se envía mediante el comando:

```
bash enviar_password_correo.sh CORREO
```

dónde CORREO es el correo del administrador y/o profesor el cual hará entrega de usuarios y contraseña a los respectivos grupos de trabajo, se sugiere el uso de correo Gmail, sin embargo, funciona con otros motores de correo llegando a la bandeja de SPAM.

Nota 4: Otra forma de transferir el archivo enjaulados.txt es desde /root/setup_chroot copiar al directorio con el siguiente comando

```
cp enjaulados.txt /home/admin/admin
```

Luego por SFTP usando FileZilla se hace la conexión usando las siguientes credenciales asignadas para el administrador.

Se descarga el archivo al sistema Windows, se abre en Word y se convierte en PDF, para luego subirlo al Drive o al Meiweb en Recursos Del Profesor y que éste tenga las claves de los usuarios en el respectivo semestre.

2. ADMINISTRACIÓN DE LOS USUARIOS DE PROGRAMACIÓN WEB

Con el objetivo de mantener un control sobre los límites de transferencia de archivos, se crearon unos scripts los cuales ofrecen diferentes opciones de control al administrador de la instancia, en este caso el profesor de la materia de programación web. El cual se encuentra en la siguiente ruta.

```
/root/script
```

2.1. Abrir usuario

Al finalizar el semestre después de haberse desactivado la actualización del sitio mediante sftp, se pueden abrir nuevamente usuario específico como un caso especial durante un periodo o indefinidamente para esto tenemos el siguiente

formato donde se define en primer lugar por un lapso e interminablemente como se observa en la tabla 3, el formato es el siguiente

```
bash abrir_grupo.sh USUARIO TIEMPO
```

Tabla 3 Formato de ejecución del script para abrir usuario

SCRIPT	USUARIO	TIEMPO (min)
abrir_grupo.sh	20171B101	10
abrir_grupo.sh	20171B101	

Nota 5: El grupo representa un usuario en el sistema el cual equivale al líder de un grupo de trabajo de clase o simplemente el grupo puede representar a un usuario individual en caso de que no existan grupos de trabajo de clase.

2.2. Abrir diferentes usuarios

Con los formatos previamente definidos para la generación de los usuarios le damos la opción al administrador de activar la transferencia de datos de un grupo de usuarios que posean las mismas características a nivel de grupo como se observa en la tabla 4, con el siguiente formato:

```
bash abrir_materia.sh USUARIO TIEMPO
```

Tabla 4: Abrir grupo de programación web

SCRIPT	USUARIO	TIEMPO (min)
abrir_materia.sh	B1	10
abrir_materia.sh	B1	

Nota 6: El script anterior activaría todos los estudiantes del grupo B1 que se hayan creado según el formato definido para la creación de usuarios.

2.3. Cerrar usuarios

Con este script impedimos la transferencia de archivos SFTP; la ejecución se realiza de acuerdo con el formato que observamos en la tabla 5:

```
bash cerrar_grupo.sh USUARIO
```

Tabla 5: Formato de ejecución de script cerrar usuarios

SCRIPT	USUARIO
cerrar_grupo.sh	20171B101

2.4 Cerrar grupos

Con base en el formato definido en la generación de usuarios para evitar estar cerrando cada uno de los usuarios lo realizaremos de acuerdo con la característica de subgrupo facilitando el trabajo del administrador el cual se ejecuta como se observa en la tabla 6.

```
bash cerrar_materia.sh USUARIO
```

Tabla 6: Formato de ejecución del script de cerrar grupo de usuarios

SCRIPT	USUARIO
cerrar_materia.sh	B1

Nota 7: El script anterior desactiva todos los estudiantes del grupo B1 que se hayan creado según el formato definido para la creación de usuarios.

2.5 Eliminar usuarios

El Script cumple con la funcionalidad de eliminar un usuario, pero conservando los archivos del directorio /home y del directorio /var/www. Se ejecuta de la siguiente manera:

```
bash eliminar_usuario.sh USUARIO
```

Tabla 7. Formato de ejecución del script de eliminar usuarios

SCRIPT	USUARIO
Eliminar_usuario.sh	20171B101

2.6 Eliminar usuarios y archivos

Script que cumple con la funcionalidad de eliminar un usuario y todos sus archivos del directorio /home y del directorio /var/www. Se ejecuta de la siguiente manera:

```
bash eliminar_usuario_archivos.sh USUARIO
```

Tabla 8. Formato de ejecución del script de eliminar usuarios

SCRIPT	USUARIO
Eliminar_usuario_archivos.sh	20171B101

Nota 8: Una vez eliminado un usuario no hay manera de recuperar la información así que se recomienda usar sólo cuando esté seguro de realizar esta acción.

2.7 Backups de todos los proyectos de usuarios con la base de datos

Permite almacenar y comprimir los archivos de proyectos de los usuarios con el siguiente formato de ejecución, el cual se almacena dentro de la carpeta /root/backups, este script está programada la ejecución en un crontab.

```
bash projects_backup.sh
```

2.8 Backups del proyecto de un usuario con la base de datos:

Permite almacenar y comprimir los archivos del proyecto de un usuario con el siguiente formato de ejecución, el cual se almacena dentro de la carpeta /root/backups, este script está programada la ejecución en un crontab.

```
bash usuario_backup.sh USUARIO
```

Tabla 9. Formato de ejecución del script de eliminar usuarios

SCRIPT	USUARIO
usuario_backup.sh	20171B101

3. EJEMPLO DE USO

Para subir los contenidos deben usar un cliente ftp que les permita conexiones sftp como filezilla, WinSCP o scp el punto de entrada está establecido así 192.168.94.116

Un ejemplo de datos de conexión para un usuario sería así:

Servidor 192.168.64.116

Usuario 20171B101

Contraseña Y6PggOipwf

4. TÉRMINOS Y CONDICIONES DEL SERVICIO.

Los términos y condiciones del servicio se encuentran indexados en la ubicación /var/www del servidor y pueden ser consultados sólo visitando la dirección IP del servidor.

Anexo C. Manual administrativo usuarios enjaulados usando contenedores

MANUAL ADMINISTRATIVO DE LA INSTANCIA DE PROGRAMACIÓN WEB

Para que los estudiantes puedan subir sus contenidos se crearon usuarios enjaulados con un único directorio con permiso de escritura, el cual no podrán entrar por SSH a la MV anfitriona.

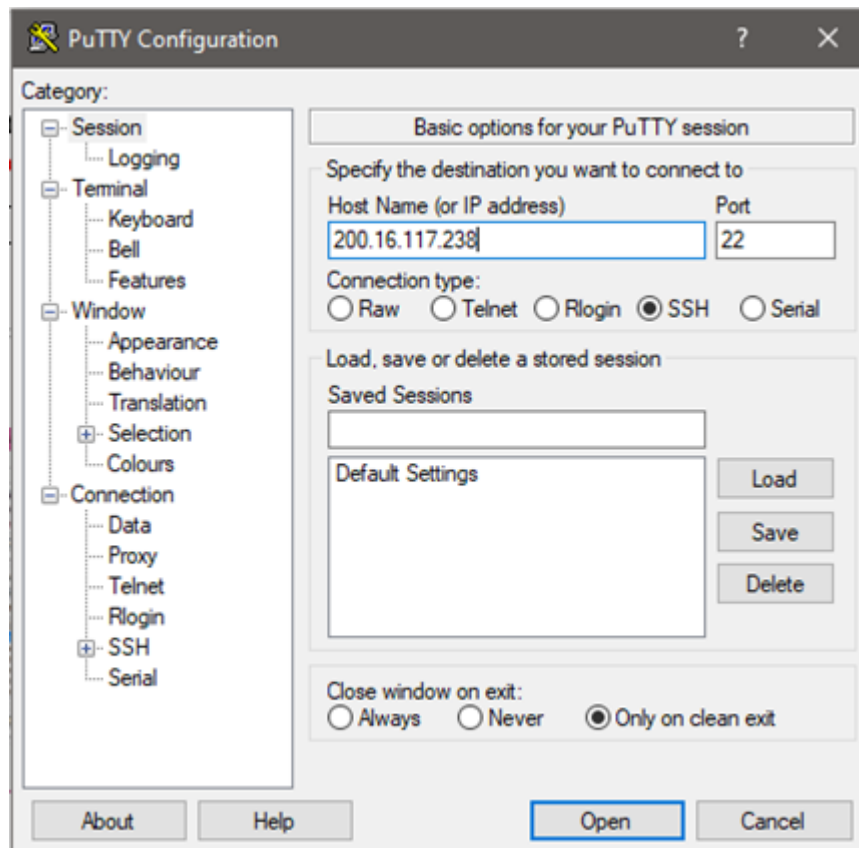
1 GENERACION DE GRUPOS DE PROGRAMACION WEB

1.1 Conexión por SSH:

Para el uso de esta instancia se le hace entrega de usuario y password para el ingreso a la respectiva instancia, donde el profesor (en caso de que el profesor sea el administrador de la instancia) o el administrador (según información de usuarios o grupos reportados por el profesor) estará en las condiciones de generar los usuarios enjaulados que considere necesarios, el cual describiremos a continuación:

Debe generar una conexión SSH con la IP asignada puede usar el programa de PuTTY como se observa en la figura 1

Figura 1. Conexión a PuTTY



Fuente: autor

El usuario administrador realiza su login y su passwd, no se puede hacer login directamente como root (configuración de ssh.config)

1.2. Ubicación de archivos dentro del el sistema

Posteriormente nos movemos dentro las carpetas del sistema operativo como se observa donde se encuentran los archivos necesarios para iniciar el proceso de generación de cada uno de los usuarios el sistema en la siguiente ruta

```
cd /root/setup_chroot
```

```
cd /root/backups
```

```
cd /root/archivos
```

```
cd /root/scripts
```

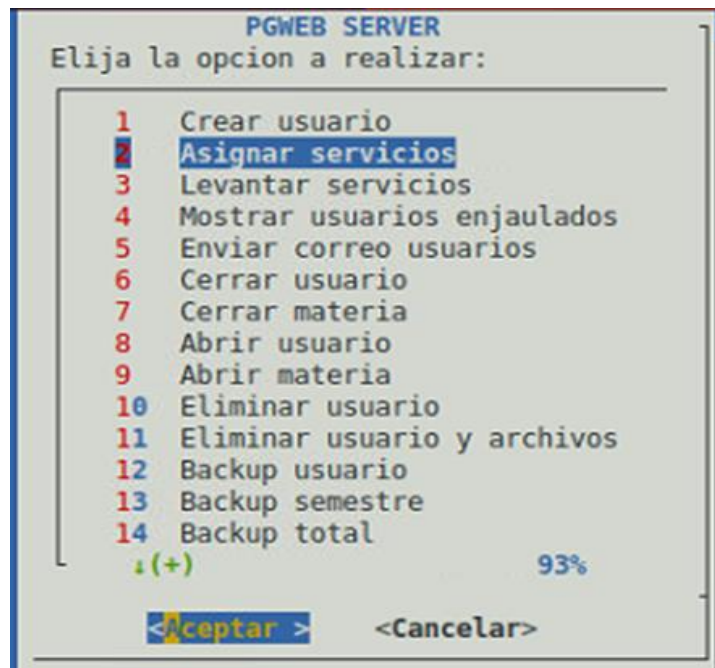
2. USO DEL MENÚ

El menú cuenta con las funciones completas para administrar el sistema sin necesidad de ejecutar script por script. Para ejecutar el Menú tenemos que ser root y ejecutarlo con el siguiente comando:

```
bash /root/setup_chroot/menu.sh
```

En este menú que se hizo con la herramienta dialog, se encuentran las funciones:

Figura 2. Menú Administrativo



Fuente: Autor

Seleccionamos el número de la acción que deseamos realizar y cada opción mostrará el modo de uso.

A continuación, se explica los scripts usados para el menú:

2.1 Creación de los usuarios de programación web dentro del sistema

Para generar los usuarios enjaulados debemos ejecutar el script generación de usuarios el cual se puede ejecutar así:

bash setup_chroot.sh USUARIO //importante hay que realizarlo usuario por usuario y no tiene ningún problema; genera como salida el archivo enjaulados.txt, en forma acumulativa. Después de terminado el semestre se puede limpiar el archivo enjaulados.txt si el administrador lo desea.

Al ejecutar el script se sugiere el siguiente formato con el objeto de tener un control cronológico el cual se observa en la tabla 1.

Tabla 1. Formato de generación de usuarios

USUARIO	FORMATO	20202B101
AÑO	XXXX	2020
SEMESTRES	X	2
GRUPO	XX	B1
SUBGRUPO	XX	01

Este script crea un usuario nuevo dentro del sistema con privilegios para ejecutar SFTP, en el archivo enjaulados.txt que se encuentra dentro de la misma carpeta. Se almacena la dupla usuario y contraseña. Las instrucciones para el envío de este archivo al administrador y/o profesor se encuentran especificadas en el literal 2.3

Nota 1: El archivo `setup_chroot.sh` incluye automáticamente a los usuarios en el grupo “sconly” así quedan enjaulados para FileZilla, si los usuarios creados no se agregan a este grupo, los usuarios pueden navegar entre las diferentes carpetas, es decir no quedan en la jaula.

Para comprobar que los usuarios están en el grupo se ejecuta el siguiente comando:

```
less /etc/group
```

y se ubica el grupo sconly donde deben estar los usuarios creados, en caso de que no pertenezcan al grupo sconly, se deberá incluir manualmente con el editor nano o vim.

2.2. proceso cerrar los sitios fin de semestre o después de haber creado los usuarios en /root/scripts

- `abrir_usuario.sh`: funciona para un grupo de trabajo y cambia el propietario
Ejemplo: `bash abrir_usuario.sh 20202B108`
- `cerrar_usuario.sh`: impide subir o actualizar el sitio; le cambia el propietario a root, root es el propietario
Ejemplo: `bash cerrar_usuario.sh 20202B108`
- `cerrar_usuario_mitadSemestre.sh`: cambia a propietario a root; pero no interfiere con las BD porque todavía no existe, además coloca derechos como `chmod 755`, permitiendo el acceso a los sitios para leer.
Ejemplo: `bash cerrar_usuario_mitadSemestre.sh 20202B108`
- `abrir_materia.sh`: funciona para abrir un grupo de usuarios que tenga el mismo formato
Ejemplo: `bash abrir_materia.sh 20202B1`

- cerrar_materia.sh: funciona para cerrar un grupo de usuarios que tenga el mismo formato
Ejemplo: *bash cerrar_materia.sh 20202B1*
- eliminar_usuario.sh: funciona para eliminar un usuario, pero conservando los archivos.
Ejemplo: *bash eliminar_usuario.sh 20202B1*
- eliminar_usuario_archivos.sh: funciona para eliminar por completo a un usuario y todos sus archivos.
Ejemplo: *bash eliminar_usuario_archivos.sh 20202B1*

2.3. Envío de claves al profesor de programación web

Se generaron contraseñas pseudoaleatorias por el script "setup_chroot.sh" y están organizadas en duplas 'usuario' 'contraseña'. Como ejemplo:

20171B101 Y6PjQOipqw
20171B102 wX_BkayZol

Como resultado de la ejecución se obtuvo el archivo enjaulados.txt, este archivo se envía mediante el comando:

bash enviar_password_correo.sh CORREO

dónde CORREO es el correo del administrador y/o profesor el cual hará entrega de usuarios y contraseña a los respectivos grupos de trabajo, se sugiere el uso de correo Gmail, sin embargo, funciona con otros motores de correo llegando a la bandeja de SPAM

2.4. Administración de los usuarios de programación web

Con el objetivo de mantener un control sobre los límites de transferencia de archivos, se crearon unos scripts los cuales ofrecen diferentes opciones de control al administrador de la instancia, en este caso el profesor de la materia de programación web. El cual se encuentra en la siguiente ruta.

/root/script

2.4.1 Abrir usuario

Al finalizar el semestre después de haberse desactivado la actualización del sitio mediante SFTP, se pueden abrir nuevamente usuario específico como un caso especial durante un periodo o indefinidamente para esto tenemos el siguiente formato donde se define en primer lugar por un lapso e interminablemente como se observa en la tabla 2, el formato es el siguiente:

```
bash abrir_grupo.sh USUARIO TIEMPO
```

Tabla 2. Formato de ejecución del script para abrir usuario

SCRIPT	USUARIO	TIEMPO (min)
abrir_grupo.sh	20171B101	10
abrir_grupo.sh	20171B101	

Nota 2: El grupo representa un usuario en el sistema el cual equivale al líder de un grupo de trabajo de clase o simplemente el grupo puede representar a un usuario individual en caso de que no existan grupos de trabajo de clase.

2.4.2. Abrir diferentes usuarios

Con los formatos previamente definidos para la generación de los usuarios le damos la opción al administrador de activar la transferencia de datos de un grupo de usuarios que posean las mismas características a nivel de grupo como se observa en la tabla 3, con el siguiente formato:

```
bash abrir_materia.sh USUARIO TIEMPO
```

Tabla 3. Abrir grupo de programación web

SCRIPT	USUARIO	TIEMPO (min)
abrir_materia.sh	B1	10
abrir_materia.sh	B1	

Nota 3: El script anterior activaría todos los estudiantes del grupo B1 que se hayan creado según el formato definido para la creación de usuarios.

2.4.3. Cerrar usuarios

Con este script impedimos la transferencia de archivos SFTP; La ejecución se realiza de acuerdo con el formato que observamos en la tabla 4:

```
bash cerrar_grupo.sh USUARIO
```

Tabla 4. Formato de ejecución de script cerrar usuarios

SCRIPT	USUARIO
cerrar_grupo.sh	20171B101

2.4.4 Cerrar grupos

Con base en el formato definido en la generación de usuarios para evitar estar cerrando cada uno de los usuarios lo realizaremos de acuerdo con la característica de subgrupo facilitando el trabajo del administrador el cual se ejecuta como se observa en la tabla 5.

```
bash cerrar_materia.sh USUARIO
```

Tabla 5. Formato de ejecución del script de cerrar grupo de usuarios

SCRIPT	USUARIO
cerrar_materia.sh	B1

Nota 4: El script anterior desactiva todos los estudiantes del grupo B1 que se hayan creado según el formato definido para la creación de usuarios.

2.4.5 Eliminar usuarios

Script que cumple con la funcionalidad de eliminar un usuario, pero conservando los archivos del directorio /home y del directorio /var/www. Se ejecuta de la siguiente manera:

```
bash eliminar_usuario.sh USUARIO
```

Tabla 5. Formato de ejecución del script de eliminar usuarios

SCRIPT	USUARIO
Eliminar_usuario.sh	20171B101

2.4.6 Eliminar usuarios y archivos

Script que cumple con la funcionalidad de eliminar un usuario y todos sus archivos del directorio /home y del directorio /var/www. Se ejecuta de la siguiente manera:

```
bash eliminar_usuario_archivos.sh USUARIO
```

Tabla 6. Formato de ejecución del script de eliminar usuarios

SCRIPT	USUARIO
Eliminar_usuario_archivos.sh	20171B101

Nota 5: Una vez eliminado un usuario no hay manera de recuperar la información así que se recomienda usar sólo cuando esté seguro de realizar esta acción.

2.4.7. Backups de todos los proyectos de usuarios:

Permite almacenar y comprimir los archivos de proyectos de los usuarios con el siguiente formato de ejecución, el cual se almacena dentro de la carpeta */root/backups*, este script está programada la ejecución en un crontab.

```
bash projects_backup.sh
```

2.4.8 Backups del proyecto de un usuario:

Permite almacenar y comprimir los archivos del proyecto de un usuario con el siguiente formato de ejecución, el cual se almacena dentro de la carpeta */root/backups*, este script está programada la ejecución en un crontab.

```
bash usuario_backup.sh USUARIO
```

Tabla 7. Formato de ejecución del script de eliminar usuarios

SCRIPT	USUARIO
usuario_backup.sh	20171B101

3. Asignar servicios:

Los contenedores usados por el momento son MEAN STACK, MERN STACK Y LEMP, sus prototipos estarán ubicados en la dirección: (Necesario iniciar como root)

/root/archivos

El administrador tendrá que mover esos prototipos dependiendo de la necesidad del estudiante a los usuarios enjaulados creados, se puede usar los comandos manualmente o usando el menú.

Ejemplo: Los cascarones que son los archivos de cada pila tecnológica o de desarrollo, estarán ubicados en */root/archivos*, para moverse manualmente se debe usar el comando *cp* y llevarlo a la dirección del usuario, incluyendo el archivo (.env) que está en modo oculto.

Si el grupo es 2022B101:

```
cp -r * /home/2022B101/2022B101
```

Luego se revisa si el .env también se copió en la carpeta 2022B101, en dado caso que no, se copia.

```
cp .env /home/2022B101/2022B101
```

3.1. Editar puertos:

Cada contexto tendrá un archivo (.env), que contiene los puertos a usar en los contenedores.

El formato por seguir se evidencia en la tabla 8., teniendo como ejemplo un contenedor con pila de desarrollo LEMP y con el primer grupo:

Tabla 8. Formato de los puertos

NOMBRE	PUERTO	GRUPO 01
HTTP	80	801
HTTPS	443	4431
PHPMYADMIN		
N	8080	8081
PHP	800	8001
MYSQL	3306	33061

La cantidad de puertos va de 0-65535 existentes, por lo que cuando se asignan los puertos debe estar en ese rango, en este caso PHPMYADMIN no se le puede colocar "80801". Se elimina el último número del puerto actual y se reemplaza por el número del grupo.

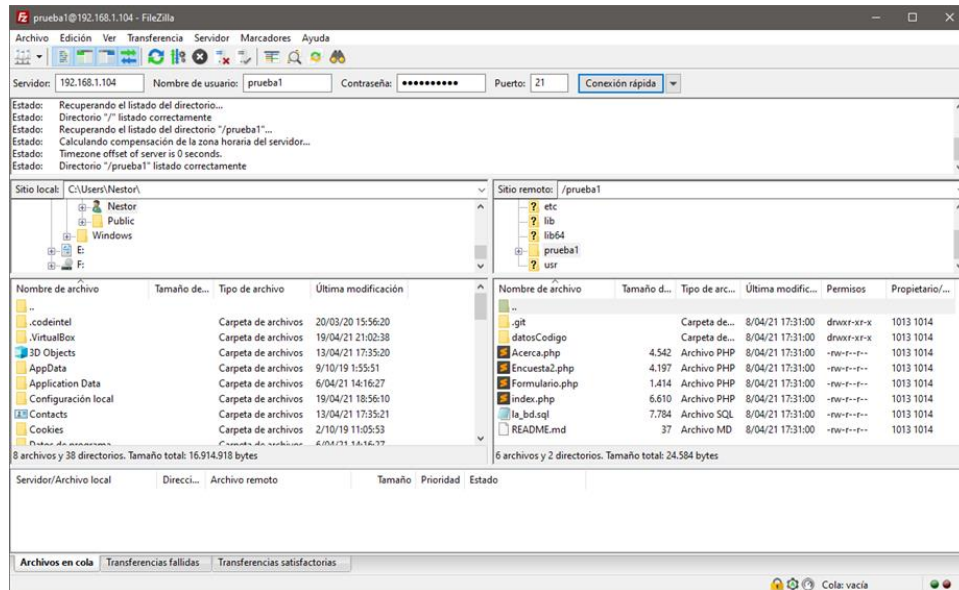
Si se cambian los puertos hay que actualizar el DockerFile

Nota 6: Recuerde que los puertos no deben ser iguales, cada grupo debe tener distinto puerto para un funcionamiento adecuado.

4. Proceso de subir archivos:

Con el proceso anterior damos por finalizado el procedimiento de generación de las instancias para los usuarios de programación web donde los estudiantes pueden acceder y transmitir sus archivos por ftp como se observa en la figura (3) ofreciéndoles un entorno real de hosting en producción.

Figura 3. Entorno FileZilla



5. Estudiantes de proyecto de grado:

A los estudiantes de proyecto de grado se les va a entregar una máquina virtual con su ip privada en Ubuntu server 20.04 LTS y el software necesario instalado, que estará ubicado en los servidores del grupo GID-CONUSS.

Ellos mismos por parte de temática, deberán investigar y conocer sobre Docker, Docker Swarm y todo lo que conlleva estas herramientas para poder usarlas en sus aplicaciones web.

Tendrán a disposición los contenedores usados y al finalizar el proyecto, deberán documentar lo que hicieron.

5.1. Mostrar aplicativo web al exterior:

La persona que desee usar una IP pública para mostrar un aplicativo web, tendrá una explicación de lo que se usa con los manuales que tiene el administrador, sin embargo, ellos mismos deberán buscar qué opción les facilita para usarlo.

6. Ejemplo de uso:

Para subir los contenidos deben usar un cliente ftp que les permita conexiones sftp como filezilla, WinSCP o scp el punto de entrada está establecido así 192.168.94.116

Un ejemplo de datos de conexión para un usuario sería así:

Servidor 192.168.64.116

Usuario 20171B101

Contraseña Y6PggOipwf

7. Términos y condiciones de uso:

Los términos y condiciones del servicio se encuentran indexados en la ubicación principal del servidor y pueden ser consultados sólo visitando la dirección IP y el puerto asignado.