

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR



FACULTAD DE INGENIERÍA

MAESTRÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

TESIS

“Implementación de un prototipo de una Plataforma como Servicio (PaaS) basada en Contenedores y Dockers sobre la nube privada virtual en Centurylink de Ecuador”.

AUTOR: Andrés Ricardo Sánchez Jacome, Ing.

DIRECTOR: Juan Francisco Chafra Altamirano, Ing., MSc.

Quito - 2022

DEDICATORIA

A mis padres Yolanda y Telmo por creer siempre en mí y brindarme una educación sustentada en ética, honestidad, lealtad y responsabilidad.

A mis hermanos, Rodrigo y Orlando por sus atenciones cuando más los necesitaba y estar siempre pendientes de mí.

A mi hermana, Gaby por sus consejos y darme aliento para continuar siempre adelante.

A mi amada esposa Karina por que vi en ti que todo es posible con disciplina, amor y perseverancia.

A mi tía, Hilda gracias por todo su apoyo que a través de la distancia siempre fue una motivación más para continuar.

Este trabajo está dedicado a ustedes con todo el amor del mundo y gratitud.

AGRADECIMIENTO

A Dios, por cobijarme con su manto divino y darme el don de la sabiduría para concluir esta especialización que amo, sin ningún percance.

A mi madre por su ternura, a mi padre y su sacrificio que hicieron que hoy me encuentre en lugar que estoy.

A Karina, por ser el combustible de mi motor y el hombro donde descansar en tiempos difíciles, gracias por el apoyo incondicional.

A la PUCE y todos mis maestros, por la paciencia y consejos que me supieron dar y me han servido para concluir con éxito la maestría.

Un agradecimiento especial al Ing. Juan Francisco Chafra Altamirano, Ing. David Zambonino por su tiempo y asesoramiento para realizar el trabajo de investigación.

ÍNDICE DE CONTENIDOS

DEDICATORIA	II
AGRADECIMIENTO	III
ÍNDICE DE CONTENIDOS	IV
ÍNDICE DE FIGURAS.....	VII
ÍNDICE DE TABLAS	X
RESUMEN.....	XI
INTRODUCCIÓN.....	XII
JUSTIFICACIÓN.....	XIV
PLANTEAMIENTO DEL PROBLEMA	XVI
OBJETIVOS	XVII
Objetivo General:	XVII
Objetivos Específicos:	XVII
CAPÍTULO I.....	18
1. FUNDAMENTACION TEORICA	18
1.1 CONCEPTOS BASICOS	18
1.1.1 Virtualización	18
1.1.2 Cloud Computing.....	22
1.1.3 Contenedores.....	25
1.2 ESTADO DEL ARTE	27
1.2.1 Antecedentes.....	27
1.2.2 Línea de Tiempo de los contenedores.....	29
1.2.3 Actualidad de los Contenedores.....	30
CAPÍTULO II.....	33
2. DISEÑO DEL PROTOTIPO	33
2.1 Introducción al diseño.....	33
2.2 Consideraciones de diseño.....	35
2.2.1 Facilidad de despliegue de aplicaciones.....	35
2.2.2 Optimización de cómputo.....	36
2.2.3 Administración y orquestación.....	36
2.2.4 Tolerancia fallos.....	36
2.2.5 Alta disponibilidad.....	36
2.3 Construcción de Diseño.....	37
2.4 Configuración de PaaS.....	41

2.4.1	Preparación de nodos o máquinas virtuales.	41
2.4.2	Instalación de Dockers.	43
2.4.3	Configuración de Docker-Swarm.	46
2.4.4	Instalación y configuración de la plataforma de gestión.	47
2.5	Diseño y configuración de la capa virtual del IaaS.....	51
2.5.1	Diseño del plano de control de virtualización (Vcenter).....	52
2.5.2	Instalación de Hipervisores en Host de cómputo.	54
2.5.3	Creación y afinamiento de Clúster de cómputo.....	57
2.5.4	Diseño y Configuración de los componentes en capa de red virtual.	61
2.5.5	Configuración de la capa de Almacenamiento.	67
2.6	Diseño y configuración de la capa física del IaaS.....	70
2.6.1	Preparación y Configuración de Nodos de cómputo.	71
2.6.2	Diseño y Configuración de la Red de Datos.....	75
2.6.3	Diseño y Configuración de la Red SAN Fibra Canal.	84
CAPÍTULO III.....		90
3.	FUNCIONAMIENTO.	90
3.1	Pruebas del funcionamiento Prototipo	90
3.2	Resultados Obtenidos.	96
CAPÍTULO IV		98
4.	Análisis Financiero, Caso de Negocio para el modelo de CaaS (Container as a Service).	98
4.1	Análisis de Mercado	98
4.2	Organización y Recursos.....	103
4.2.1	Capex.....	104
4.2.1.1	Servidores	104
4.2.1.2	Almacenamiento.....	104
4.2.1.3	Networking.	105
4.2.1.4	Seguridad de Red.....	105
4.2.1.5	Hardware para Backup.....	105
4.2.1.6	Alojamiento de Infraestructura.	105
4.2.2	Opex.....	105
4.2.2.1	Licencia de Backup.....	105
4.2.2.2	Licencia de Monitoreo.....	105
4.2.2.3	Licenciamiento de Sistemas Operativos.	105
4.2.2.4	Licenciamiento de virtualización de servidores.	106
4.2.2.5	Licenciamiento de Docker Engine.....	106

4.2.2.6	Energía.....	106
4.2.2.7	Gestor de Networking.....	106
4.2.2.8	Gestor de Virtualización.....	106
4.2.2.9	Gestor de Sistemas Operativos.....	106
4.2.2.10	Gestor de Contenedores y Devops.....	106
4.2.2.11	Gestor de Seguridad.....	107
4.2.2.12	Operación Base.....	107
4.2.2.13	Servicios Profesionales.....	107
4.2.3	Plan y Proyección Financiera.....	107
CONCLUSIONES.....		112
RECOMENDACIONES.....		114
BIBLIOGRAFÍA.....		115

ÍNDICE DE FIGURAS

Figura 1.- Arquitectura Tradicional vs Virtualizada.	19
Figura 2.- Tipos de Hipervisores.	21
Figura 3.- Modelos de Servidores.	25
Figura 4.- Diferencia entre VM y Contenedores.	26
Figura 5.- Línea del Tiempo de los Contenedores.	30
Figura 6.- Arquitectura Docker.	31
Figura 7.- Orquestación Docker-Swarm.	32
Figura 8.- Responsabilidad Compartida.	33
Figura 9.- Diseño en alto Nivel.	35
Figura 10.- Capa Motor de Contenedores.	37
Figura 11.- Nodos Manager y Nodos Workers.	38
Figura 12.- Funcionamiento del Plano de Control.	39
Figura 13.- Cargas de trabajo.	40
Figura 14.- Diseño del prototipo en bajo nivel.	41
Figura 15.- Plataformas compatibles con Docker.	42
Figura 16.- Configuración de Permisos.	44
Figura 17.- Instalación de Docker.	45
Figura 18.- Instalación de Docker-Compose.	45
Figura 19.- Docker activo y corriendo.	45
Figura 20.-Version de Docker-Compose.	46
Figura 21.- Numero de nodos administrados en el Clúster.	47
Figura 22.- Direccionamiento Nodos Máster.	47
Figura 23.-Estado de salud de Clúster.	47
Figura 24.- Funcionalidades Portainer.	48
Figura 25.- Arquitectura Portainer.	49
Figura 26.- Instalación de Portainer.	49
Figura 27.- Visualización del Clúster Swarm en Portainer.	50
Figura 28.- Vista de clúster Portainer.	51
Figura 29.- Requerimientos de Hardware para Vcenter.	53
Figura 30.- Requerimientos de Almacenamiento para Vcenter.	53
Figura 31.- Instalación de Vcenter.	54
Figura 32.- Activación de Hiperproceso.	55
Figura 33.- Configuración Esxi.	56

Figura 34.- Red de vMotion.....	60
Figura 35.- Tarea de VMotion.	60
Figura 36.- Arquitectura vDS.....	62
Figura 37.- Arquitectura vDS con LAG.....	63
Figura 38.- Diagrama de Red de la solución.	64
Figura 39.- Diagrama Virtual.	65
Figura 40.- Data Storage Esxi.....	67
Figura 41.- Storage Vmotion.	69
Figura 42.- Clúster de Storage.....	70
Figura 43.- Análisis de Compatibilidad VMware.....	73
Figura 44.-Configuración UEFI Lenovo.....	74
Figura 45.- Diagrama de Conexión de Host.	75
Figura 46.- Private Vlans	76
Figura 47.- Trafico de Vlans Privadas	77
Figura 48.- Arquitectura de Red.....	79
Figura 49.- Topología de Red Recomendada por Cisco.	81
Figura 50.-Single-Homed FEX Topology.....	82
Figura 51.- Red De Administración	83
Figura 52.- Red de Producción.	84
Figura 53.- Componentes SAN	86
Figura 54.- Topología de la RED SAN.	88
Figura 55.- Asignación de Zonas y Alias.....	89
Figura 56.- Inicio de Plataforma Portainer.....	91
Figura 57.- Registro de sitio Docker-Hub en Portainer.....	92
Figura 58.- Despliegue de contenedores como un servicio.....	93
Figura 59.- Detalles del despliegue de servicios.	94
Figura 60.- Descarga de la imagen del contenedor.....	94
Figura 61.- configuración del Puerto del servicio de Prueba.	94
Figura 62.- Calibración de límites de consumo de CPU y RAM de la aplicación de prueba.	95
Figura 63.- Ejecución de contenedor como Servicio.	95
Figura 64.- Vista Grafica de la ejecución del contenedor de Prueba.	96
Figura 65.- Interfaz Web de la aplicación de Prueba.....	96
Figura 66.- Uso de la aplicación de Prueba.	97
Figura 67.- Local CSP-Tier1	99
Figura 68.- Public Cloud CSP	99

Figura 69.- Panel de configuración AWS Elastic BeanStalk.....	100
Figura 70.- Docker ejecutándose con Elastic BeanStalk.....	101
Figura 71.- Panel de Configuración de Azure App Service.....	102
Figura 72.- Cuadrante de clasificación de servicios CaaS.....	103
Figura 73.- POD para el Servicio CaaS.....	104

ÍNDICE DE TABLAS

Tabla 1.- Tipos de Virtualización.....	20
Tabla 2.- Software de virtualización.....	21
Tabla 3.- Orquestadores de Dockers.....	27
Tabla 4.- Capas del modelo propuesto.....	34
Tabla 5.- Nodos Manager Swarm.....	38
Tabla 6.- Distribución de Nodos máster y Nodos Worker en cada host físico.....	40
Tabla 7.- Host asignados al diseño.....	40
Tabla 8.- Recursos de Máquinas Virtuales.....	42
Tabla 9.- Recomendación de instalación Docker.....	43
Tabla 10.- Instalación de Docker-Compose.....	44
Tabla 11.- Resumen de configuración Docker-Swarm.....	46
Tabla 12.- Recursos Instalación de Vcenter.....	54
Tabla 13.- Recursos de Red Esxi.....	57
Tabla 14.- Configuración de DRS Clúster.....	58
Tabla 15.- Configuración HA clúster.....	59
Tabla 16.- Parámetros vDS Administración.....	66
Tabla 17.- Parámetros vDS Producción.....	66
Tabla 18.- Configuración Storage.....	68
Tabla 19.- Parámetros Clúster de Storage.....	69
Tabla 20.- Especificaciones ThinkSystem SR630.....	71
Tabla 21.- Características Nodo de Computo.....	72
Tabla 22.-Requisitos Mínimos.....	73
Tabla 23.- Asignación de Vlans.....	78
Tabla 24.- Redes Cisco basado en capas.....	80
Tabla 25.- Equipamiento Adquirido.....	83
Tabla 26.- Recursos IP.....	84
Tabla 27.- Equipos Red SAN.....	87
Tabla 28.- Lista de Materiales 1 CAPEX.....	107
Tabla 29.- Lista de Materiales 2 OPEX.....	108
Tabla 30.- Lista de Materiales 3 OPEX.....	108
Tabla 31.- TCO del Proyecto.....	109
Tabla 32.- Caso de Negocio.....	110
Tabla 33.- Comercialización del Servicio.....	111

RESUMEN

En presente trabajo se plantea la elaboración de un prototipo de una plataforma como un servicio PaaS orientada a la ejecución de contenedores implantada sobre la nube privada de la empresa Centurylink que actualmente se llama Lumen una empresa dedicada a brindar servicios de Tecnologías de la Información, en este caso Lumen sería el Proveedor de Servicio, uno de los objetivos principales es incluir esta solución al portafolio de servicios que brinda la empresa, el prototipo será implementado sobre la Infraestructura de Nube de Lumen, el desarrollo comprende de la implementación de un prototipo desde cero, es decir; se realizará la configuración tanto de la Infraestructura como de la plataforma, el prototipo responde a una solución en capas, la capa más baja es llamada como Infraestructura como Servicio, en la parte media se tiene la capa de Plataforma como Servicio y en la parte superior al ser un servicio orientado a la ejecución de contenedores se encuentran los Contenedores como un Servicio.

La solución se compone de una capa física y virtual, la capa virtual será desarrollada mediante la herramienta de virtualización VMware, en el caso de la capa física las tecnologías usadas serían; para la red LAN se usarán equipos Cisco Nexus, para la red SAN se tiene equipamiento Brocade, en equipos de almacenamiento se disponen de Hitachi Ventara y los nodos de cómputo serían Servidores Lenovo ThinkSystem, en la capa de plataforma se tiene seis máquinas virtuales con sistema operativo CentOS, en la capa de contenedores se realizará el uso del motor Docker-Container bajo de la herramienta de orquestación Docker-Swarm la misma que nos ayudará a levantar el clúster de contenedores, la solución no estaría completa sin una interfaz amigable para el usuario, la misma que sería implementada con la ayuda de la plataforma Portainer, Portainer ofrece una interfaz web para la administración, despliegue y ejecución de contenedores.

De esta manera la plataforma estaría bajo un esquema de responsabilidad compartida, Lumen es responsable de la estabilidad de la plataforma y los recursos, mientras que el cliente tiene la responsabilidad del desarrollo y ejecución de los contenedores y sus aplicaciones. Por último, se prueba el funcionamiento de la plataforma con una aplicación básica y se documentará los resultados. Adicional se hará un breve análisis financiero de los costos y beneficios económicos de la implementación del prototipo.

INTRODUCCIÓN

Se efectuará un estudio exploratorio del estado actual de la tecnología de contenedores en conjunto con un ambiente de virtualización, se recreará el ambiente de una Plataforma como Servicio basándonos en las herramientas nativas que nos ofrece Docker como Swarm para la creación, administración y orquestación de contenedores que se implementará sobre la nube privada de la empresa Centurylink en la actualidad conocida como Lumen.

Para realizar el diseño óptimo de la infraestructura donde ejecute los contenedores, se va a realizar un estudio de las mejores prácticas propuestas por cada fabricante en su entorno de trabajo, es decir; para el diseño de la infraestructura física se va a considerar cuatro elementos importantes, como los nodos de cómputo (memoria, CPU), Almacenamiento, y red (SAN, LAN).

Cada fabricante en su campo propone una configuración recomendada de acuerdo a su experiencia tecnológica, en este caso se va a adoptar el diseño recomendado de la marca en sus guías de implementación para diseñar un ambiente IaaS (Infraestructura como Servicio), ya que sobre este ambiente se configurará el ambiente PaaS (Plataforma como Servicio) orientado a un servicio de Contenedores.

La tecnología que se usará en el montaje de la infraestructura en cuanto a nodos de cómputo (CPU y RAM) sería Lenovo ThinkSystem de alta gama para el Centro de Datos, en el campo de almacenamiento tendremos tecnología Hitachi sobre una red SAN por FC (Fibra Canal) con equipos Brocade 6510 en Alta disponibilidad, para la capa de red se acogerá la tecnología Cisco Nexus 9000, 5000 y 2000 diseñada para DC en alta disponibilidad con una configuración Single Home, VMware será el mejor aliado en la ejecución de la capa virtual y despliegue de contenedores con su solución nativa de Docker llamada Swarm el encargado de Ejecutar, administrar y Orquestar nuestro ambiente de contenedores, tanto la capa física como la capa virtual se apegará a las mejores prácticas de cada fabricante.

En el capítulo I, se revisará los conceptos básicos de todos los componentes que se involucran tanto en la capa física como la capa virtual para ejecución de los contenedores, así como un estudio de investigaciones previas y los resultados obtenidos, el estado del arte y la actualidad de los contenedores.

En el capítulo II, se realizará el diseño e implementación del prototipo planteado, el mismo que se comprende del diseño de la capa física, el diseño de la capa virtual y la implementación del ambiente de contenedores basados en un motor Docker.

En el capítulo III, se realizará las pruebas de funcionamiento del prototipo, y se documentará los resultados obtenidos, funcionamiento de la plataforma, y la ejecución de ejemplos simples que nos permita evaluar el funcionamiento.

En el capítulo IV, se efectuará un análisis financiero de la implementación de la solución, la situación actual del mercado y de la competencia, la inversión y ganancias que pueden dejar el proyecto.

Finalmente se escribirán las conclusiones y recomendaciones que nos dejó como experiencia la elaboración del prototipo.

JUSTIFICACIÓN

Al proveer de una Plataforma como Servicio sobre la nube privada basada en contenedores se le proporciona al cliente otra opción más al momento de solicitar o dar de alta nuevos servicios en la nube, al contar con esta solución los beneficiados resultan ser las dos partes, la empresa genera una fidelidad del cliente, ya que tiene un amplio portafolio de servicios en la nube, el cliente se sentirá respaldado y acompañado por la empresa al encontrar la solución en la propia nube y no subcontratar a proveedores de terceros o migrar a otra nube para atender su requerimiento.

Para la ejecución de los contenedores en la plataforma es importante tener los recursos necesarios para ejecutar el ambiente de contenedores, se debe analizar la clase de nodos de cómputo a utilizar como la RAM (Random Access Memory) y la CPU (Central Processing Unit), el tipo de disco así como el acceso a ellos, el Ancho de banda requerido, también se debe cumplir con esquema de alta disponibilidad en todos los componentes que garanticen la operabilidad de los servicios, el tráfico de la red debe ser seguro, el diseño de la infraestructura donde se ejecutara el ambiente de contenedores debe obedecer a las mejores prácticas recomendadas por las marcas involucradas en la virtualización.

En estudios se ha demostrado que el uso de contenedores se distribuye mejor el uso de los recursos de cómputo, se sabe que una máquina virtual debe ejecutar todo una serie de procesos que son vitales para el funcionamiento del S.O (Sistema Operativo) provocando el consumo elevado de disco, CPU y RAM, por otro lado, los contenedores solo ejecutan la instancia que en ese momento es solicitada, consumiendo solo los recursos precisos para su ejecución, de ahí la importancia del uso de contenedores para la optimización de recursos.

El cliente al tener una Infraestructura como Servicio (IaaS) el Proveedor de Servicio entrega los recursos necesarios para el despliegue de las máquinas virtuales, el cliente depende de su personal de tecnología que la administre, debiéndose preocupar de que las máquinas virtuales tengan todos los recursos indispensables para ejecutar las aplicaciones que forman parte de su organización que son CPU, RAM, disco y recursos de red.

Con la implementación de contenedores el cliente tiene la opción de acceder a este servicio donde su única preocupación sería del desarrollo de las aplicaciones, es decir; cuando el cliente tenga la necesidad de levantar un nuevo servicio en contenedores solo debe solicitarla en la plataforma y se despliega el servicio para su afinamiento siendo un servicio

más flexible y rápido, internamente en las capas inferiores el proveedor de servicio Lumen es el responsable y debe operar, velar y garantizar los recursos óptimos para que funcione la plataforma, de esta manera el cliente se desliga de la administración y operación de las máquinas virtuales y se enfoca solo en la administración y funcionamiento de las aplicaciones. De esta manera se entregaría una plataforma como servicio (PaaS) orientada a la ejecución de contenedores como servicio (CaaS).

PLANTEAMIENTO DEL PROBLEMA

Ante la ausencia de una Plataforma como Servicio orientado a la ejecución de contenedores en la empresa Centurylink (Lumen) los clientes buscan esta solución en otros proveedores, o en otros casos deben concurrir a desarrolladores de tecnología externos para que implementen todo un ambiente de contenedores en su ambiente de Infraestructura como Servicio, Centurylink (Lumen) al no tener dicho el servicio en su portafolio corre el riesgo de perder clientes, ganancias y prestigio como empresa de tecnología en la nube.

Al proveer de una infraestructura como servicio a los clientes, no se optimizan de una mejor manera todos los recursos que pueden ser entregados por los nodos de cómputo como CPU, RAM, DISCO, recursos de RED y se ha evidenciado que el Sistema Operativo de las máquinas virtuales demandan dichos recursos más que las propias aplicaciones del cliente. Cuando se debe dar de alta un nuevo nodo de cómputo este conlleva a la liberación de recursos de red, storage, energía, etc., por el cual es inevitable la adquisición de equipamiento nuevo y el mismo ocupando un espacio en el rack del Centro de Datos, técnicamente esto también conlleva que al tener un mayor número de equipos dificultando su operación y administración.

Cuando el cliente planifica levantar un nuevo servicio en su infraestructura virtual en este caso un ambiente de contenedores basándose en cualquier tecnología disponible esto implica la creación de una nueva máquina virtual en su ambiente. Los tiempos de entrega de una nueva máquina virtual dependen de la cantidad de recursos libres de parte del clúster, a la falta de recursos el cliente presenta malestar por los tiempos elevados que se le da a su requerimiento, para proveer de recursos a la infraestructura virtual se debe adquirir hardware y software que incurren en gastos a la empresa y a su vez el llenado de infraestructura física en los racks del Centro de Datos.

OBJETIVOS

Objetivo General:

Implementar un prototipo de una Plataforma como Servicio (PaaS), orientada a la ejecución de Contenedores como servicio (CaaS) sobre una Infraestructura como servicio (IaaS) virtualizada, mediante la orquestación de contenedores Docker-Swarm administrada mediante la plataforma Portainer.

Objetivos Específicos:

- Estudiar los conceptos básicos y documentar el estado del arte de las soluciones implementadas sobre contenedores corriendo sobre ambientes virtualizados.
- Diseñar e implementar un prototipo en la infraestructura de virtualización que nos permita ejecutar un ambiente de contenedores en clúster con Docker-Swarm, y que permita aprovechar de una mejor manera el uso de los recursos de cómputo y todos los componentes involucrados en la capa física como en la capa virtual.
- Probar el funcionamiento del prototipo implementado con la ejecución de ejemplos simples, analizar los costos de la implementación, así como el caso del negocio.

CAPÍTULO I

1. FUNDAMENTACION TEORICA

En el presente capítulo se estudiará los conceptos elementales que se involucran en la construcción de prototipo, se analizará los conceptos básicos y teóricos de la capa física y virtual, así como una investigación del estado del arte que conlleva al uso de contenedores con Docker sobre ambientes virtuales.

1.1 CONCEPTOS BASICOS

1.1.1 Virtualización

Antes de involucrar y estudiar aspectos técnicos es necesario conocer un poco de la historia detrás de esta tecnología, en realidad existe mucha documentación que citan muchas empresas que empezaron a desarrollar y trabajar esta técnica, pero aún no está claro quien fue el pionero, pero la verdad es que esta tecnología nació para quedarse, la virtualización es la tecnología más usada por todas sus prestaciones que ya los estudiaremos más adelante.

Los primeros estudios registrados se remontan a los años 60`s por IBM, pero por recursos de cómputo en los años 2000 es donde empieza a tomar fuerza la virtualización, primero IBM lo llamo “supervisor” al sistema que hacía posible tener varios sistemas operativos alojados en un solo hardware, con el pasar del tiempo el nombre cambio a hipervisor. Un hipervisor es algo parecido un emulador de software, firmware, hardware que ejecuta máquinas virtuales, si un servidor ejecuta un hipervisor se lo dominara Host y las máquinas virtuales se la llama máquina invitada. El hipervisor presenta recursos de hardware a la máquina virtual invitada y gestiona la ejecución del sistema operativo, pueden existir múltiples sistemas operativos Linux, Windows, MacOS, en la misma instancia de hardware virtualizado.

En esta época las empresas se debían a servidores físicos y entornos de TI de un solo distribuidor lo que hacía que dependan de un solo hardware, ya que sus aplicaciones no podrían ejecutaren en otro proveedor de hardware haciendo que se desperdicie recursos, esto dio a que las empresas de tecnología repotencien la técnica de virtualización que el parecer esa era la solución a la problemática. La idea era de dividir los recursos de los

servidores y ejecutar las aplicaciones en sistemas operativos de cada proveedor de TI, Lo que hizo que las empresas empezaran a adquirir sistemas de virtualización haciendo que se reduzcan los costos de compras, instalación, operación, mantenimiento, etc. Como se muestra en la Figura 1 de aquí también se dio el nacimiento al cloud Computing.

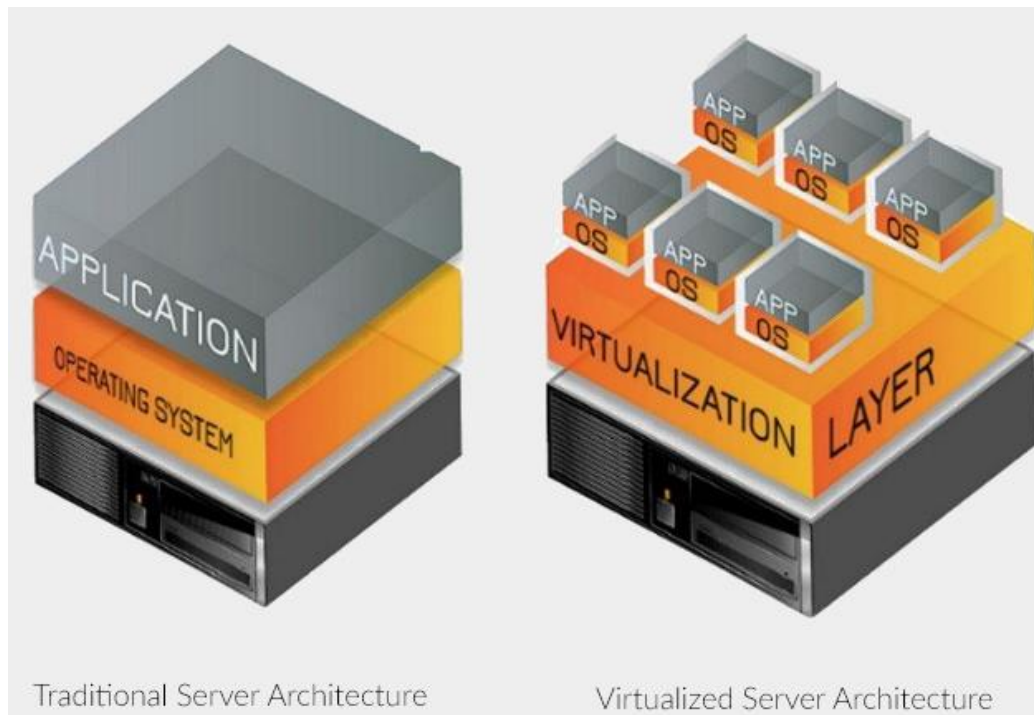


Figura 1.- Arquitectura Tradicional vs Virtualizada.
Fuente: (IDCMINI, 2022)

Como ya se lo vio la virtualización se hace posible gracias a los hipervisores que dividen los recursos de hardware y los ambientes virtuales que van requiriendo estos recursos. Los hipervisores se instalan directamente en el hardware que es la técnica más común, esto no quiere decir que podemos instalar el hipervisor en un Sistema operativo. Las máquinas virtuales se las trata como un archivo único, haciendo que este se traslade de un host a otros. Cuando un recurso virtual está en ejecución el usuario o programa solicita recursos de cómputo al host físico, el hipervisor toma la petición y provee de recursos a la máquina virtual. Existen tipos de virtualización entre los más usados son la virtualización de escritorios y de sistemas operativos, en la Tabla 1 podemos ver una descripción de estos. (RedHat, 2018)

Ítems	Descripción.
Virtualización de Datos	Técnica que consiste en distribuir los datos en diferentes ubicaciones y acceder a ellos mediante una sola fuente.
Virtualización de Escritorios.	Permite implementar muchos escritorios en una sola máquina.
Virtualización de los servidores	Consiste en dividir sus elementos y puedan usarse para realizar múltiples instrucciones, se puede ejecutar funciones específicas.
Virtualización de los sistemas operativos.	La idea consiste en ejecutar varios Sistemas Operativos en un mismo hardware, entre los más comunes ejecutar entornos Windows y Linux en forma paralela.
Virtualización de red	Consiste en definir los recursos de red y dividir el hardware de los equipos de red en varias redes definidas por software totalmente independientes.

Tabla 1.- Tipos de Virtualización.

Fuente: (RedHat, 2018)

Las ventajas de la virtualización son muchas entre ellas está la agilidad y flexibilidad desplegar una nueva vm poder tomar horas o aumentar recursos puede tomar minutos lo que anteriormente podrían tardar días. La reducción de costos es lo más importante para las empresas en dos servidores robustos se podría ejecutar varias máquinas virtuales que antes se lo hacía en varios servidores. Se puede tener una administración centralizada reduciendo los costos de operación. Dan una continuidad del negocio gracias a la alta disponibilidad se reduce la indisponibilidad de estos. Y tenemos un uso eficiente de los recursos la virtualización permite una mejor distribución de recursos y uso de estos.

Al momento de la virtualización existen varias alternativas de software para realizar la virtualización que van desde el uso libre hasta con licenciamiento, en la Figura 2 se puede visualizar la arquitectura de un esquema de virtualización, en el host se instala el hipervisor de cualquier proveedor y en la capa superior se ejecutan las máquinas virtuales.

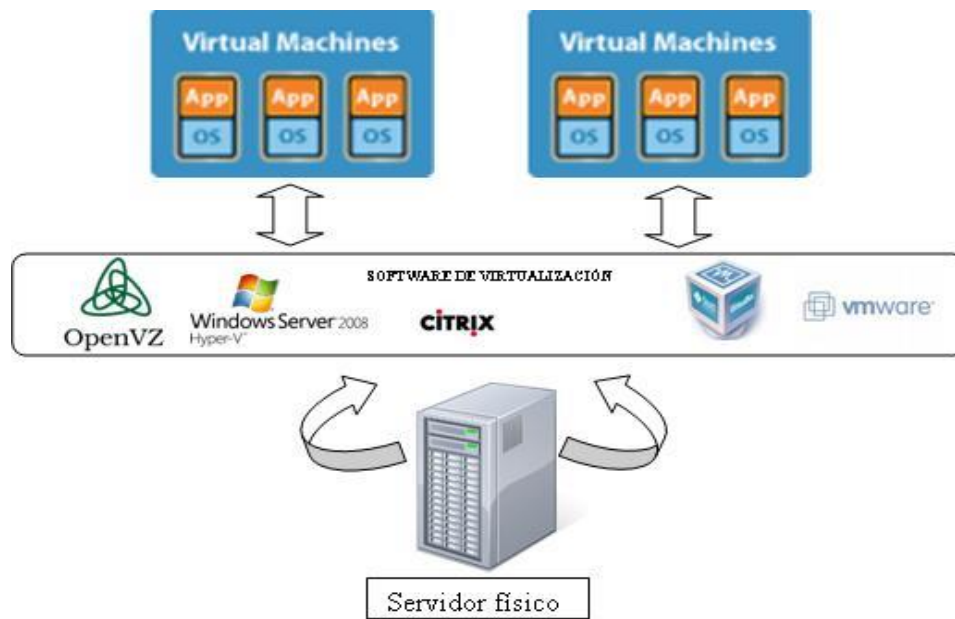


Figura 2.- Tipos de Hipervisores.
 Fuente: (Meysa I.T. Solutions, 2019)

A continuación, en la Tabla 2 se describe los distintos proveedores de software de virtualización todos tienen sus fortalezas y debilidades, la elección depende del cliente final y sus necesidades, técnicamente finalmente todos cumplen con la su función de virtualizar un ambiente.

Ítem	Descripción.
Oracle VirtualBox	Virtual Box se puede descargar de manera gratuita y tiene un tipo de licencia GNU de acceso libre.
Citrix XenServer	Es una plataforma de código abierto usada para la virtualización de infraestructuras. Cuenta con su propio hipervisor.
VMware	VMware proporciona una serie de herramientas para la virtualización de infraestructuras más populares en el mundo.
Microsoft Hyper-V	Desarrollado por Microsoft es una alternativa para la virtualización de ambientes.

Tabla 2.- Software de virtualización.
 Fuente: (ayudaley, 2020)

Son algunos de los más conocidos, en el caso de la implementación del prototipo, por la experiencia obtenida y el conocimiento técnico que se maneja VMware como la mejor opción para el desarrollo final.

1.1.2 Cloud Computing

De la mano de la virtualización, toda la industria de tecnología empezó a mirar de reojo la idea de desligarse del mantenimiento y la operación de los equipos que se adquieren para su producción de esa manera la idea de contratar un servicio de computación en la nube no era tan descabellada, con esa necesidad los proveedores de tecnología empezaron a trabajar en la computación en la nube, la principal función de la cloud Computing es ejecución de cargas de trabajo directamente en la nube, los entornos de cloud Computing son en donde se ejecutan las diferentes aplicaciones, detrás de la nube existen tecnologías agrupadas de software, hardware que proveen del cómputo. Cuando hablamos de Cloud Computing existen tipos de cómputo en la nube que al momento de elegir todo depende del tipo de negocio y recursos del cliente. En los grupos de nube tenemos la nube pública, la nube privada, y la nube Híbrida.(Red Hat, 2018)

La nube pública es el cloud Computing más común, todos los recursos como servidores, equipos de red, almacenamiento son responsabilidad y propiedad el proveedor y los servicios son consumidos mediante el internet, una de las desventajas se puede decir de la nube pública es que se comparte los recursos de hardware, almacenamiento y dispositivos con otras organizaciones que están en la misma nube y las ventajas serían costos inferiores, confiabilidad y escalabilidad. Estos entornos son usados en ambientes no productivos por la privacidad de datos, entre las nubes más comunes tenemos Google Cloud, AWS, Azure entre otros.(Azure, 2022)

La nube privada al igual que la nube pública se compone de recursos informáticos destinados a ejecutar cargas de trabajo, aplicaciones, etc., pero en este caso son exclusivamente usados por una empresa u organización, toda la infraestructura es destinada para el cliente, así como la red privada. Es así como con el surgimiento de la nube pública no puede desaparecer la nube privada como vemos sus aplicaciones son diferentes, en la nube privada las organizaciones pueden personalizar sus ambientes, las nubes privadas en su mayoría son utilizadas por agencias de gobierno, bancos, cooperativas financieras, y empresas de tamaño mediano a grande que sus cargas de trabajo son muy críticas para su negocio. Una de las desventajas de la nube privada sería el costo, en cambio, en las ventajas podemos tener la flexibilidad de personalizar el entorno de la nube a los intereses del cliente, control de sus recursos, ya que son exclusivos del cliente y no comparte con nadie, y la escalabilidad de crecer en recursos en minutos.(Azure, 2022)

Ya sea la nube pública o privada que sea elegida como proveedor de servicio, ambas manejan los mismos tipos de servicio que se brinda al cliente final, entre ellas podemos hablar de las más comunes como Infraestructura de Servicio, Plataforma como Servicio, Software como Servicio cada una con sus características y funciones.

En un servicio IaaS por sus siglas es conocida como Infraestructura como un servicio en la nube, es un ambiente de Cloud Computing que presta los servicios de infraestructura a través de la red corporativa o internet. El usuario final es el administrador de las aplicaciones, datos, Sistema Operativos que se ejecutan en ella, el proveedor de IaaS en cambio es el administrador de los elementos de virtualización, almacenamiento, red y servidores. (Red Hat, 2020).

El proveedor de servicio al prestar el servicio de IaaS es el encargado de la administración de los cuatro componentes fundamentales que se involucran en el servicio que las iremos desarrollando en esta sección, las máquinas virtuales, los sistemas de almacenamiento, la red LAN, SAN y la granja de servidores.

Las máquinas virtuales es un entorno virtual opera prácticamente igual que una máquina física es decir tiene sus propios recursos de memoria, CPU, almacenamiento e interfaces de red, como lo vimos en la sección de virtualización el hipervisor es el encargado de separar los recursos de la máquina virtual con el hardware del host físico para irlos distribuyendo de una forma controlada a cada máquina virtual. Una máquina virtual puede ejecutar cualquier sistema operativo como Windows, Linux, o macOS, la experiencia de usuario es la misma como si estuviera frente a una máquina física.

Una LAN (red de área local) es una red pequeña que se la implementa generalmente en un edificio, empresa o Centro de Datos, es la encargada de la interconectividad de los diferentes componentes en el Centro de Datos como conmutadores y servidores, entre las marcas más populares se puede encontrar HP, Cisco, Dell. Basándose en la norma ANSI/TIA-942 que determina la topología de conexión del Centro de Datos se presenta 4 tipos (Furukawa, 2021):

- Centralizada Cross-connect: se compone de un switch central ubicado en un área conocida como MDA, desde ese punto central se realiza las cruzadas de cables hacia los servidores llamado como EDA.

- T.o.R (Top-of-Rack): En el área de equipo llamada EDA se instala un equipo de red en la parte superior del rack, cada servidor o equipo se conecta directamente a este equipo.
- E.o.R. (End of Row): El rack ubicado en el área de interconexión de equipos se los llama HDA el mismo ubicado al final de la fila y mediante un cableado horizontal atiende a los racks al largo del área de racks EDA's.
- M.o.R. (Middle-of-Row): El rack HDA se lo ubica en la parte central de la fila de todos los racks destinados a los servidores y mediante el cable horizontal se atiende de forma semejantes.

Una red SAN, conocida como red de área de almacenamiento, la misma que es una red que se caracteriza por su alta velocidad generalmente de 16-32 Gbps y que proporciona el acceso a la red a nivel de bloques. Existen elementos que conforma la SAN que se los puede describir a continuación como los switches de Fibra canal, host de cómputo, y las cajas de discos, todo están comunicados mediante una variedad de tecnologías, topologías y protocolos. Las SAN también pueden abarcar múltiples sitios. La SAN presenta dispositivos de almacenamiento a un host de modo que el almacenamiento parece estar conectado localmente. Esta presentación simplificada de almacenamiento a un host se logra mediante el uso de diferentes tipos de virtualización. (SNIA, 2020).

Los sistemas de almacenamiento son equipos con grandes cantidades de espacio en disco, entre ellos pueden ser discos mecánicos SAS 7.2Krpm (revoluciones por minuto), 10Krpm, 15Krpm, y discos Flash de alto rendimiento SSD, el acceso a estos espacios de disco suelen ser mediante FC (Fibra Canal) y iSCSI(Internet SCSI) a través de una topología SAN (Storage Area Network) o NAS (Network Attached Storage), SAN se basa en un almacenamiento en bloques y se presenta a los servidores como una unidad de disco lógica llamada LUN (logical unit number), la LUN es un pedazo de un bloque grande de almacenamiento compartido y se lo presenta al host para que lo consuma como un disco local entre los principales fabricantes tenemos a Hitachi Ventara, Dell EMC, IBM, NetApp.(kionetworks, 2019).

La idea principal de un centro de datos es alojar a los servidores y este sea un hotel lujoso para ellos, no siempre es así existen diferentes tipos de centros de datos cada uno con la tecnología diferente, el personal del centro de datos debe velar que los servidores se encuentren actualizados y funcionen perfectamente tanto en Software como hardware, los servidores deben tener un ambiente ideal para su funcionamiento como la seguridad y refrigeración. Los servidores son equipos robustos en recursos de cómputo como CPU,

RAM, Nic`s, Hba`s, Discos, existen diferentes modelos Torre, Blade, Rack como se muestra en la Figura 3, entre los más comunes fabricantes tenemos Lenovo, Hp, Dell, Cisco.



Figura 3.- Modelos de Servidores
Fuente: (Datos101, 2022)

En cambio, por su lado PaaS la plataforma como servicio es un nivel superior, un ambiente en la nube donde el usuario permite desarrollar, gestionar y distribuir sus aplicaciones. Sus principales funciones son: (IBM, 2020)

- En la plataforma como servicio se puede probar, desarrollar y alojar aplicaciones, el mejor ejemplo del uso de la plataforma es cuando se ejecutan contenedores.
- Permite a las empresas dedicarse el 100% al desarrollo, los recursos de la infraestructura son administrados por el proveedor del servicio.
- El proveedor del servicio es el encargado de gestionar la seguridad, así como el funcionamiento del sistema operativo, y las copias de seguridad.
- Facilitar el acceso a los proveedores de soporte si los equipos se operan en remoto.

1.1.3 Contenedores.

Cuando se habla de contenedores prácticamente se habla de Dockers (Docker Containers) en idioma tecnológico es un sinónimo por ser este el motor más usado, para la explicación de contenedores de software es muy didáctico usar como referencia a los contenedores que son transportados por los barcos de un lugar a otro, no sabemos lo que está en su interior, pero lo más importante es que sean transportados de un lado a otro, así son los contenedores de software en su interior existen todas las dependencias que son necesarias para que la aplicación se ejecute, como su código, todas las librerías necesarias del sistema, el entorno de ejecución y el tipo de configuración. (RODRÍGUEZ, 2019).

Es muy común que se confunda los contenedores con una máquina virtual, ya que en realidad son conceptos similares, pero funcionan de manera diferente. A su vez que una máquina virtual ejecuta su propio Sistema Operativo alojado en un entorno controlado por un hipervisor; en cambio un contenedor comparte recursos de sistema operativo y kernel mediante un virtualizador de sistema operativo.

Se puede concluir que las máquinas virtuales utilizan una abstracción de hardware mientras que los contenedores utilizan una abstracción de sistema operativo, siendo esta más eficiente en la utilización de recursos de cómputo como se puede apreciar en la Figura 4. (NetApp, 2021)

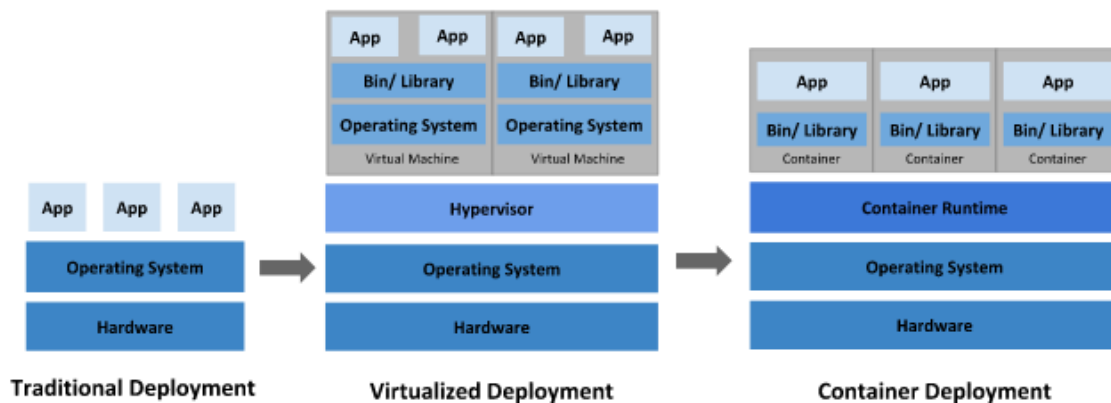


Figura 4.- Diferencia entre VM y Contenedores.
Fuente: (kubernetes, 2014)

Ahora que ya se tiene claro el concepto de los contenedores o Dockers, los mismos se pueden ir creando o ejecutando de acuerdo con las necesidades, la problemática se da cuando se tienen cientos de ellos lo que implica complejidad en su administración y control del ciclo de vida. Es por lo que para ambientes de esta escala se desarrollaron orquestadores de contenedores; estos pueden optimizar la programación, escalar la capacidad de las aplicaciones, balancear la carga, manejar esquemas de alta disponibilidad y tolerancia a fallos, gestionar redes definidas por software. En la Tabla 3 se listan los orquestadores más populares para el manejo de contenedores o Dockers. (Asad Ali, 2020)

Item	Descripción.
Kubernetes	Fue diseñado por Google y es una plataforma de código abierto en la actualidad es desarrollada por Cloud Native Computing Foundation
OpenShift Container Platform	Desarrollado por Red Hat compatible Docker y Kubernetes.
Nomad	Es considerado un orquestador para ejecutar cargas de trabajo simple, de fácil uso para implementar y administrar contenedores.
Docker Swarm	Esquema de orquestación de contenedores nativo de Docker. Es una muy buena opción para ambientes de Prueba.
AKS	AKS es un servicio de Kubernetes administrado por el proveedor Azure. Funciona en modalidad serverless.
AWS EKS	EKS de AWS es un servicio de Kubernetes administrado. Funciona en modalidad serverfull con máquinas virtuales sobre el servicio de AWS EC2 como en modalidad serverless con el servicio de AWS Fargate.

Tabla 3.- Orquestadores de Dockers.

Fuente:(Asad Ali, 2020)

1.2 ESTADO DEL ARTE

1.2.1 Antecedentes.

A medida que la tecnología avanza las exigencias del mercado cada vez son más fuertes, en un ambiente tan competitivo como el tecnológico si los proveedores no actualizan su portafolio de servicios pues corren el riesgo de quedarse rezagados, un análisis que justamente lo hacen en (Singh & Singh, 2017) donde escribe el futuro del uso de contenedores y Docker en la nube sobre un ambiente virtual, el documento nos da una visión de cómo las grandes marcas empiezan a dar este tipo de servicio dado a la optimización de recursos que este nos brinda, así podemos citar algunos proyectos. Vmware con sus proyectos 'Project Lightwave 'y' Project Photon' que permiten gestionar los contenedores y máquinas virtuales en una sola plataforma, IBM con su proyecto Bluemix oferta plataforma como servicio PaaS, Microsoft anuncio su lanzamiento de Windows Server Containers y Hyper-V Containers en el próximo Windows Server 2016, La mayoría de los proveedores de nube pública como IBM, Amazon Web Services (AWS), Google, Rackspace y Joyent tienen algún tipo de oferta de CaaS Contenedores como servicio.

En (Dua et al., 2014) se analiza el uso de las diferentes PaaS disponibles en las diferentes arquitecturas de proveedores existentes basándose en distintos contenedores como Linux Containers, Docker, Warden Container, Imctfy y OpenVZ, su conclusión es que el uso de contenedores es la mejor alternativa para la optimización de recursos de cómputo, los mismos les sacan una ventaja a las máquinas virtuales tradicionales en términos de rendimiento y ejecución. Se evidencia que la Capa IaaS es muy importante para la ejecución y soporte para el tiempo de vida de los contenedores.

En (Nakagawa & Oikawa, 2017) se puede observar un análisis del uso de la memoria al ejecutar contenedores en un ambiente virtual, se evidencia que la carga es ligera sobre el hipervisor, pero también observan que algunos contenedores consumen más recursos de los esperados, ante esto los investigadores proponen un método para aislar estos contenedores y limitarlos para que no usen más memoria de la entregada, este trabajo de investigación nos dará una guía del uso de contenedores y tener una idea de la demanda de memoria que estos necesitan.

Para (Kumar & Kurhekar, 2017) es muy importante la eficiencia de los recursos de virtualización usando contenedores y Docker, su conclusión más importante es la escalabilidad que llegan a tener los contenedores en este tipo de ambientes, y la alternativa de usar la tecnología de Docker en lugar de hipervisores, siendo esta tecnología Docker de menor tamaño, permiten que se inicien y ejecuten de una manera más rápida ya que al ser solamente una instancia y no todo un sistema operativo que es una máquina virtual. Los resultados son prometedores al demostrar que los contenedores son mejores ante una máquina virtual, tanto en tamaño, hardware, portabilidad y Performance.

El rendimiento de infraestructura es muy importante tenerla en cuenta cuando se habla de contenedores en (Preeth et al., 2016) precisamente realizan ese estudio, como es comportamiento de los recursos de hardware cuando ejecutan contenedores, la evaluación son los recursos de CPU y RAM, Los resultados incluyen utilización de CPU, utilización de memoria, conteo de CPU, tiempos de CPU, partición de disco, contador de E / S de red. El estudio usa un software Bonnie ++ y psutil sobre una infraestructura BareMetal, basándose en estos resultados los investigadores concluyen que los Docker funcionarían mucho mejor en un nodo virtualizado.

En (Lingayat et al., 2018) se propone la evaluación del rendimiento de Dockers en BareMetal y una Máquina Virtual, para ello proponen un sistema donde calculan el tiempo de inicio de los contenedores ejecutando un servidor Web apache sobre un entorno conocido como BareMetal y sobre una Máquina Virtual, evaluando cual es la mejor acción

sobre los dos ambientes. Ayudando a decidir en cuál de ellos se tiene un mejor rendimiento. Los resultados obtenidos dicen que en BareMetal existe un 50% mejor rendimiento que en una máquina virtual esto se debe a que al ser un entorno virtualizado no dedicado para contenedores ejecuta un hardware emulado involucrando capas adicionales.

Al momento de la implementación de contenedores es de mucha utilidad la evaluación del ambiente en los que estos se desenvuelven mejor, por es para (Abdelbaky et al., 2015) plantea un prototipo llamado C-Ports que permiten la implementación y administración de contenedores Dockers en múltiples nubes Híbridas y clústeres tradicionales. Con C-Ports los usuarios pueden utilizar el servicio de contenedores de una manera más rápida, eficiente y efectiva en una federación dinámica compuesta de hasta siete recursos heterogéneos y distribuidos, esto incluye cisco nubes (FutureSystems, Google Cloud Engine, Amazon AWS, Chameleon Cloud, y IBM Bluemix) y dos clústeres (uno en IBM y otro en la Universidad de Rutgers).

La Implementación automatizada de contenedores de software con Dockers también es una alternativa al momento de trabajar con este tipo de tecnología como lo muestran en (Kharb, 2015) el documento se enfoca en las características , ventajas, limitaciones y su rendimiento, su conclusiones arrojan que con el uso de Docker las organizaciones reducen la entrega de aplicaciones de años a meses y una eficiencia de quince veces más en el uso de los recursos informáticos, Docker hace que mucho más aplicaciones se ejecuten en el mismo hardware además de desplegar aplicaciones mucho más rápida para los desarrolladores de software.

1.2.2 Línea de Tiempo de los contenedores.

La computación basada en contenedores no es una tecnología nueva de hecho por los años 80`s ya se manejaba la idea de aislar los procesos y se veía mucho el uso del comando chroot en sistemas operativos similares a Unix, el resultado de esta operación es cambiar el directorio raíz para el proceso en ejecución actual y sus subprocesos. En la Figura 5 se muestra como los contenedores fueron evolucionando en el tiempo, en el 2013 Docker fue anunciado de manera oficial por Solomon Hykes, sin embargo, en la década de los años 2000 existieron algunos predecesores como: Jails FreeBSD, Linux VServer, Solaris Containers, OpenVZ, Linux Containers, entre otros.(Mauricio Collazos, 2019)

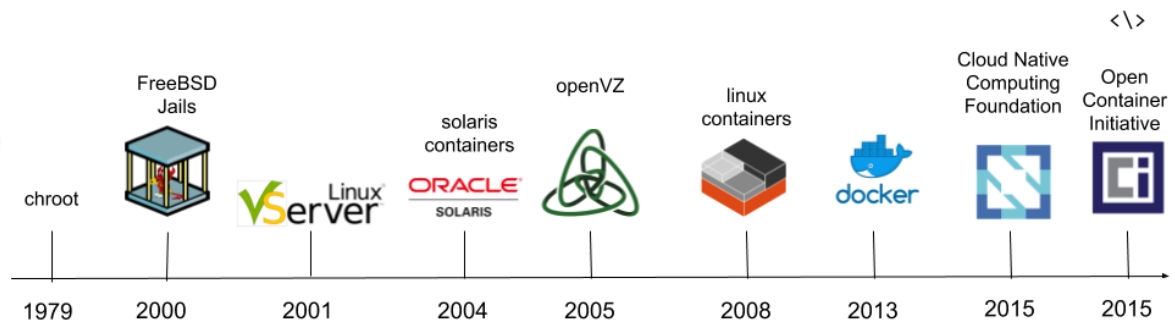


Figura 5.- Línea del Tiempo de los Contenedores
Fuente: (Mauricio Collazos, 2019)

1.2.3 Actualidad de los Contenedores.

Docker en la actualidad es uno de los motores más usado en el mundo para ejecutar contenedores, con miles de partners asociados que contribuyen con su desarrollo. Ahora que ya se ha hablado de los conceptos fundamentales de los contenedores es el momento de conocer la actualidad de la arquitectura de Docker-Container, la arquitectura comprende tres componentes fundamentales. (Chowdhury, 2021)

- Docker Daemon (dockerd): Como su nombre lo indica en un demonio que se ejecuta en segundo plano esperando las instrucciones del cliente.
- Docker Client: Comúnmente conocido como CLI que viene ser la interfaz de línea de comandos y es el encargado de transportar los comandos ingresados por los usuarios.
- Rest api: Es el nexo o comunicación entre el Dockerd y el Client. Al digitar una instrucción en la CLI pasa a través de la API para finalmente llegar al dockerd.

Así en la Figura 6 se puede evidenciar lo que sucede cuando se ejecuta un contenedor Docker. El usuario ejecuta el comando “run” para arrancar el contenedor desde una imagen con el nombre “hello-world”, se da la instrucción al demonio el mismo busca la imagen “hello-word” y la ejecuta, el demonio realiza una búsqueda local de la imagen en el caso de no encontrarla procede hacer una búsqueda en los repositorios públicos registrados y procede hacer una copia de la imagen en un repositorio local, finalmente con la imagen local el demonio ejecuta el contenedor.

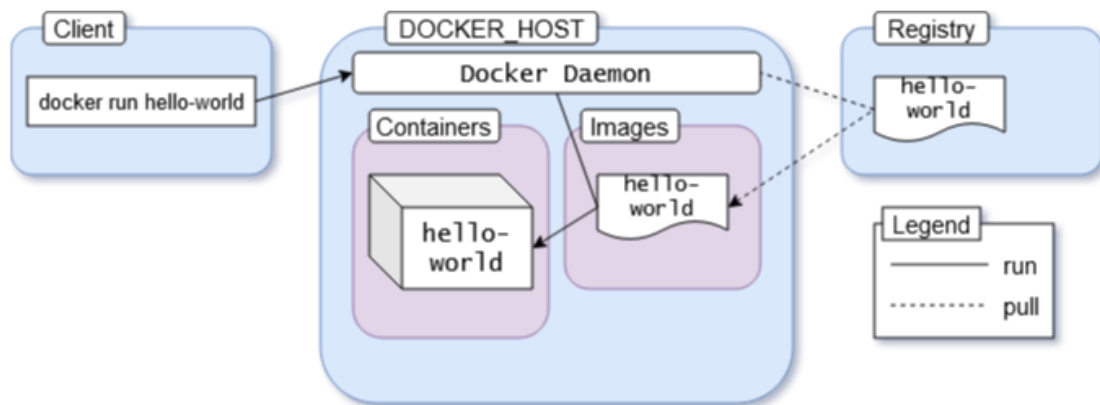


Figura 6.- Arquitectura Docker.
 Fuente: (Chowdhury, 2021)

A los tres paquetes fundamentales en la actualidad Docker las llama como Docker-Engine, para la búsqueda y descarga de las imágenes de un registro público Docker hace un “pull” a Docker-Hub que es el repositorio más grande del mundo para compartir las imágenes. Docker en la actualidad también maneja proyectos como Docker-Desktop que es una aplicación de fácil instalación en Windows o Mac que permite al usuario crear y compartir microservicios y aplicaciones en contenedores, el paquete de Docker-Desktop incluye, Docker content trust, Docker cli client, Docker Compose, Docker Engine, Kubernetes. Docker-Compose es una herramienta con la cual podemos crear un Stack de servicios donde se ejecutan aplicaciones de varios contenedores, Compose usa un archivo YAML donde se escribe la configuración de los servicios, con un solo comando puedes ejecutar todo un Stack de contenedores y servicios. (Dockers Doc, 2022)

En la actualidad Docker-Swarm es la solución nativa para la orquestación de Dockers-Containers su función es convertir un conjunto de host de Docker en un único servidor virtual, haciendo posible levantar un clúster de Docker bajo la orquestación de Swarm. La arquitectura de Swarm se aprecia en la Figura 7, donde constan de 3 nodos mánager que son los encargados del plano de control y 3 nodos Workers donde se ejecutan las cargas de trabajo.

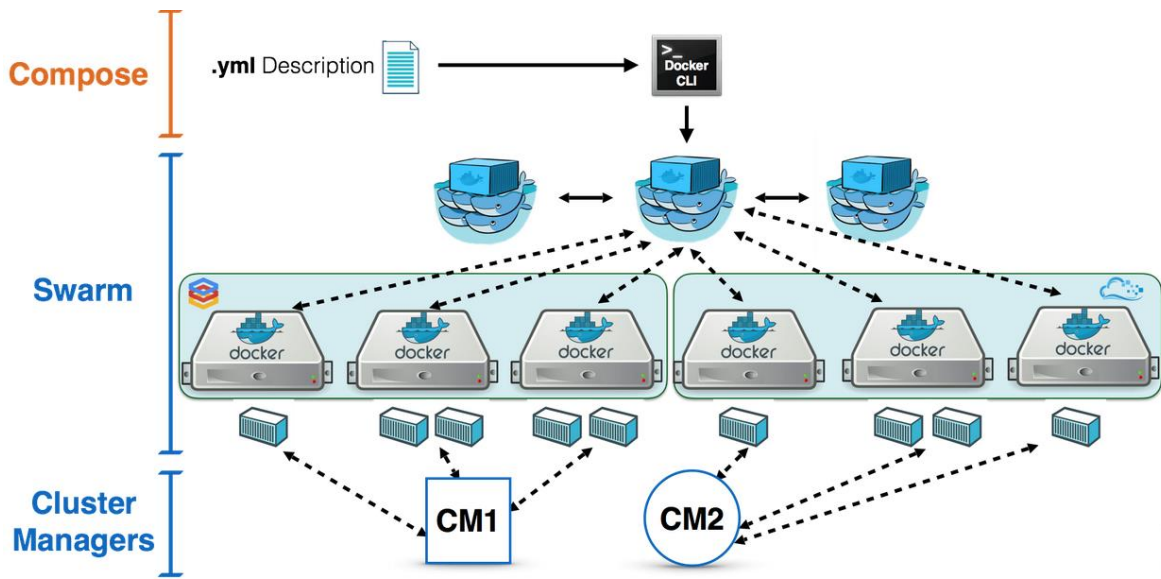


Figura 7.- Orquestación Docker-Swarm
Fuente:(Speroni, 2018).

CAPÍTULO II

2. DISEÑO DEL PROTOTIPO

En el capítulo II se realizará el diseño del prototipo así como las configuraciones de las diferentes capas que conforman la puesta en marcha de la Plataforma, se encontrara el diseño en alto nivel y las consideraciones para el diseño de bajo nivel, seguidamente se realizará la configuración de la Plataforma como servicio (PaaS) orientada a Contenedores como servicio (CaaS), por último se encontrara el diseño y configuración de la capa virtual y capa física que conforman la infraestructura como servicio (IaaS).

2.1 Introducción al diseño

En este desarrollo se considera todas las fases de diseño de un servicio PaaS basado en contenedores ejecutándose sobre un entorno IaaS. El objetivo principal es ofrecer capacidad de cómputo como recurso a los clientes finales eliminando la responsabilidad de administración del motor de contenedores, servidores e infraestructura. Este tipo de servicio va a ofrecer características de elasticidad de crecimiento como facilidad de la actualización de las aplicaciones.

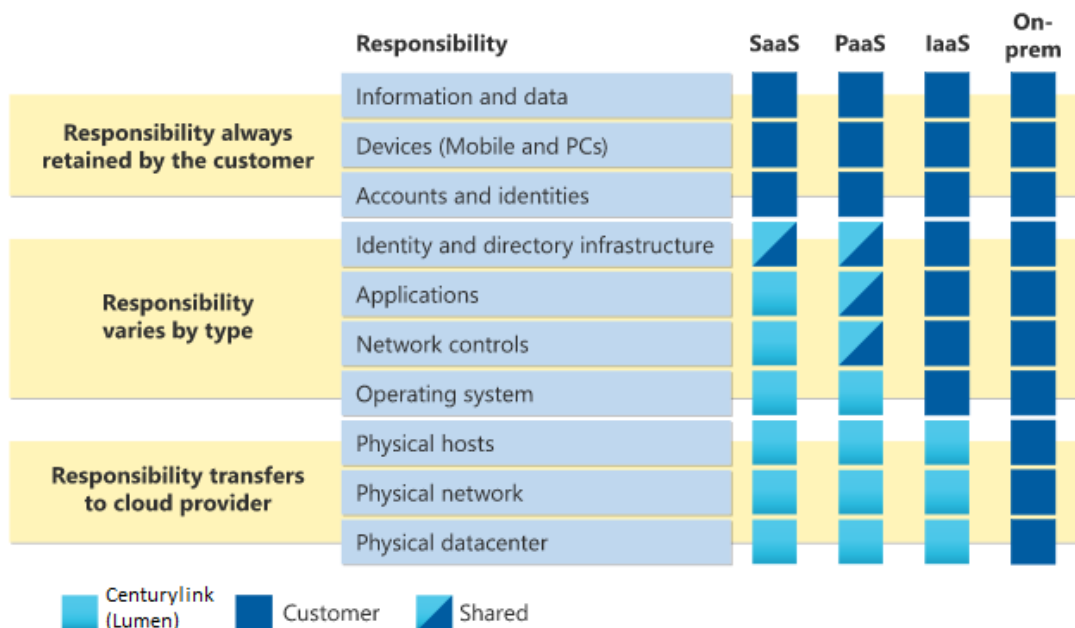


Figura 8.- Responsabilidad Compartida.
Fuente:(Berry, 2022)

Cuando se habla de un esquema de provisión de computación en la nube se hace relación a un modelo de responsabilidad compartida en donde ciertas capas que componen el servicio son responsabilidad del proveedor de nube (Cloud Service Provider) y otras son responsabilidad del cliente. Para este caso en particular el servicio CaaS está conformada por las siguientes capas como se muestra en la Tabla 4 y su correspondiente responsabilidad ver Figura 8. (AWS, 2022).

Capa	Descripción.
Contenedores.	Unidad de software que empaqueta el código de la aplicación del cliente como sus dependencias. La responsabilidad de la aplicación que está en el contenedor es de cliente.
Ejecución de contenedores. "Run Time"	Es una abstracción o virtualización de recursos de sistema operativo para entregar aplicaciones, el run time virtualiza el sistema operativo como un hipervisor virtualiza un servidor físico.
Plataforma.	La plataforma esencialmente permite desarrollar las aplicaciones con el software instalado en el Sistema Operativo. Se puede entregar con Sistemas operativos Linux o Windows.
virtualización de servidores.	La virtualización es la base del modelo, conlleva a la abstracción de recursos de hardware mediante un hipervisor instalados en los hosts del clúster.

Tabla 4.- Capas del modelo propuesto.

Fuente: Propia.

Tomando en cuenta las capas necesarias para el servicio CaaS se modela en la Figura 9, un diseño de alto nivel del servicio, el mismo que considera componentes tecnológicos necesarios para su despliegue.

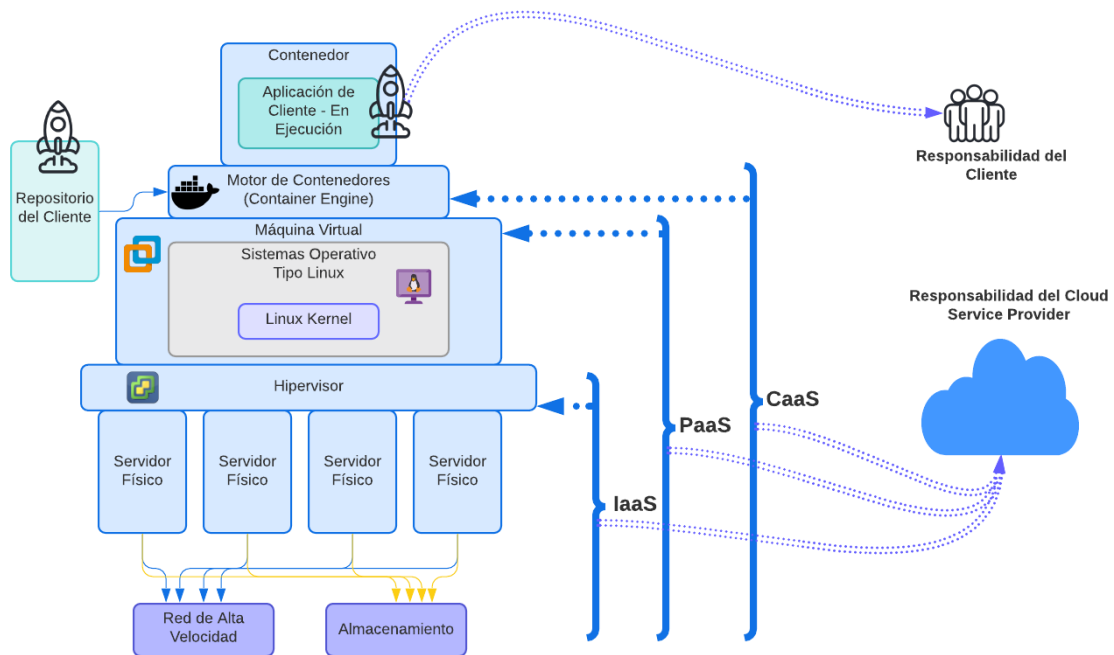


Figura 9.- Diseño en alto Nivel.
Fuente: Propia.

Para toda la capa de virtualización se elige a VMware como software, en términos comerciales la empresa Centurylink (Lumen) es partner de la marca y se tienen todo el licenciamiento y soporte necesario, en términos técnicos VMware es la compañía más grande del mundo en proveer todo un Stack para la virtualización.

Se usará Docker por ser el motor más usado para ejecutar contenedores, se encuentra en constante desarrollo y se puede contar con soporte de millones de individuos en el mundo. Así como su simpleza en el uso, y despliegue de contenedores usando este motor.

2.2 Consideraciones de diseño.

En esta sección se encontrarán los criterios a considerar para el diseño del prototipo, así como la topología final en bajo nivel de la solución.

2.2.1 Facilidad de despliegue de aplicaciones.

El servicio está considerado para permitir que los usuarios finales o los clientes puedan desplegar aplicaciones de manera ágil y en corto tiempo, es por esto por lo que es

necesario que el servicio contemple tanto el uso como repositorios locales como de repositorios externos para almacenar las imágenes de los contenedores.

2.2.2 Optimización de cómputo.

El servicio debe permitir que los clientes puedan solicitar capacidad de cómputo (CPU y RAM) de manera flexible para la ejecución de sus aplicaciones de contenedores. Desde la perspectiva del proveedor de servicio es deseable que la capacidad de cómputo tanto de máquinas virtuales como de servidores físicos sea aprovechada en su totalidad siempre y cuando no degrade el rendimiento de ejecución de los contenedores.

2.2.3 Administración y orquestación.

El servicio debe permitir que el personal del proveedor de servicio pueda administrar tanto por la línea de comandos como por una interfaz gráfica en un navegador web la capa de ejecución de contenedores. A su vez el servicio debe permitir la orquestación de contenedores esto quiere decir que los mismos pueden ser desplegados como servicios o Stack de aplicación.

2.2.4 Tolerancia fallos.

Se espera del servicio que con respecto a la capa de ejecución de contenedores se pueda contar con mecanismos de tolerancia a fallos para las aplicaciones; esto debe ser considerado tanto para las capacidades de cómputo como las publicaciones a través de la red.

2.2.5 Alta disponibilidad.

Se espera que el servicio en la capa virtual cuente con mecanismos de alta disponibilidad esto para garantizar que los recursos de la capa de ejecución de contenedores estén disponibles.

2.3 Construcción de Diseño.

Se inicia la construcción del diseño del servicio, dividiendo la capa del motor de contenedores en dos planos: el plano de administración y orquestación y el plano de ejecución de cargas de trabajo. Esto se puede apreciar en la Figura 10.

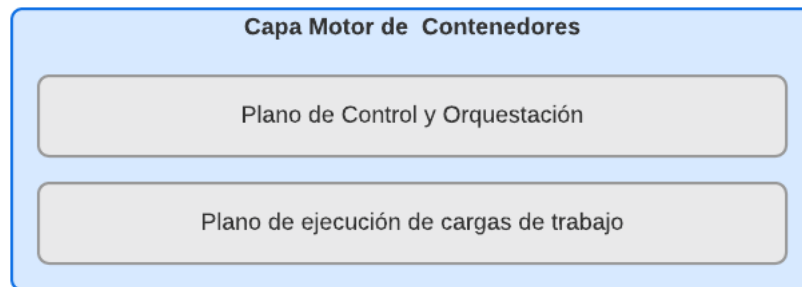


Figura 10.- Capa Motor de Contenedores.
Fuente: Propia.

Los componentes de cada plano se ejecutarán sobre el esquema de IaaS sobre máquinas virtuales con sistema Operativo Linux CentOS, y los servicios de cada plano en contenedores basados en Docker. Tomando en cuenta que la integración del usuario es a través de un repositorio, se diseña la solución para que pueda integrarse a nivel de la capa del motor de contenedores con repositorios externos los que pueden ser provistos por el Cloud Service Provider (CSP) o por el cliente. Para garantizar el criterio de tolerancia a fallas tanto para las cargas de trabajo que se ejecutarán en el servicio como para los servicios que habilitarán este, se considera el despliegue de los planos de control sobre tres máquinas virtuales. Con base en esta consideración se elimina la existencia de un único punto de falla, se aísla los fallos, al tener los mismos servicios y cargas de trabajo desplegados en varios nodos de cómputo, existe una contención de fallos, ya que cada carga de trabajo y servicio si bien son idénticos no tienen dependencia entre ellos.

Cumpliendo el criterio de orquestación a nivel de contenedores, se considera la utilización de la solución de orquestación nativa de Docker que se conoce como Docker Swarm, sobre la cual se despliega la arquitectura del plano de control y orquestación tanto como el plano de ejecución de las cargas de trabajo, en lo que considera un Enjambre, Clúster o Swarm de Docker.

Dentro de la arquitectura de Docker Swarm, con respecto al plano de control, las máquinas virtuales sobre las que se ejecuta, se denominan nodos "Managers" o nodos "Máster"; mientras que las máquinas virtuales donde se ejecutan las cargas de trabajo se denominan

“Workers”. Para garantizar la tolerancia a fallos dentro del plano de control, Docker Swarm utiliza un algoritmo de consenso o quorum llamado Raft para determinar el estado global del clúster. El criterio del algoritmo determina, que se soporte $(N-1) / 2$ fallas y se requiere que la mayoría de los managers tengan la misma información del clúster, lo que se traduce en considerar $(N/2) + 1$ managers en la solución como se muestra en la Tabla 5.

Nodos Manager - Swarm	Mayoría	Tolerancia a fallos
3	2	1

Tabla 5.- Nodos Manager Swarm
Fuente: Propia.

Para la solución propuesta, con la consideración de tolerancia a fallos, se contarán con tres managers, de los cuales dos tendrán la mayoría del consenso del clúster, y se contará con uno para tolerancia a fallos. El mismo criterio aplica, para garantizar la información de los contenedores que ejecutarán las cargas de trabajo, por lo que se tendrá dos Workers como mayoría de consenso y uno de tolerancia a fallos. Esto se puede apreciar en la Figura 11.



Figura 11.- Nodos Manager y Nodos Workers.
Fuente: Propia.

Como ya se mencionó, los servicios que manejan y administran el clúster en el plano de control también son colocados en un esquema de tolerancia a fallos como se muestra a continuación en la Figura 12.

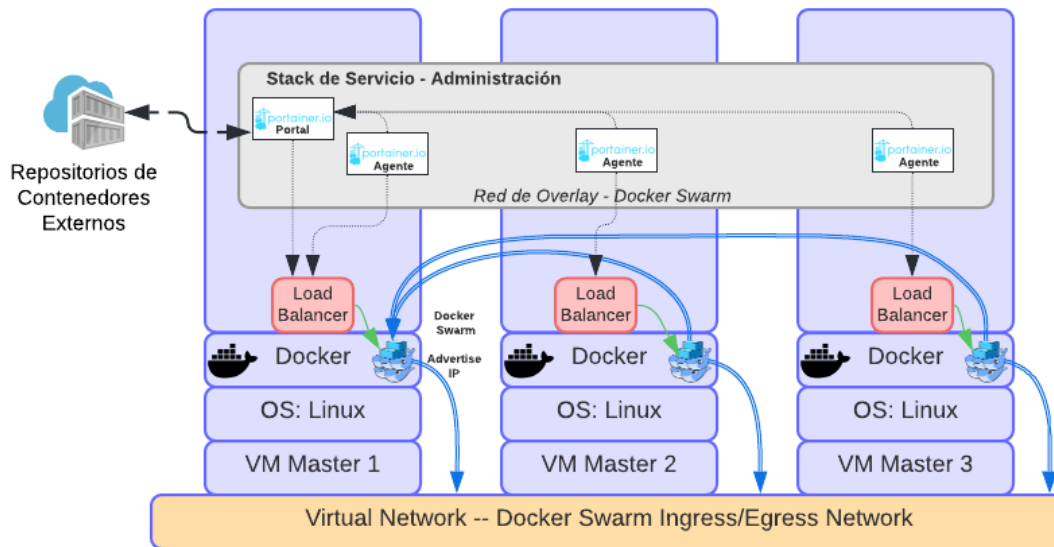


Figura 12.- Funcionamiento del Plano de Control.
 Fuente: Propia.

Cada nodo mánager o máster como cada nodo Worker utilizan una red de ingress/egress definida sobre una virtual network a nivel de la virtualización de infraestructura tanto para comunicar el plano de control como para permitir comunicaciones hacia afuera del clúster como para publicar servicios o conectar repositorios externos. Las funciones básicas de orquestación Docker Swarm se ejecutan como una funcionalidad de Docker Engine, mientras que las funciones de balanceo de carga de red dentro del clúster se ejecutan en conjunto con Docker Swarm sobre cada nodo. Todas las comunicaciones que salen de los contenedores hacia el exterior se comunican a través del balanceador de carga usando la red de Ingress/Egress.

Dentro del plano de control, de manera adicional, se utiliza una herramienta de administración de contenedores (Docker) y de clústeres de contenedores (Docker Swarm), que es Portainer, la cual es desplegada sobre una red definida por software usando un esquema de conexión (controlador) overlay, que permite comunicación dentro del clúster. Esta herramienta utiliza un componente central y portal que se deben ejecutar en un nodo mánager a la vez, y usa agentes que deben ser desplegados en todos los nodos managers como Workers.

De igual manera, para cumplir con el criterio de facilidad de despliegue de aplicaciones, se considera la integración de Portainer con repositorios externos los cuales podrán ser vinculados con los que tenga el cliente, o nuevos que cree el CSP. A continuación, en la

Figura 13 se ilustra el diseño considerando una carga de trabajo de referencia sobre los nodos Workers.

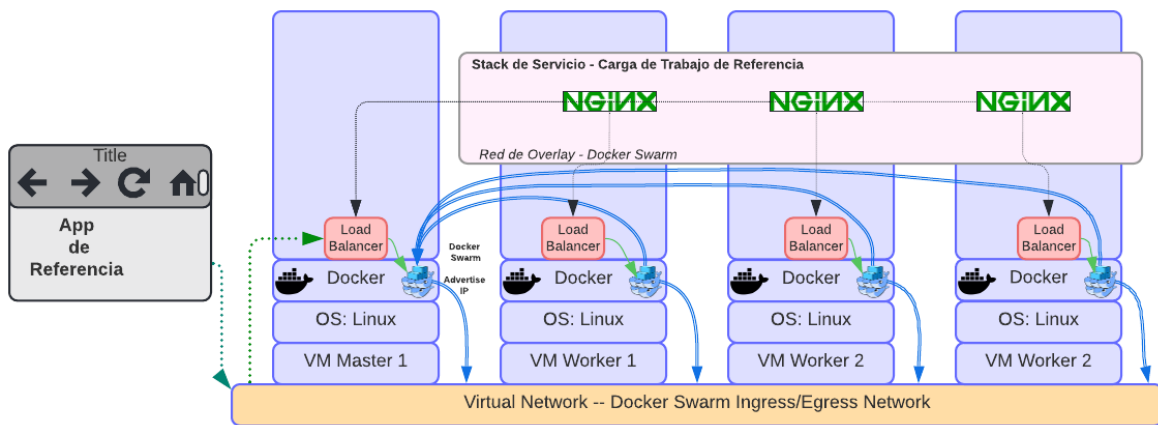


Figura 13.- Cargas de trabajo.
Fuente: Propia.

Por último, para cerrar el diseño, se plantea el esquema de alta disponibilidad para disponibilidad los nodos managers y Workers sobre el entorno IaaS. Se considera que cada máquina virtual manager debe estar aislada en un host físico distinto y lo mismo se aplica para las máquinas virtuales Workers. Esto se logra aplicando el concepto denominado regla de afinidad en la capa de virtualización como se aprecia en la Tabla 6.

Rol	Nodos	Host 1	Host 2	Host 3
Manager	3	1	1	1
Worker	3	1	1	1

Tabla 6.- Distribución de Nodos máster y Nodos Worker en cada host físico.
Fuente: Propia.

El diseño planteado considera tres servidores físicos para el servicio como se muestra en la Tabla 7.

Cantidad de hosts de la plataforma	% de utilización máxima de recursos antes de agregado de nuevo host
3 hosts	66%

Tabla 7.- Host asignados al diseño.
Fuente: Propia.

Se toma en cuenta que la capacidad ociosa en mínimamente uno de los hosts del clúster de virtualización de infraestructura debe ser igual o mayor que los recursos de la máquina

virtual más grande, a nivel recursos, definida en el clúster. El objetivo de esta definición es poder brindar alta disponibilidad en caso de falla del host que contiene la VM en cuestión, para que la misma pueda ser migrada a otro host del clúster.

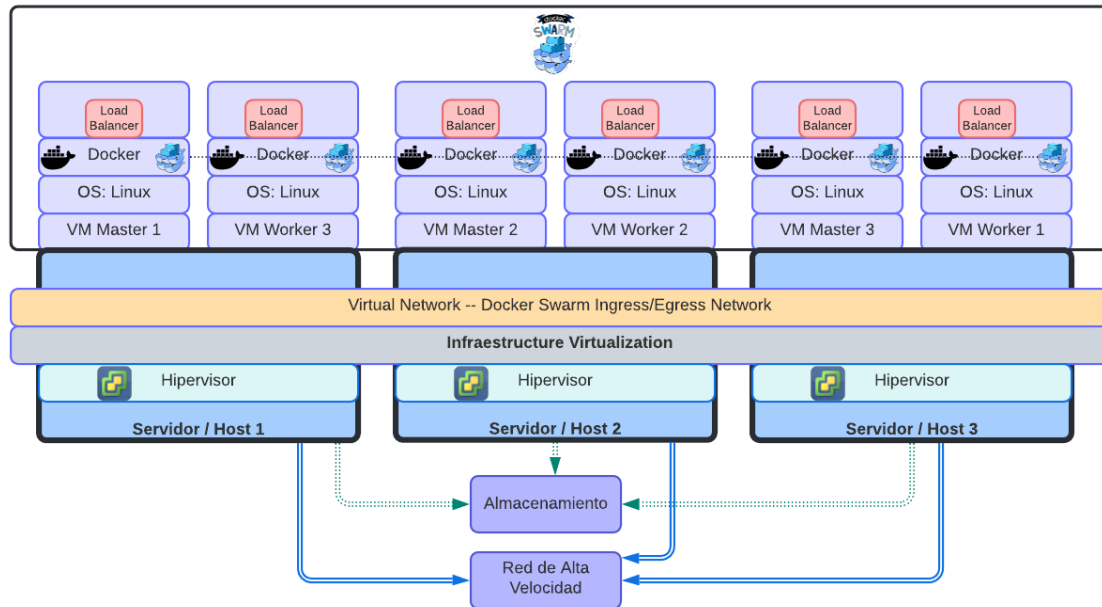


Figura 14.- Diseño del prototipo en bajo nivel.
Fuente: Propia.

En base al diseño de la Figura 14, se detalla la configuración del servicio.

2.4 Configuración de PaaS

En la presente sección se evidenciará la preparación previa que deben ser configurados los nodos o máquinas virtuales, seguidamente la instalación del motor Docker y las configuraciones del orquestador Docker-Swarm del prototipo, por último, la instalación y configuración de la plataforma de gestión del prototipo.

2.4.1 Preparación de nodos o máquinas virtuales.

Antes de la instalación de los componentes de la solución, lo fundamental es definir los prerrequisitos que se necesitan para el despliegue del ambiente, con criterio en el diseño de la solución, se necesita desplegar previamente 6 nodos (3 máster y 3 Workers) con sus respectivos recursos y sistema operativos, en la Figura 15 se muestra la matriz de compatibilidad de Docker con el sistema operativo correspondiente.

Platform	x86_64 / amd64	arm64 / aarch64	arm (32-bit)	s390x
CentOS	✓	✓		
Debian	✓	✓	✓	
Fedora	✓	✓		
Raspbian			✓	
RHEL				✓
SLES				✓
Ubuntu	✓	✓	✓	✓
Binaries	✓	✓	✓	

Figura 15.- Plataformas compatibles con Docker.
Fuente:(Dockers Doc, 2022)

Para el presente trabajo y en base a la experiencia con las diferentes distribuciones de Linux se trabajará con CentOS, en la Tabla 8 se evidencia las características de los nodos creados, sacando ventaja a la virtualización se creará un nodo base y posteriormente se procederá a clonarlo cinco veces.

Ítem	Descripción
# de Nodos	6
Sistema Operativo	CentOS
Versión	8.5.2111
RAM	12 GB
CPU	4
Disco	140 GB
Docker-master01	Ip Adm.: 10.225.49.241/21 Ip Wrk: 172.10.10.16/24
Docker-master02	Ip Adm.: 10.225.50.196/21 Ip Wrk: 172.10.10.17/24
Docker-master03	Ip Adm.: 10.225.50.197/21 Ip Wrk: 172.10.10.18/24
Docker-worker01	Ip Adm.: 10.225.50.198/21 Ip Wrk: 172.10.10.19/24
Docker-worker02	Ip Adm.: 10.225.50.199/21 Ip Wrk: 172.10.10.20/24
Docker-worker03	Ip Adm.: 10.225.50.200/21 Ip Wrk: 172.10.10.21/24

Tabla 8.- Recursos de Máquinas Virtuales.
Fuente: Propia.

2.4.2 Instalación de Dockers.

Para la instalación de Dockers se trabajará en un nodo base para aprovechar las propiedades de la virtualización y clonarlo 5 veces, de esta manera se tendrá los 6 nodos requeridos para posteriormente trabajar en la configuración de Swarm.

Docker requiere de la instalación algunos componentes y librerías para su funcionamiento en la Tabla 9 se resumen los pasos y requisitos que se siguieron para una perfecta instalación.

Requisito #	Descripción
1	El repositorio centos-extras debe estar habilitado. Este repositorio está habilitado de forma predeterminada, pero si lo ha deshabilitado, debe volver a habilitarlo.
2	Desinstalar versiones antiguas, si están instalados, desinstálelos junto con las dependencias asociadas.
3	Instale los paquetes yum-utils (que proporciona la utilidad yum-config-manager) y perl .
4	Instale la última versión de docker engine y containerd.
5	Start Docker.
6	Verifique que el motor Docker esté instalado correctamente ejecutando la imagen hello-world.

Tabla 9.- Recomendación de instalación Docker.
Fuente:(Dockers Doc, 2022)

Continuando con el proceso de preparación del ambiente es necesario instalar Docker-Compose se puede realizar la descarga del binario desde GitHub, Github es un sitio donde los desarrolladores de todo el mundo suben el software con el objetivo de contribuir con la comunidad o mejorar el código. Con el uso del comando “curl” se procede a descargar los archivos binarios para la instalación, los únicos prerrequisitos que se requiere para la instalación de Docker-Compose es tener instalado Docker-Engine y configurar un usuario no-root con permiso para ejecutar comandos sudo, en la Tabla 10 se establece los pasos para la instalación de Docker-Compose.

Paso #	Descripción
1	Descargamos la versión estable actual de docker compose.
2	Aplicamos permisos ejecutables al binario.

Tabla 10.- Instalación de Docker-Compose
Fuente: (Dockers Doc, 2022)

La instalación de Dockers y sus componentes podría ser un poco engorroso y tomar mucho tiempo si lo hacemos a mano corriendo comando por comando, por ese motivo se procederá a crear un archivo .sh que contiene las instrucciones en lenguaje bash, la idea es que con un solo comando ejecutar todas las instrucciones para la instalación de Dockers y todos los componentes que fueron descritas anteriormente.

Para eso se creará un folder donde se va a almacenar el archivo .sh, haremos el uso de la siguiente sentencia: “mkdir /\$home/startup”, seguido se descargará el archivo Docker-Startup.sh en /\$home/startup y se otorgará los permisos de ejecución: “chmod +x Docker-Startup” como se muestra en la Figura 16, por último, ejecutamos “./docker-startup.sh”.

```
[root@170-187-136-202 startup]# nano docker-startup.sh
[root@170-187-136-202 startup]# chmod +x docker-startup.sh
[root@170-187-136-202 startup]# ls -l
total 4
-rwxr-xr-x. 1 root root 1751 Jan 20 03:34 docker-startup.sh
```

Figura 16.- Configuración de Permisos.
Fuente: Propia.

En la Figura 17, se muestra algunos segmentos de código bash para la instalación y afinamiento de Docker.

```

docker-compose.yml • ! Untitled-1 • $ docker-startup.sh X Dockerfile
C: > Users > CSTI_SANCHEZ.ANDRES > Downloads > $ docker-startup.sh
1  #!/usr/bin/bash
2  #Propósito - script para arrancar las configuraciones base de docker
3  #Autor: David Zambonino & Andres Sanchez
4  #cambio de nombre
5  hostname docker-master01
6  echo "Hostname updated"
7  #Script debe ejecutarse con root o con un sudoer
8  sudo yum update
9  echo "Operative System updated"
10 #Instalación docker
11 y yum remove docker \
12     docker-client \
13     docker-client-latest \
14     docker-common \
15     docker-latest \
16     docker-latest-logrotate \
17     docker-logrotate \
18     docker-engine
19 yum install -y yum-utils
20 yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
21 yum install -y perl
22 yum install -y docker-ce docker-ce-cli containerd.io
23 systemctl start docker
24 systemctl enable docker.service
25 systemctl enable containerd.service
26 echo "Docker installed"

```

Figura 17.- Instalación de Docker.
Fuente: Propia.

En cambio, en la Figura 18, se muestra el código usado para instalar Docker-Compose.

```

28 #instalación de docker compose
29 curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local.
30 chmod +x /usr/local/bin/docker-compose
31 docker-compose version
32 echo "Docker-Compose installed"

```

Figura 18.- Instalación de Docker-Compose.
Fuente: Propia.

De esta manera en la Figura 19 se puede evidenciar que el servicio Docker se encuentra activo y corriendo.

```

Docker version 20.10.9, build c2ea9bc
[root@docker-master01 ~]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2022-01-21 16:55:17 -05; 1 weeks 5 days ago
     Docs: https://docs.docker.com
   Main PID: 1335 (dockerd)
      Tasks: 27
     Memory: 1.9G
    CGroup: /system.slice/docker.service
            └─ 1335 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
               219422 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 9001 -container-ip 172.18.0.4 -container-port 9001
               219428 /usr/bin/docker-proxy -proto tcp -host-ip :: -host-port 9001 -container-ip 172.18.0.4 -container-port 9001

```

Figura 19.- Docker activo y corriendo.
Fuente: Propia.

Como se muestra en la Figura 20 se tiene Docker-Compose instalado.

```
[root@docker-master01 ~]# docker-compose --version
docker-compose version 1.29.2, build 5becea4c
```

Figura 20.-Version de Docker-Compose.
Fuente: Propia.

2.4.3 Configuración de Docker-Swarm.

Como se ha visto en secciones anteriores Docker-Swarm nos va a permitir realizar la orquestación de los contenedores y crear clústeres administrados, el plano de control nos ayudará con la abstracción de las aplicaciones sin la necesidad de manejar nodos específicos, así como tener una alta disponibilidad de las aplicaciones, equilibrio de carga y la tolerancia a fallas. Docker-Swarm viene ya instalado nativamente con Docker así que se procederá solamente con la configuración.

En la Tabla 11 se describe el proceso de configuración, que consiste en advertir al nodo base como Máster y mediante un token agregar dos nodos adicionales como Másteres, el mismo proceso se ejecuta en los tres nodos Workers, se los registra mediante un token Worker que se genera a través del nodo máster.

Item	Descripción
Declaramos al nodo Base como Máster.	docker swarm init --advertise-addr <MANAGER IP>
Generamos el token para registrar nodos Máster Adicionales.	docker swarm join-token manager
Copiamos y ejecutamos el token en los nodos Máster adicionales.	docker swarm join --token <TOKEN-ID>
Generamos el token para registrar nodos Worker.	docker swarm join-token worker
Copiamos y ejecutamos el token en los nodos worker.	docker swarm join --token <TOKEN-ID>

Tabla 11.- Resumen de configuración Docker-Swarm
Fuente: Propia.

En la Figura 21, evidenciamos que Swarm está activo y consta de 3 nodos Másteres y 3 nodos Workers en total administra 6 nodos.

```
[root@docker-master01 ~]# docker info | grep Swarm && docker info | grep Managers && docker info | grep Nodes
Swarm: active
Managers: 3
Autolock Managers: false
Nodes: 6
```

Figura 21.- Numero de nodos administrados en el Clúster.
Fuente: Propia

Y donde los másteres tienen el siguiente direccionamiento según la Figura 22:

```
Manager Addresses:
172.10.10.16:2377
172.10.10.17:2377
172.10.10.18:2377
```

Figura 22.- Direccionamiento Nodos Máster.
Fuente: Propia.

Así como el estado de salud de todos los nodos como se aprecia en la Figurar 23:

```
[root@docker-master01 ~]# docker node ls
ID                                HOSTNAME          STATUS    AVAILABILITY    MANAGER STATUS    ENGINE VERSION
fjzi0caw2kaeksg24337761z3 *    docker-master01  Ready    Active           Leader             20.10.9
sqhxf12zikpkqk2zz1zacckbb      docker-master02  Ready    Active           Reachable          20.10.9
ilnq778e0ldczhw7lp3ra7lo2     docker-master03  Ready    Active           Reachable          20.10.9
m945zm3sjf2o0n7pgnwdirvwi     docker-worker01  Ready    Active           -                  20.10.9
skfo6b9twpyulfez4qdaaty1m     docker-worker02  Ready    Active           -                  20.10.9
nl6jbap4kvd867izgivvpz6mr     docker-worker03  Ready    Active           -                  20.10.9
```

Figura 23.-Estado de salud de Clúster.
Fuente: Propia.

2.4.4 Instalación y configuración de la plataforma de gestión.

Docker-Container es una herramienta ideal y la más común para desarrollo de contendores, el interactuar con la CLI (Interfaz de Línea de Comandos) puede resultar en ocasiones nada amigable, en ese caso sería una muy buena opción manejar los contendores desde una consola web, Portainer Community Edition (CE) en una plataforma que brinda esta facilidad y en la Figura 24 se puede observar la compatibilidad con las plataformas.

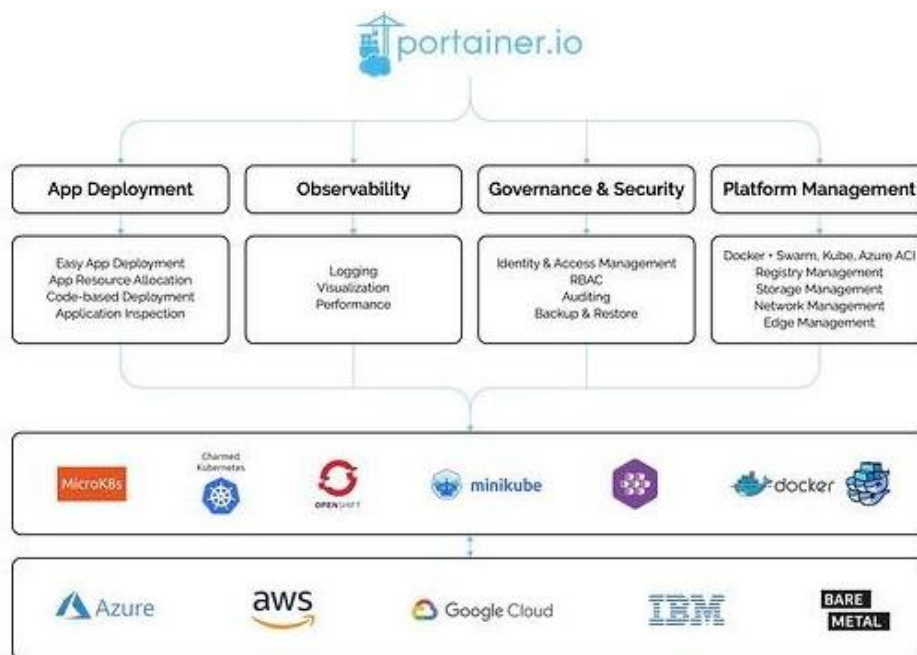


Figura 24.- Funcionalidades Portainer.
Fuente:(Cresswell, 2021)

Portainer es una plataforma orientada a la administración de la funcionalidad de Docker que brinda a los desarrolladores una interfaz gráfica para crear, publicar imágenes Docker, implementar y administrar las aplicaciones, y no solo eso Portainer tiene la capacidad de monitorear y medir el rendimiento de la aplicación ejecutada en un contenedor.

Portainer es un aliado de lujo para orquestadores como Docker-Swarm, la experiencia de la administración de un grupo de nodos con Swarm usando Portainer es muy similar al usar Docker independientes, el usuario puede determinar cómo implementar la aplicación, configurar las redes y todos los datos necesarios para tener una persistencia. Portainer trabaja traduciendo todo lo que sucede en Swarm trayéndolo a la consola gráfica y también instruye a Swarm con las instrucciones dadas por el usuario, la arquitectura de Portainer se describe en la Figura 25.(Portainer, 2022)

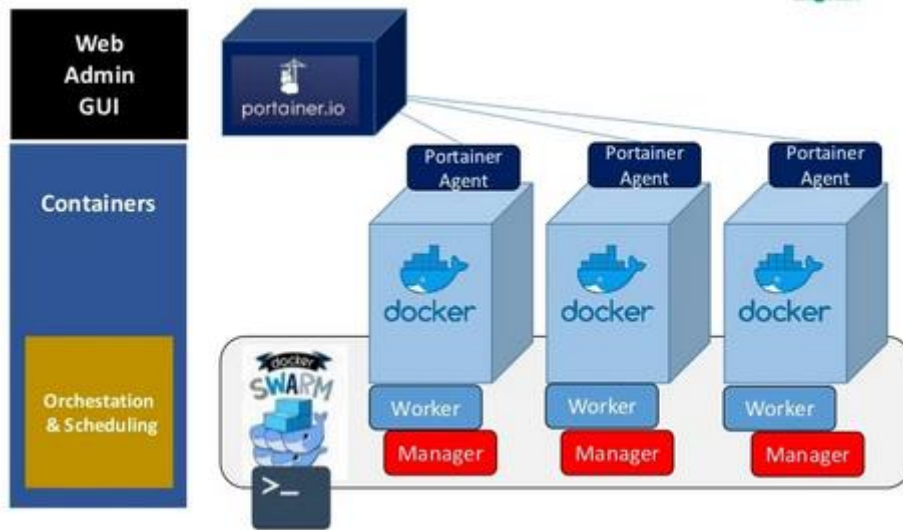


Figura 25.- Arquitectura Portainer.
 Fuente: (Veracierta, 2018)

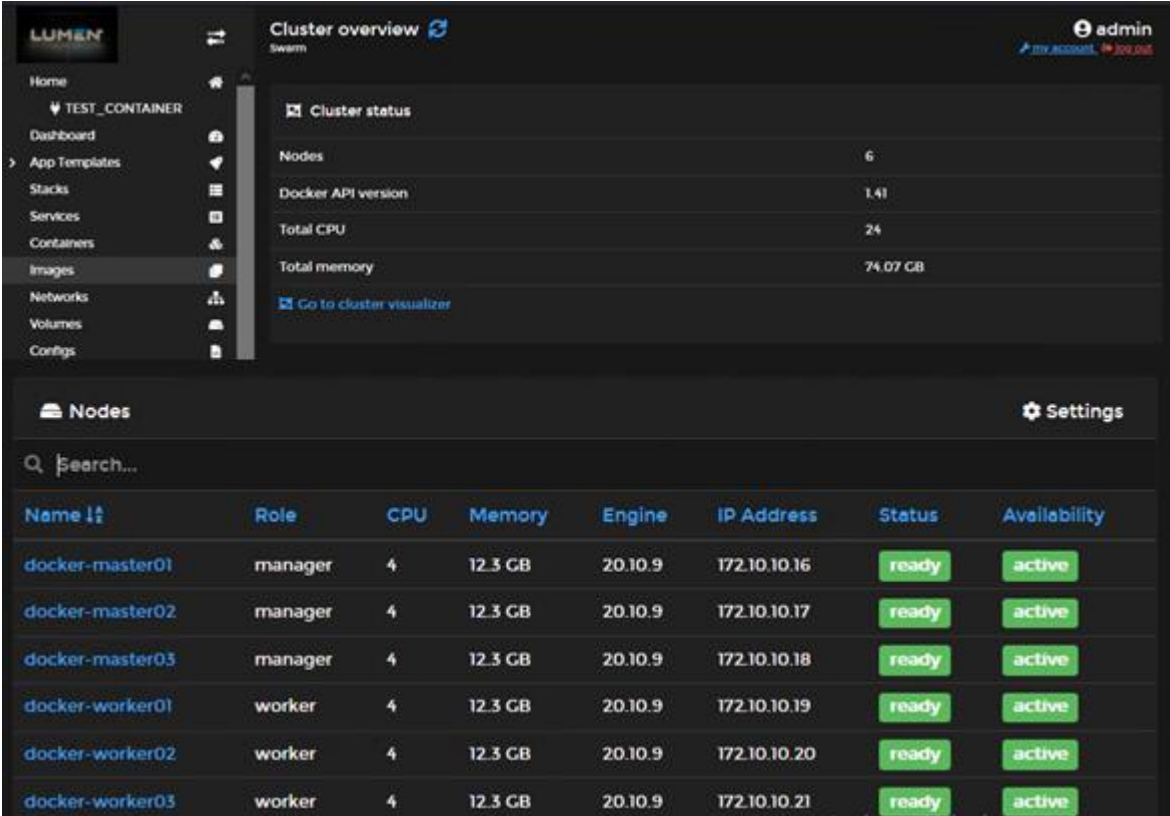
La instalación y configuración de la plataforma se desplegará como un servicio sobre el clúster Docker, la aplicación principal se desplegará en el nodo máster y los agentes se desplegarán en modo global en todos los nodos. Para la instalación se descargará el archivo. YML y se desplegará como un Stack en el clúster mediante: “docker stack deploy -c portainer-agent-stack.yml portainer” en la Figura 26 se evidencia en el nodo máster tanto el servicio y el agente ejecutándose como un Docker.

```

[root@docker-master01 startup]# docker ps | grep portainer
2fe8be63808c   portainer/agent:2.11.0   "/agent"           9 days ago    Up 9 days        0.0.0.0:9001->9001/tcp, :::9001->9001/tcp
portainer_engine_agent.fjzi0caw2kaeksg24337761z3.pw1590ffg7f9zu8ccomyso55b
cc26eec6b0fd   portainer/portainer-ce:2.11.0   "/portainer"       9 days ago    Up 9 days        8000/tcp, 9000/tcp, 9443/tcp
portainer.1.kqnz1st22z0t3vr66dcf36ktt
  
```

Figura 26.- Instalación de Portainer.
 Fuente: Propia.

Mediante la Figura 27 se evidencia como se visualiza la plataforma una vez configurada, se observa que Portainer ya reconoce al clúster Swarm con 6 nodos, 74 Gb disponibles en RAM y 24 CPU, así como su direccionamiento, que es de acuerdo con el diseño propuesto.



Cluster overview
Swarm

Cluster status

- Nodes: 6
- Docker API version: 1.41
- Total CPU: 24
- Total memory: 74.07 GB

[Go to cluster visualizer](#)

Nodes Settings

Search...

Name	Role	CPU	Memory	Engine	IP Address	Status	Availability
docker-master01	manager	4	12.3 GB	20.10.9	172.10.10.16	ready	active
docker-master02	manager	4	12.3 GB	20.10.9	172.10.10.17	ready	active
docker-master03	manager	4	12.3 GB	20.10.9	172.10.10.18	ready	active
docker-worker01	worker	4	12.3 GB	20.10.9	172.10.10.19	ready	active
docker-worker02	worker	4	12.3 GB	20.10.9	172.10.10.20	ready	active
docker-worker03	worker	4	12.3 GB	20.10.9	172.10.10.21	ready	active

Figura 27.- Visualización del Clúster Swarm en Portainer.
Fuente: Propia.

También se tiene una visualización más amigable de los Contenedores que se encuentran ejecutándose en cada nodo como se observa en la Figura 28.

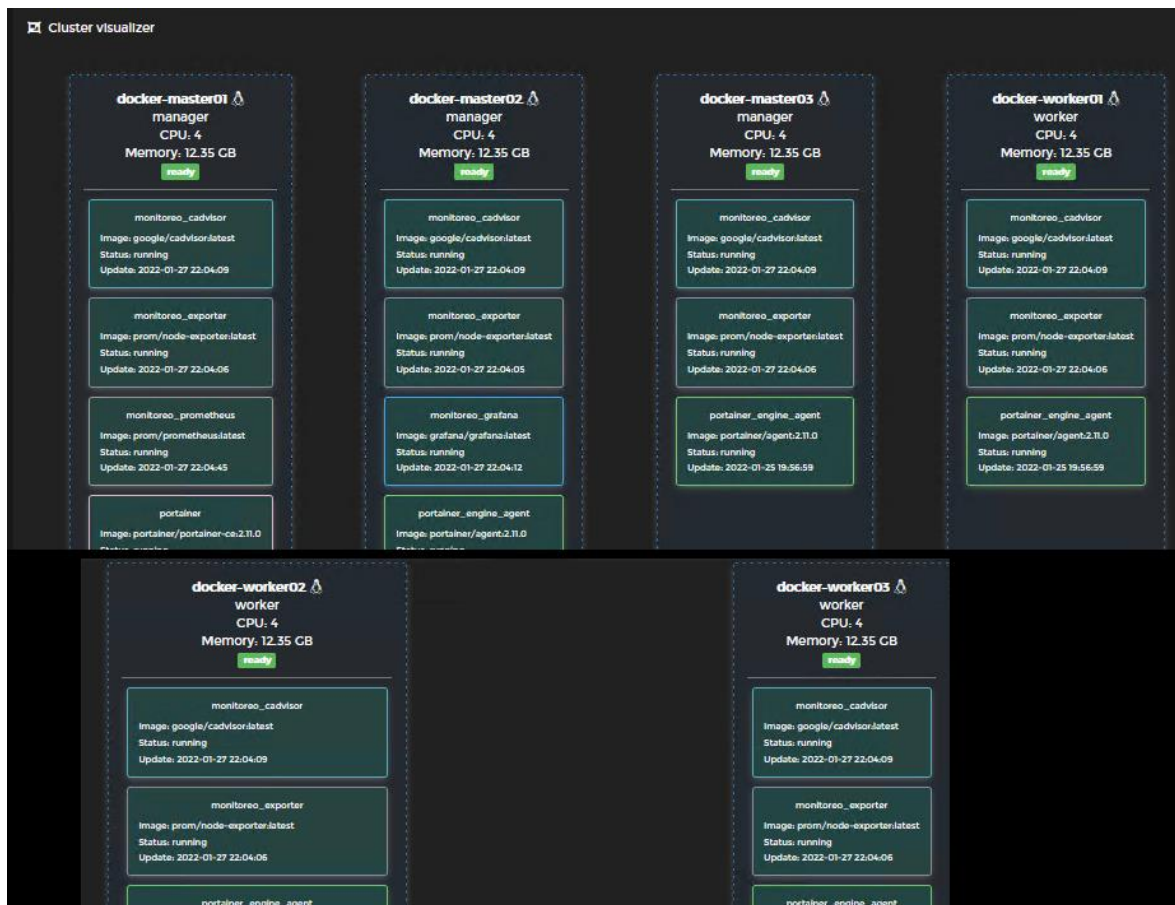


Figura 28.- Vista de clúster Portainer.
Fuente: Propia.

De esta manera se encuentra lista la plataforma para ejecutar cualquier contenedor bajo Dockers. Al usuario final se le entrega un usuario de acceso a la plataforma con los permisos necesarios para la administración. Para el uso de la plataforma por parte del cliente se lo debe realizar mediante una capacitación que constaría como parte del servicio entregado por la empresa (Lumen).

2.5 Diseño y configuración de la capa virtual del IaaS

En la siguiente sección el enfoque será en la preparación y configuración de los sistemas virtuales, así como se asemeja en la siguiente sección a la capa física como el músculo de la infraestructura, en este caso sería el desarrollo del cerebro de la solución se empezara por el Vcenter que es el encargado de centralizar todos los sistemas como host, switch, almacenamiento para poder administrarlos como uno solo en un sistema convergente y dotar de los recursos necesarios para el funcionamiento del prototipo, se verá las

consideraciones y preparación al momento de instalar los hipervisores, así como la configuración y afinamiento del clúster de cómputo que deben responder a un sistema de alta disponibilidad ayudándose de los mecanismos de HA y DRS , en el entorno virtual como en el ambiente físico debe existir una capa de red virtual que sea capaz de unificar el mundo real con el virtual y justamente se revisará la configuración de la capa de red virtual con la configuración de sus enlaces ascendentes como el propósito de las Vlan para su funcionamiento, el sistema no puede funcionar sin almacenamiento al cual acceder por esa razón también se revisará la creación y configuración del clúster de discos, asignación y administración de luns.

2.5.1 Diseño del plano de control de virtualización (Vcenter).

Vcenter es el componente más importante para una infraestructura virtual que ejecuta una solución de VMware, el sistema encargado de hacer converger todos los componentes de un ambiente virtual es justamente Vcenter, en la teoría Vcenter no deja de ser una máquina virtual más y se debe asegurar esos recursos tanto de CPU, memoria y almacenamiento para su implementación, Vcenter se implementara sobre uno del host de cómputo que se tiene disponibles. Se va a implementar Vcenter Server Appliance 7 que es una máquina virtual optimizada que ya viene pre configurada para ejecutar todos los servicios que son necesarios, dependiendo de la carga y cantidad de equipamiento que se dimensione para la solución, el momento de la instalación este demandara los recursos para ejecutar Vcenter y a continuación según la Figura 29 se observa los recursos que requiere Vcenter para su funcionamiento, en nuestro caso al ser un prototipo optaremos por una solución muy pequeña que soportara unos 10 hosts y 100 máquinas virtuales, se necesita 2 vCPUs y 12 GB en memoria RAM.

	Number of vCPUs	Memory
Tiny environment (up to 10 hosts or 100 virtual machines)	2	12 GB
Small environment (up to 100 hosts or 1,000 virtual machines)	4	19 GB
Medium environment (up to 400 hosts or 4,000 virtual machine)	8	28 GB
Large environment (up to 1,000 hosts or 10,000 virtual machines)	16	37 GB
X-Large environment (up to 2,000 hosts or 35,000 virtual machines)	24	56 GB

Figura 29.- Requerimientos de Hardware para Vcenter
Fuente: (VMware, 2021a; VMware vSphere, 2021)

De la misma manera Vcenter requiere de un espacio de almacenamiento para su funcionamiento en la Figura 30 se define el espacio requerido para este propósito, para el caso de este prototipo se requiere como mínimo 579 GB de espacio disponible.

	Default Storage Size	Large Storage Size	X-Large Storage Size
Tiny environment (up to 10 hosts or 100 virtual machines)	579 GB	1992 GB	4279 GB
Small environment (up to 100 hosts or 1,000 virtual machines)	694 GB	2046 GB	4304 GB
Medium environment (up to 400 hosts or 4,000 virtual machine)	908 GB	2140 GB	4468 GB
Large environment (up to 1,000 hosts or 10,000 virtual machines)	1358 GB	1958 GB	4518 GB
X-Large environment (up to 2,000 hosts or 35,000 virtual machines)	2283 GB	2383 GB	4620 GB

Figura 30.- Requerimientos de Almacenamiento para Vcenter.
Fuente: (VMware, 2021a; VMware vSphere, 2021)

Asegurando los requerimientos mínimos podemos empezar con la instalación de Vcenter en nuestra infraestructura, la instalación va a ser lanzada desde una máquina virtual

previamente cargada a uno de los hosts, la versión de Vcenter elegida es 7.0.1.000000. Los recursos asignados a la instalación se visualizan en la Tabla 12 a continuación.

Ítem	Descripción
Nombre de la máquina virtual.	UIO-SG-KBTS-01
Tamaño de la Implementación	Tiny
Almacenamiento	Tiny 579 GB
FQDN	UIO-SG-KBTS-01. SMPDC.GBLX
IP	10.225.0.187
Vlan	Primaria ID 100
NTP	10.225.0.8; 10.225.48.156
DNS	10.225.50.120; 10.225.0.65

Tabla 12.- Recursos Instalación de Vcenter.

Fuente: Propia.

Después de concluir con el proceso de instalación en la Figura 31 finalmente se puede visualizar la instalación de Vcenter.



Figura 31.- Instalación de Vcenter.

Fuente: Propia

2.5.2 Instalación de Hipervisores en Host de cómputo.

Para la instalación de los hipervisores en los hosts de cómputo, en la sección 2.1.1 ya se detalló los requerimientos previos y particularidades de los equipos que son obligatorios para la instalación de Esxi versión 7, para la instalación del Esxi se debe considerar ciertos aspectos en general y sobre todo la CPU y Memoria, ya que es el principal recurso que será consumido por la infraestructura.

Uno de ellos por ejemplo es desconectar o desactivar dispositivos de hardware en el host que no se vaya a utilizar como son: Network interfaces, LPT ports, Floppy drives, USB controllers, COM ports, Storage controller, Optical drives (that is, CD or DVD drives), no solo nos ayuda a optimizar los recursos sino darle seguridad al equipo.

Se va a sacar el mejor provecho del uso de la CPU del host de cómputo, esto gracias a la tecnología Hyper-Threading que permite que con un solo núcleo físico atienda dos procesos lógicos en simultáneo, duplicando la capacidad del Procesador y aumentando el rendimiento del sistema al mantener ocupada la canalización del procesador. El Hyper-Threading por defecto se encuentra activado al momento de instalar el Esxi, en la Figura 32 se puede verificar que se encuentra activado esta característica en los hosts que van a hacer parte de la infraestructura, se puede ver que tenemos dos sockets y cada socket una CPU de 16 núcleos físicos por el Hyper-Threading serían 32 procesadores lógicos, por los dos sockets serían 64 procesadores lógicos en total.

Procesadores	
Modelo	Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz
Velocidad de procesador	2,294 GHz
Sockets de procesador	2
Núcleos de procesador por socket	16
Procesadores lógicos	64
Hiperproceso	Activo

Figura 32.- Activación de Hiperproceso
Fuente: Propia.

Esxi lo hace ver como un todo a los dos procesadores y administran de manera inteligente que la carga se distribuya en todos los núcleos físicos del sistema. Si un proceso lógico no tiene trabajo lo pone en estado libre para ser ocupado por otro proceso. Otro de los recursos que los nodos nos aprovisionan es el acceso a la memoria RAM, se debe asegurar que exista la suficiente memoria que soporte la carga de las máquinas virtuales que se van a ejecutar en el nodo.

Para la instalación de Esxi se debe de proveer la Imagen ISO personalizada, una imagen personalizada es la que cada proveedor de tecnología en conjunto con VMware crean para que la misma tenga los drivers y controladores de todos los componentes del host como

tarjetas de red, tarjeta Hba, tarjetas de video, periféricos, etc. en el caso de la marca que se eligió para esta solución se va a descargar la ISO desde el sitio de Lenovo https://vmware.lenovo.com/content/custom_iso/7.0/7.0u3/, existen varios métodos para la instalación, en esta solución se usará la manera interactiva que consiste en hacer bootear la imagen ISO desde un periférico de CD/DVD-ROM, una vez conectada la ISO al servidor debemos proceder a reiniciar el host, desde la BIOS del sistema se debe escoger la ISO como medio de arranque, entre los parámetros de configuración se tiene que proveer de una contraseña, escoger el medio de almacenamiento donde se va a instalar Esxi. En la Figura 33 se visualiza la instalación del Esxi de uno de los nodos que forman parte de nuestra infraestructura.

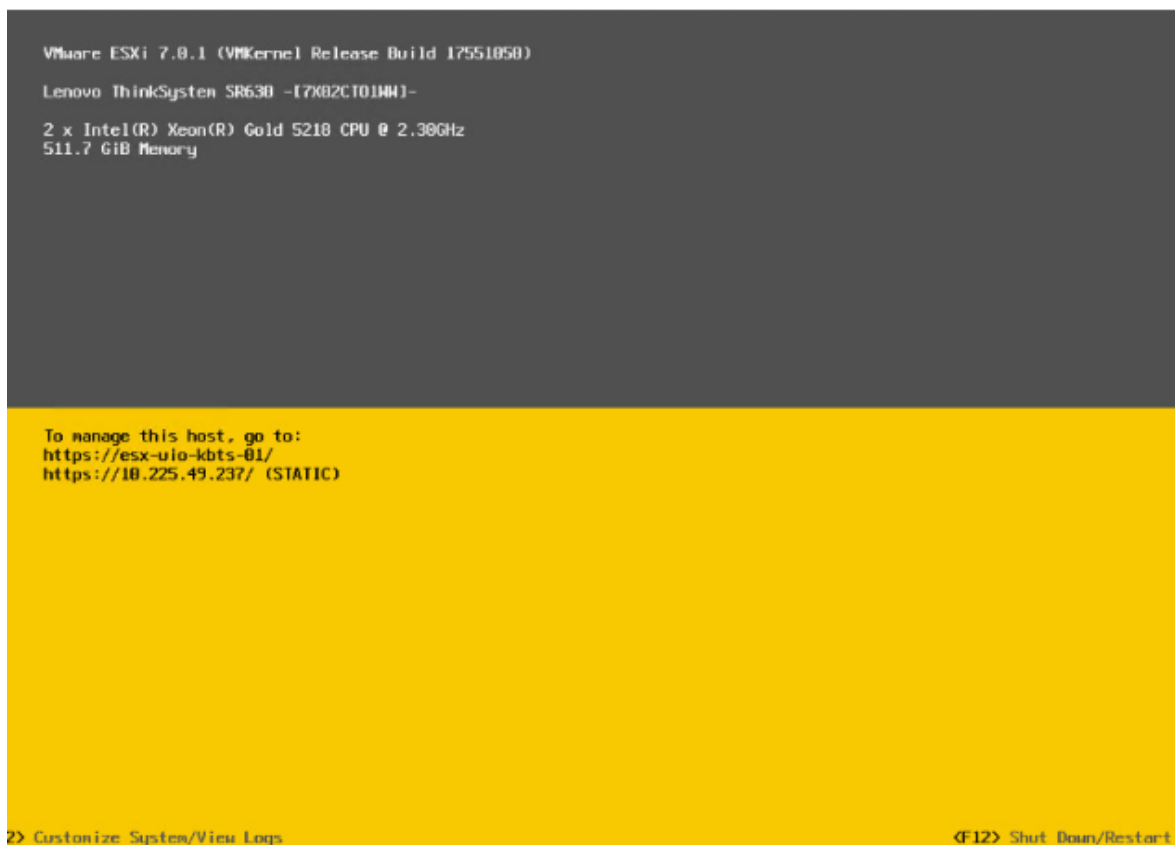


Figura 33.- Configuración Esxi.
Fuente: Propia

Los recursos de red y la configuración de cada uno de los nodos se muestran en la Tabla 13.

Ítem	Host 1	Host 2	Host 3
Network Adapter	vmk 0	vmk 0	vmk 0
Vlan	100	100	100
IP	10.225.49.237/21	10.225.49.238/21	10.225.48.140/21
Hostname	ESX-UIO-KBTS-01. SMPDC.GBLX	ESX-UIO-KBTS-02. SMPDC.GBLX	ESX-UIO-KBTS-03. SMPDC.GBLX
DNS	10.225.50.120; 10.225.0.65	10.225.50.120; 10.225.0.65	10.225.50.120; 10.225.0.65

Tabla 13.- Recursos de Red Esxi.
Fuente: Propia.

2.5.3 Creación y afinamiento de Clúster de cómputo.

Una vez que se tienen instalado y configurado Vcenter, como lo explicamos es el orquestador de toda la solución y a su vez instalados los hipervisores en los hosts que formaran parte del clúster, se va a proceder con la configuración del clúster, dentro de la configuración tenemos entre otros tres parámetros muy importantes que los debemos considerar para el correcto funcionamiento y garantizar la continuidad de la solución, se sabe que en todos los sistemas de un centro de datos se debe contar con alta disponibilidad, por esa razón las técnicas que se verá a continuación hacen posible siempre la disponibilidad de las máquinas virtuales y por ende siempre el accesos a los servicios que estas brinda. Las configuraciones utilizadas serían el DRS (Distributed Resource Scheduler), él HA (High Availability) y el vMotion.

El mecanismo de DRS garantiza que las cargas se encuentren equilibradas en base a un mejor criterio distribuyendo las máquinas virtuales en cada uno del host dependiendo de los recursos que estén disponibles, esto no solo ayuda a que las máquinas virtuales se encuentren en un host que les pueda brindar recursos suficientes para su funcionamiento sino para que no exista una sobresaturación de recursos en el host.

En la versión vsphere 7 el algoritmo de DRS ya no se basa en un punto de equilibrio sino en puntuaciones de DRS de esta forma analiza los recursos de cada miembro del clúster y el aporte que puede contribuir para ejecutar determinada máquina virtual. Justamente por esta razón una de las recomendaciones al momento de crear un clúster es que todos sus miembros sean de igual capacidad de CPU y memoria esto mejora la previsibilidad y la estabilidad del rendimiento.

Algo que se debe considerar para que DRS funcione bien es que todos los Cpus de los hosts sean iguales, se puede utilizar EVC (Enhanced vMotion Compatibility) para facilitar

el funcionamiento de DRS, EVC hace que procesadores de diferente familia o generación los vea DRS como iguales, otro punto a considerar para que funcionen de manera óptima DRS es que todos los hosts que pertenecen al clúster tengan conectividad en común con su almacén de datos y sean accesibles.

En la Tabla 14 se establece los parámetros necesarios para la configuración de DRS en el diseño del prototipo.

Ítem	Descripción
Nombre del Clúster	KBTS-LNV-01
Nivel de Automatización	Totalmente Automatizado
Umbral de migración	Nivel 3, sintonizado para funcionar bien para cargas de trabajo en su mayoría estables.
Distribución de Máquinas Virtuales	Aplique una distribución más uniforme de las máquinas virtuales entre los hosts del clúster para los fines de disponibilidad.

Tabla 14.- Configuración de DRS Clúster.

Fuente: Propia

El mecanismo de alta disponibilidad nos brinda la posibilidad de reducir la indisponibilidad de las aplicaciones, HA está en constante monitoreo de los almacenes de datos, hosts, máquinas virtuales, ante un posible deterioro o falla de uno de los componentes HA reiniciará a las máquinas virtuales en otro host, incluso HA tienen un mecanismo de ser predictivo mediante estadísticas o síntomas ya conocidos y determinar que algo va a suceder con un host, para que el mecanismo funcione todos los miembros del clúster pasan a una elección de un host que sería el principal o maestro y los demás subordinados o esclavos, el host maestro es el que comanda el clúster es el encargado mediante su algoritmo de recopilar información de sus esclavos como estado de salud e informar a su Vcenter y así tomar las decisiones de la protección de las máquinas virtuales, conmutación por error de un host, el ser el host principal no influye sobre su rendimiento o desempeño en producción.

Un host principal cuando no poder recopilar estadísticas del estado de salud de sus esclavos por alguna razón de problemas en la red, el host recurre a su plan B que es monitorear el latido del almacén de datos y determinar si un host esclavo sigue vivo. Para esto por defecto todos los hosts del clúster deben tener al menos 2 almacenes de datos en común. Al momento de activar HA, Vcenter reserva los recursos necesarios para realizar

la conmutación ante la falla de un host. En la Tabla 15 se aprecias las configuraciones aplicadas en la configuración de HA.

Ítem	Descripción
Respuesta de error de host	Reiniciar las máquinas virtuales, cuando se detecta un error en el host.
Errores del host que tolera el clúster	1
Definir la capacidad de conmutación por error.	Porcentaje de recursos del clúster,
Degradación de rendimiento que toleran las VM`s	80%
Nivel de atomización	Automatizado, las máquinas virtuales se migrarán a host en buen estado.
Corrección	Modo Mixto, se evitar usar los hosts parcialmente degradados.

Tabla 15.- Configuración HA clúster.

Fuente: Propia.

Por último se revisará un término vMotion, sin vMotion las dos técnicas revisadas anteriormente no pueden funcionar, DRS o HA el momento que toma la decisión de un movimiento de máquinas virtuales por algún error o presunto error en uno de los hosts el movimiento de las mismas se la hace por VMotion, mientras el Ancho de Banda de la red de VMotion es mayor mejor es el rendimiento de esta característica, se recomienda usar interfaces de 10 Gbps para obtener el máximo rendimiento, en el diseño de la red la Vlan 410 es la asignada para este funcionamiento y efectivamente conectado a interfaces de 10 Gbps, Al ejecutar una tarea de vMotion los hosts Esxi reservan de manera oportunista los recursos de memoria y CPU tanto en el origen como en el destino. En la Figura 34 en el diseño propuesto se evidencia la conexión de dos interfaces a la red de 10 Gbps por donde se ejecutará tareas de vMotion.

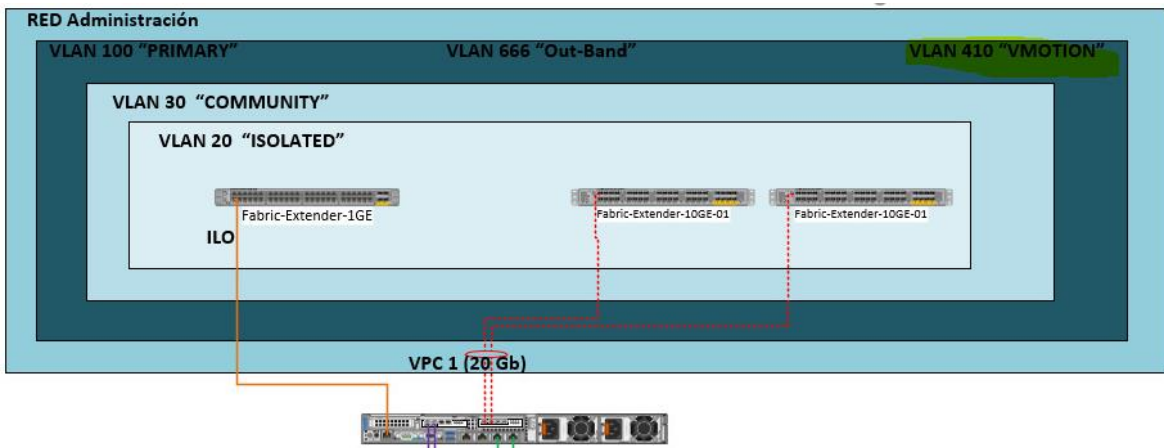


Figura 34.- Red de vMotion
Fuente: Propia.

En la Figura 35 se muestra de una forma más dinámica el proceso de movimiento de las máquinas virtuales en un proceso de vMotion. Vcenter realiza el proceso de reinventar las máquinas en el nuevo host.

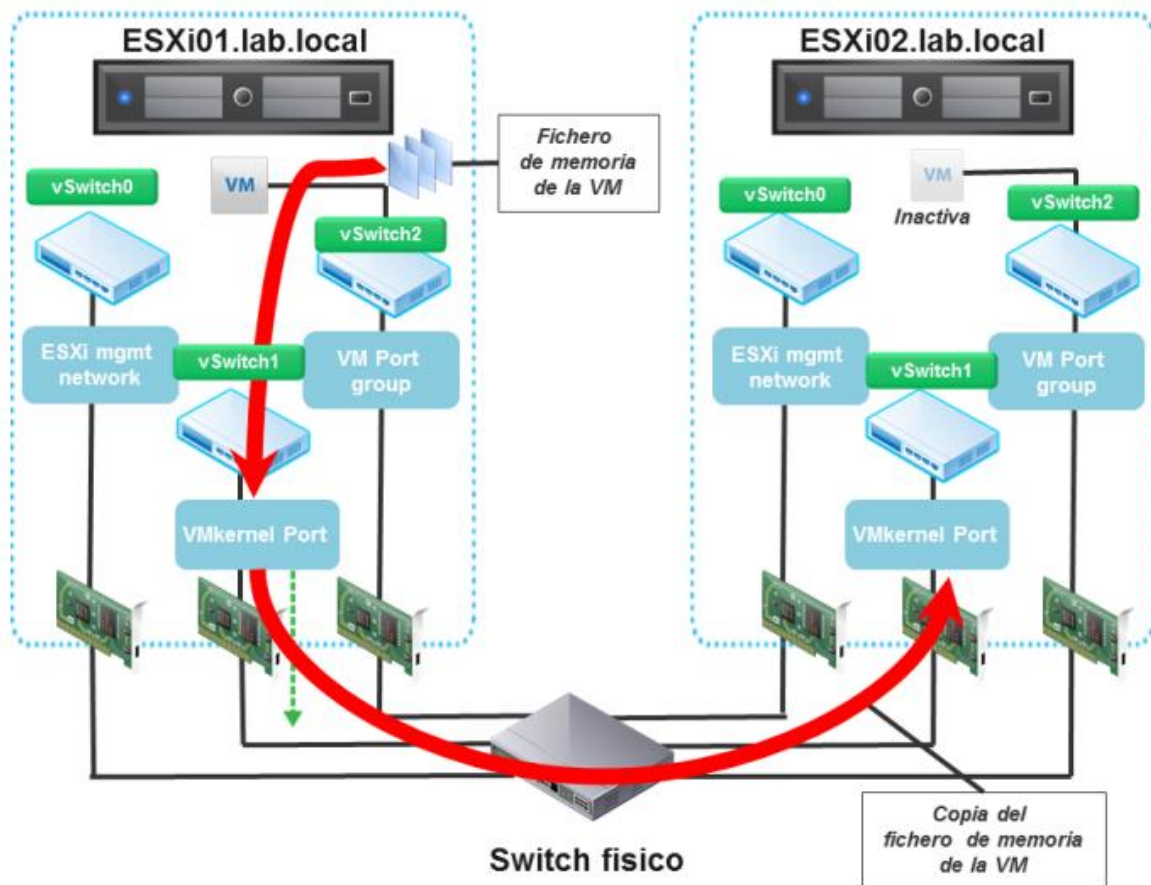


Figura 35.- Tarea de VMotion.
Fuente: (Panic, 2016)

2.5.4 Diseño y Configuración de los componentes en capa de red virtual.

En esta sección se revisará todo lo correspondiente a las redes virtuales bajo la supervisión de VMware, se realizará el diseño de la red de la capa virtual bajo el esquema de switches distribuidos con el uso de LAG (Link Agregation) con alta redundancia.

Para el diseño de la capa de red virtual se debe tomar en cuenta el diseño previo de la capa red física que se explicó en la sección 2.1.2, se debe pensar que de alguna manera se busca que el ambiente virtual haga convergencia con un entorno físico y el mismo funcione en conjunto, VMware no ofrece dos alternativas al momento de configurar las redes, una es con un switch estándar es el componente más básico para crear una red virtual, el switch estándar solo pertenece al host donde se lo crea, es ideal para redes pequeñas y básicas, su utilidad es básicamente hacer un puente para la conectividad de las máquinas virtuales con redes externas en su misma Vlan. La otra alternativa es usar switch distribuidos, su principio de funcionamiento es asociar a todos los hosts en un solo Switch para tener una administración centralizada y supervisar las redes. Los switches distribuidos se los crea en Vcenter y se asocia a los hosts que pertenecen al clúster de esta manera las máquinas virtuales tendrán un esquema de red coherente independientemente en el host que se encuentran ejecutándose.

En la Figura 36 se muestra la arquitectura que recomienda VMware y es adoptada para la solución, la red consta de dos planos lógicos fundamentales el plano de datos y el plano de administración, en el plano de datos es donde se realiza la conmutación de paquetes, empaquetado y desempaquetado de los mismos, es la capa donde comprende todos los protocolos IP, el plano administración es donde se realiza la configuración del funcionamiento del switch distribuido, el plano de Administración pertenece a Vcenter y reside en el mismo con la finalidad de mantener las configuraciones centralizadas, todo lo que se crea, elimina o reconfigura Vcenter las descarga de forma automática a los planos de datos los mismos que pertenecen o se descarga localmente a cada host.

En la Figura 36 se observa varios componentes que conforman la arquitectura, centramos la atención en dos componentes fundamentales, el uno es el grupo de puertos de vínculo superior que se encuentran de color verde y el otro es el grupo de puertos distribuidos de color celeste. Los grupos de puertos de vínculo superior o más conocidos como dvuplink solo los enlaces que unen prácticamente la capa virtual con la capa física, los mismo pueden tener uno o más vínculos, se recomienda de dos a más para tener alta disponibilidad y un equilibrio de carga, los miembros del vínculo superior son asociados a

los puertos físicos del host mediante el switch distribuido de ahí la importancia de que sus características sean de 1 Gbps a más, las configuraciones, directivas, equilibrio conmutación por error son definidas en el switch distribuido y estas se descargan a todos los miembros host, de esta manera todos los vínculos tienen una configuración uniforme.

Por otro lado, se tiene los grupos de puertos distribuidos, es el encargado de proveer la conectividad a las máquinas virtuales y a los puertos de administración de los hosts más conocidos como VMkernel, se puede crear múltiples grupos de puertos y cada uno es asociado una etiqueta para identificarlos dentro del entorno virtual, este grupo de puertos puede ser configurados en modo acceso o troncal y mantener un ID de Vlan así como establecer directivas, conmutación por error, equilibrio de carga, seguridad, catalogación del tráfico de la misma manera como sucede en los grupos de puertos de vínculo superior las configuraciones de los grupos de puertos se lo hace en Vcenter y se descargan automáticamente a cada host que es miembro del switch distribuido.

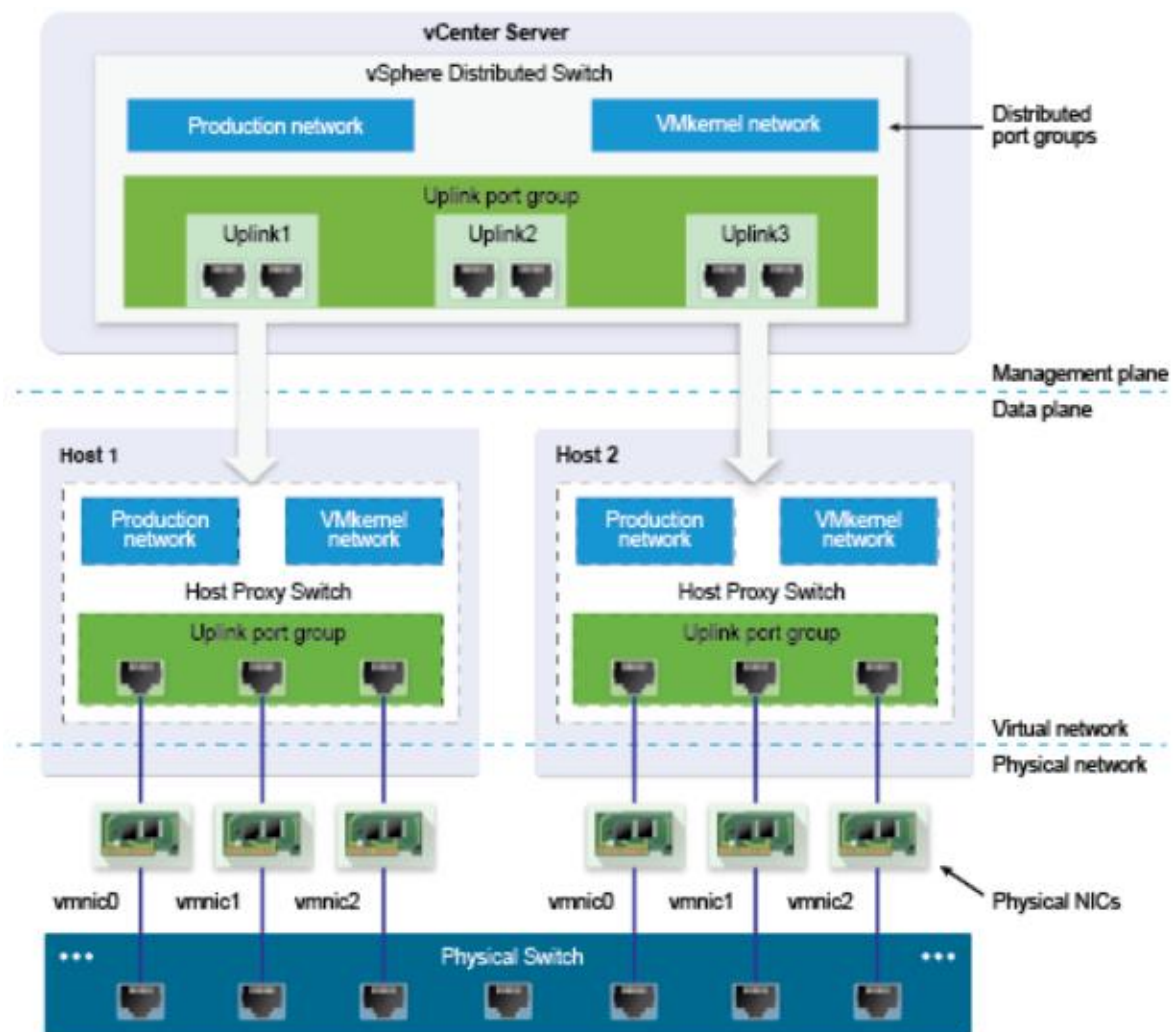


Figura 36.- Arquitectura vDS
Fuente:(VMware vSphere, 2019)

Al principio de esta sección se manifestó que la idea de usar redes virtuales es que interactúen con las redes físicas desarrolladas en la sección 2.1.2 donde al seguir las recomendaciones se hace el uso de LAG o vPC que no es más que un port-channel conectado de una manera que se garantice la alta disponibilidad que se exige en los centros de datos, de esta manera se tiene la necesidad de implementar LAG en los switches distribuidos de la solución, el protocolo que habla LAG es LACP (Link Aggregation Control Protocol) o 802.3ad en modo activo, en este caso los grupos de vínculo superior formaran parte de un LAG que físicamente están conectados los conmutadores del centro de Datos, de esta forma evitar bucles de capa dos y combinar la velocidad de las interfaces y verlos como una sola en la Figura 37 se puede apreciar la arquitectura de la solución.

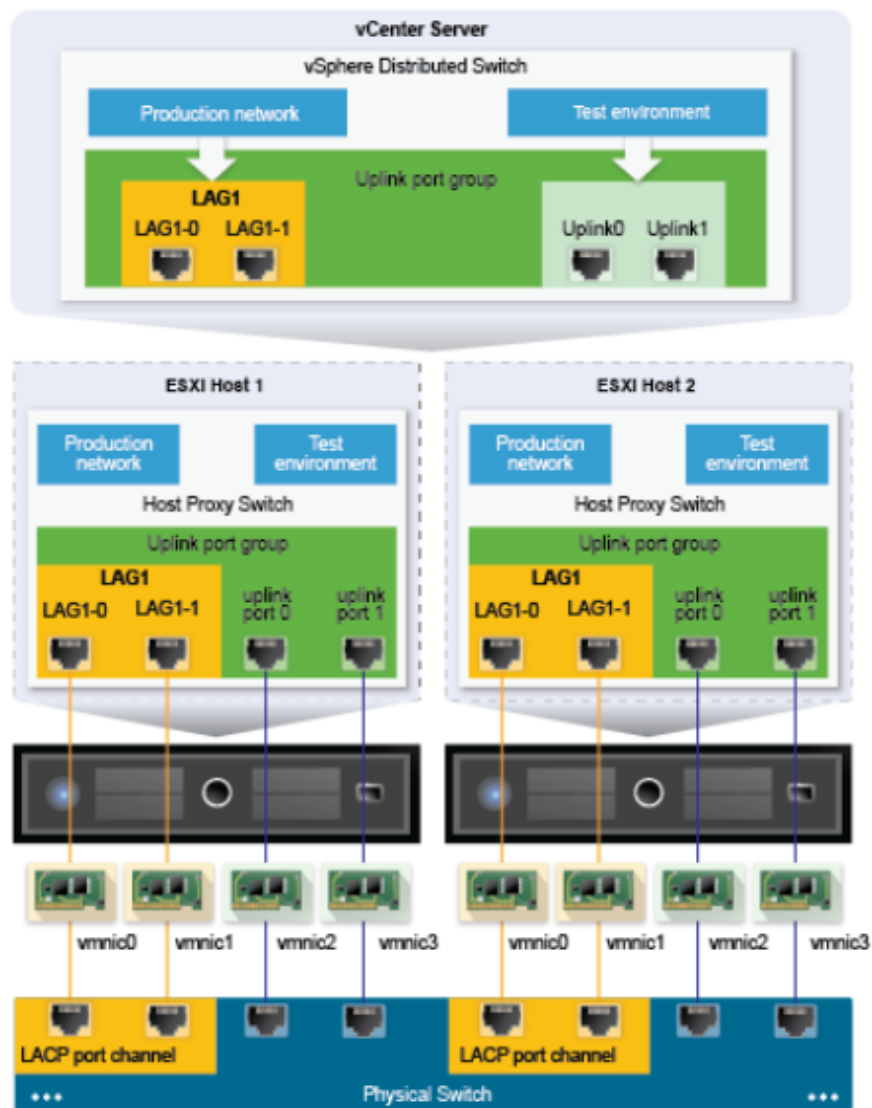


Figura 37.- Arquitectura vDS con LAG
Fuente: (VMware vSphere, 2019)

Así de esta manera se puede establecer el diagrama topológico de la solución, la Figura 38 se muestra como es la conexión de los nodos con la red de datos de esta manera se garantiza la alta disponibilidad de los recursos.

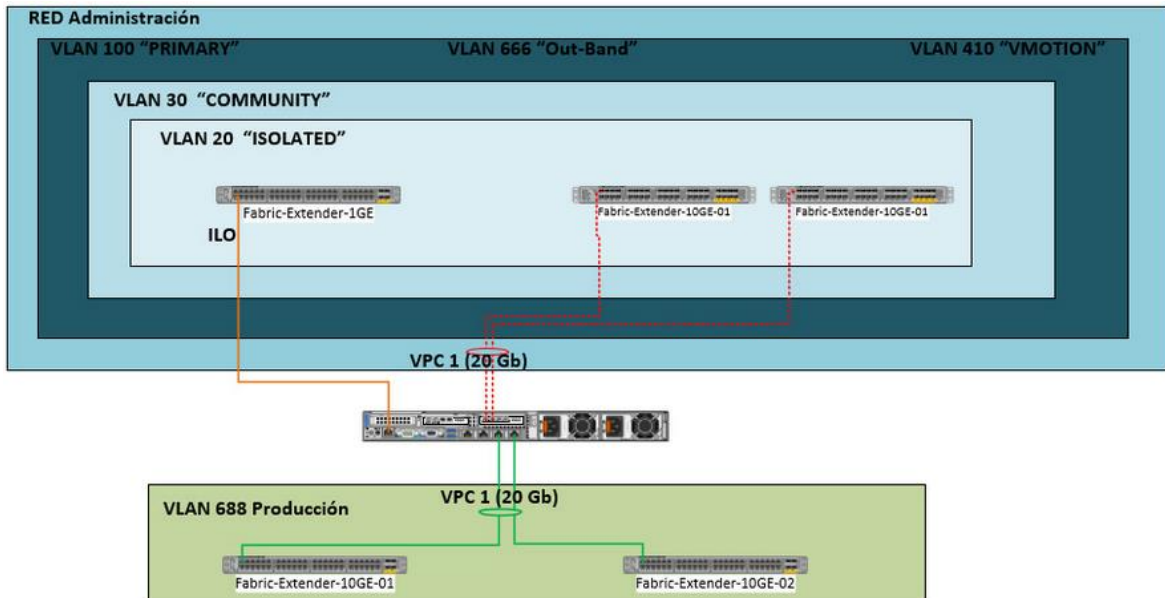


Figura 38.- Diagrama de Red de la solución.
Fuente: Propia.

En la Figura 39 se presenta el esquema virtual configurado, en la parte superior se muestra el switch distribuido que corresponde a la red de producción y en la parte inferior el switch de la red de Administración, vMotion, administración fuera de banda, en ambos caso se conforman un LAG con dos vínculos de nivel superior que se visualiza en la parte de la derecha, en la parte izquierda se muestran los grupos de puertos distribuidos y la asignación a cada máquina virtual así como los VMkernel para la administración de los hosts, de esta manera se verifica que ante la falla de un vínculo el otro toma el mando.

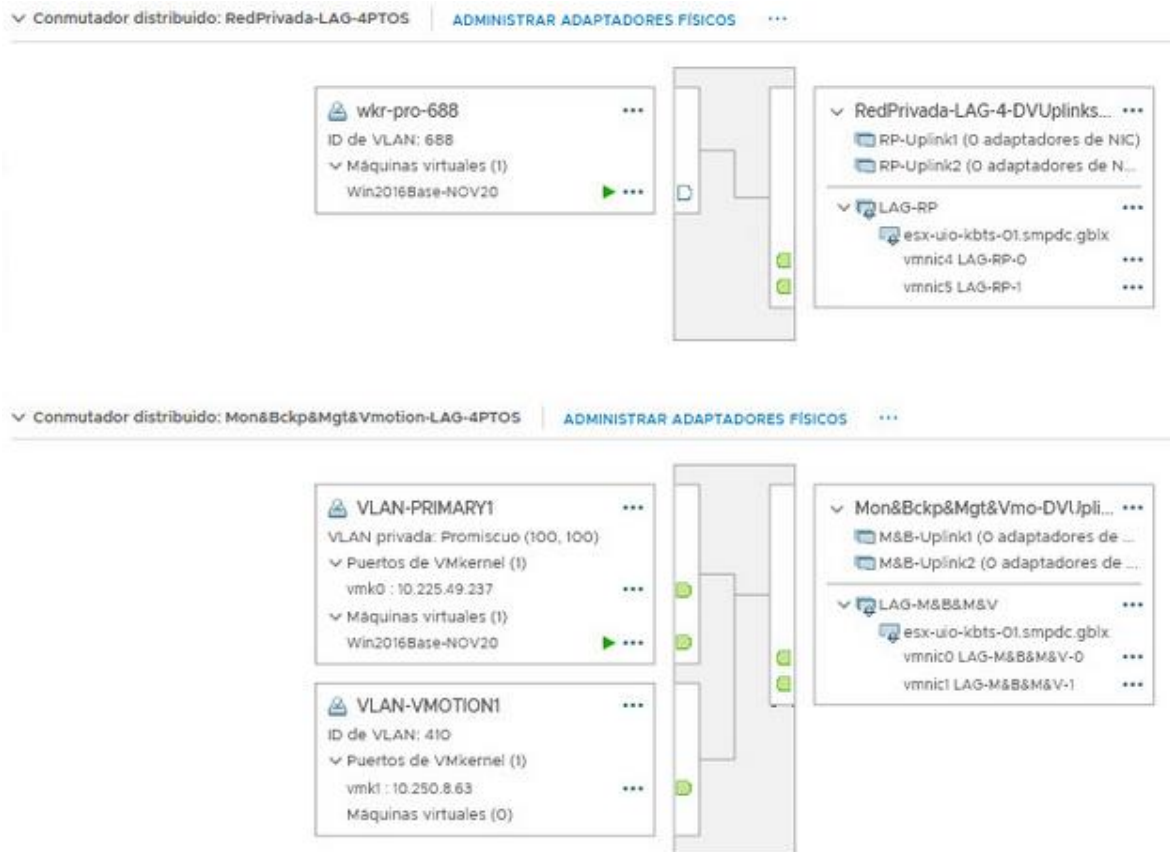


Figura 39.- Diagrama Virtual.
Fuente: Propia.

De esta manera en la Tabla 16 y Tabla 17 se muestra los parámetros configurados al crear el vDS para la red de Administración y la red de Producción, como se explicó en la sección 2.1.2 por temas de seguridad se maneja un esquema de Vlans privadas, en el switch distribuido también se debe realizar estas configuraciones.

Ítem	Descripción
Nombre Distributed Switch	Mon&Bcpg&Mgt&Vmotion.
MTU (Bytes)	1500
Protocolo de detección Tipo	Cisco Discovery Protocolo
Nombre Vinculo Superior	M&B-Uplink1 y M&B-Uplink2
Nombre LAG	LAG-M&B&M&V.
Cantidad de puertos	2
Modo de operación	Activo
Vlan Primaria	100
Vlan Isolated	30
Vlan Comunidad	20
Grupo de puertos	Vlan-Primaria
ID-vlan	Promiscuo (100, 100)
Grupo de puertos	Vlan-Aislada
ID-vlan	Aislada (100, 30)
Equilibrio de carga	Enrutar según la carga de la NIC física
Grupo de puertos	Vlan-Vmotion
ID-vlan	410

Tabla 16.- Parámetros vDS Administración
Fuente: Propia

Ítem	Descripción
Nombre Distributed Switch	RedPrivada
MTU (Bytes)	1500
Protocolo de detección Tipo	Cisco Discovery Protocolo
Nombre Vinculo Superior	RP-Uplink1 y RP-Uplink2
Nombre LAG	LAG-RP
Cantidad de puertos	2
Modo de operación	Activo
Grupo de puertos	Wkr-pro-688
ID-vlan	688
Equilibrio de carga	Enrutar según la carga de la NIC física

Tabla 17.- Parámetros vDS Producción.
Fuente: Propia.

2.5.5 Configuración de la capa de Almacenamiento.

En esta sección se definirá la forma de configuración del almacenamiento para los datos y sus características, al igual que en las otras secciones en lo que se refiere al almacenamiento también tiene sus buenas prácticas y recomendaciones de parte de VMware.

Las máquinas virtuales se conforman de múltiples archivos de configuración como los Datos que se encuentra almacenados en los discos duros que tienen un formato como VMDK, archivos de configuración como vmx, nvram, etc. que definen el hardware de la máquina virtual, estos archivos necesitan en donde almacenarse de una manera segura, confiable e íntegra, y debe estar siempre disponible es decir contar con alta disponibilidad.

A este espacio de almacenamiento se lo conoce como LUN (logical unit number) esta debe ser común por todos los hosts del clúster, la LUN es reconocida por los protocolos SCSI y estas son en un pedazo de espacio que se crean a partir de un conjunto de discos comúnmente en arreglos conocidos como RAID, los discos que conforman el Raid pueden ser mecánicos que se caracterizan por ser de revolución operando a 10 Krpm o 15 Krpm, o discos de estado sólido, cada uno da sus ventajas y desventajas siendo más eficientes los de estado sólido o SSD. Al momento de presentar una LUN al clúster Vcenter lo da un formato, el sistema de archivos VMFS6 (VMware Virtual Machine File System). En la Figura 40 se puede apreciar la manera que los Hosts Esxi miran en este caso a dos luns en común con formato VMFS donde se almacena toda la información de las máquinas virtuales.



Figura 40.- Data Storage Esxi
Fuente:(Gonzalez, 2015)

Como se trabajó en secciones anteriores el acceso a el espacio de almacenamiento se lo hace por SAN que trabaja bajo el protocolo de Fibra Canal a una velocidad de 16 Gbps con doble HBA. En la Tabla 18 se muestra la configuración y características de las luns presentadas al clúster.

Ítem	Descripción
Modelo Storage	Hitachi VSP G400
Nombre de la LUN 1	TanzuKbts
Tamaño	1 Tb
Raid	Raid6(6D+2P)
LUN ID	2
Nombre de la LUN 2	TanzuKbts1
Tamaño	1Tb
Algoritmo de acceso	Round-Robin

Tabla 18.- Configuración Storage.
Fuente: Propia.

De la misma manera como se crea un clúster con múltiples nodos de cómputo, se puede crear clúster de almacén de datos, de esta manera garantizamos siempre el acceso a un espacio de almacenamiento, ante el fallo de una LUN todo migra a una LUN disponible mediante la técnica de storage vMotion. Cuando se ejecuta una operación de Storage vMotion Vcenter lee los datos de los discos en el origen y va escribiendo en el almacén de destino, al mismo tiempo la máquina virtual sigue con operaciones de escritura y lectura de manera normal sin interrupciones mientras se transporta los datos del origen con problemas a un espacio con mejor desempeño. En la Figura 41 se muestra en proceso de Storage vMotion.

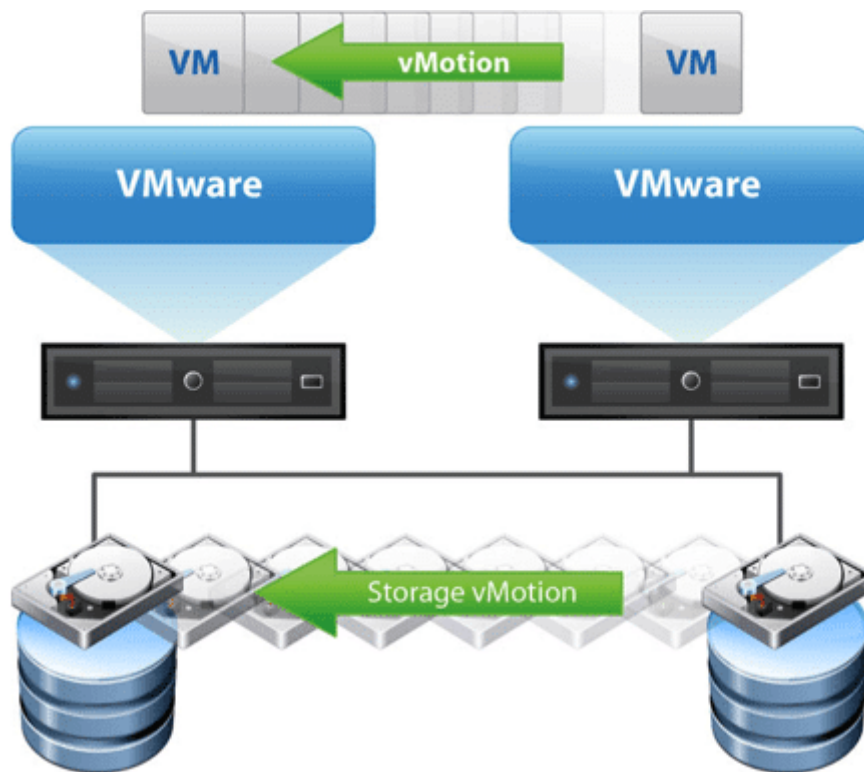


Figura 41.- Storage Vmotion.
Fuente: (Gonzalez, 2015)

En la Tabla 19 se muestra los parámetros de configuración que se emplearon para la configuración del clúster de Storage, de esta manera garantizamos la alta disponibilidad en almacenamiento.

Ítem	Descripción
Nombre del Clúster	KBTS-CLUSTER-STG
Nivel de Automatización	Totalmente automatizado, los archivos se migrarán automáticamente para automatizar el uso de recursos.
Umbral de latencia de E/S	15 ms, si la latencia de E/S está por debajo de 15ms la Lun se encuentra degradada y no está disponible para soportar nuevas máquinas virtuales.
Umbral de espacio.	80%, Si el Espacio sube de ocupación de 80% empieza el equilibrio hacia otros storage con más espacio disponible.
Cantidad de Luns del clúster.	2
Tamaño total	2 Tb

Tabla 19.- Parámetros Clúster de Storage.
Fuente: Propia.

En la Figura 42, se muestra la configuración final implementada en la solución del prototipo propuesto, así se finaliza la configuración de lo que corresponde a el espacio de almacenamiento.

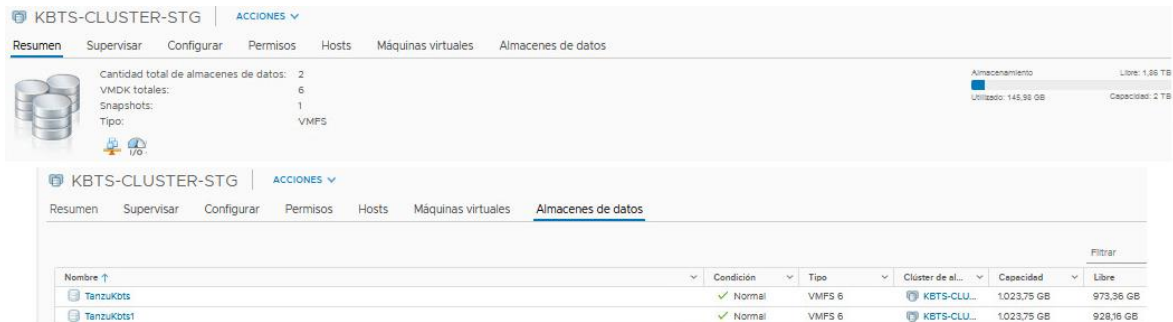


Figura 42.- Clúster de Storage.
Fuente: Propia.

En toda la sección anterior se realizó la implementación de todas las capas tanto a nivel virtual como a nivel físico, la misma al converger se transforman en una plataforma como servicio más conocido como PaaS, al momento la infraestructura puede recibir cualquier tipo de máquina virtual con sus respectivos Sistemas Operativos soportado por VMware versión 7. De aquí en adelante se trabajará en la plataforma como tal del prototipo, el concepto es entregar PaaS orientada a contenedores CaaS.

2.6 Diseño y configuración de la capa física del IaaS

La presente sección se centra en el diseño y configuración de la capa física, que es el músculo del prototipo propuesto, se definirá la configuración y tipo de nodos que se utilizará apegándose a las recomendaciones de las diferentes marcas y las mejores prácticas de acuerdo a las experiencias en previas implementaciones en los diferentes centros de datos del mundo, se realizará la configuración de la capa de red que es la que se encargara la comunicación de todos los sistemas y nos brindara el acceso a los mismos de una manera eficiente y rápida. También se verá la configuración y despliegue de la red SAN de alta velocidad la misma que nos permite el acceso a los sistemas de almacenamiento manejando protocolos de Fibra Canal emulando el acceso a los mismos como si fueran discos locales que es el objetivo final de la tecnología SAN.

2.6.1 Preparación y Configuración de Nodos de cómputo.

La elección de los nodos de cómputo es fundamental el momento de la implementación de un ambiente virtual, los nodos de cómputo son el músculo de la infraestructura, ya que son donde residirán nuestros ambientes virtuales, de la misma forma los nodos de cómputo son los que accederán a los recursos tanto de red LAN, SAN, Storage y todos los componentes que conforman la infraestructura que los iremos estudiando a detalle en el documento.

Algo importante en la elección de los nodos de cómputo se debe considerar la compatibilidad del hardware con la versión de hipervisor que se va a instalar, en este caso Lenovo nos proporcionara uno de sus modelos de nodos de cómputo para la implementación del prototipo, el modelo SR630-ThinkSystem se muestra en la Tabla 20.

Ítem	Descripción.
Formato/altura	Servidor de 1 Unida de Rack
Procesador	Dos procesadores Intel® Xeon® Platinum de segunda generación.
Slot de Memoria	24 ranuras DIMM TruDDR4 de 128 GB a velocidad de reloj 2666 MHz / 2933 MHz.
Ranuras de expansión	4 ranuras 3.0 PCI express.
Bahías para discos	12 bahías para discos de 3,5" y 2.5", 2 unidades M.2 duplicadas para arranque.
Soporte de HBA/RAID	Soporte hasta 16 puertos HBA. RAID con caché flash.
Seguridad y disponibilidad	LED`s de diagnóstico Light Path; ventiladores y PSU`s redundantes; PFA; diagnóstico de acceso frontal con puerto USB exclusivo. TPM 1.2/2.0; funcionamiento continuo a 45°C.
Interfaces de red.	1 puerto 1GbE para gestión fuera de banda, 2-4 puertos de 10GbE con Base-T o SFP+, 2-4 puertos de 1GbE.
Consumo	2 fuentes redundantes: 80 PLUS Titanium de 750 W CA u Platinum de 550 W/750 W/1100 W CA.
Gestión del sistema.	Integración con la infraestructura agrupada XClarity Administrator, Administración integrada con XClarity Controller
SO soportados	VMware vSphere, Microsoft, Red Hat, SUSE.
Garantía limitada.	3 años asistencia a domicilio al siguiente día laborable 9x5.

Tabla 20.- Especificaciones ThinkSystem SR630

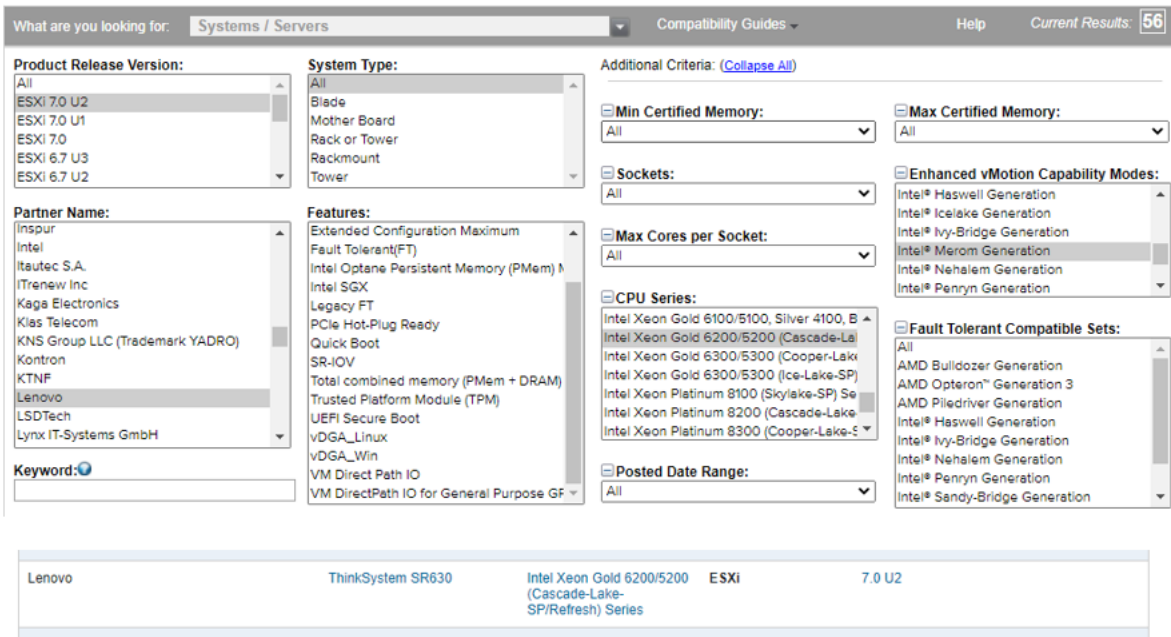
Fuente: (Lenovo., 2021)

De todas las características disponibles que nos ofrece la marca en nuestro caso se optó por 3 nodos con las siguientes características como se describe en la Tabla 21.

Ítem	Descripción.
Procesador	2 procesadores Intel® Xeon® Gold 5218 @ 2.30GHz, 16 Núcleos, 32 Subprocesos.
Memoria	512 Gb DIMM TruDDR4 2933 MHz.
HBA	QLogic 16Gb FC Dual-port HBA Adaptador
Disco	ThinkSystem M.2 SATA, SSD, 120 GB, RAID 1
Seguridad y disponibilidad	LED`s de diagnóstico Light Path; ventiladores y PSU`s redundantes; PFA; diagnóstico de acceso frontal con puerto USB exclusivo. TPM 1.2/2.0; funcionamiento continuo a 45°C.
Interfaces de red.	Intel LOM X722 10GbE de 4 SFP+; Intel X710 2x10GbE SFP+
Fuentes	2 hot-swap/redundantes: 80 PLUS Platinum de 750 W
Gestión del sistema.	Integración con la infraestructura agrupada XClarity Administrator, Administración integrada con XClarity Controller.
SO	VMware vSphere

Tabla 21.- Características Nodo de Computo
Fuente: Propia.

La versión del hipervisor que utilizara para la implementación del prototipo sería la última versión disponible Esxi, lo que nos lleva a validar si nuestros nodos son compatibles, VMware nos ofrece una alternativa para validarlo en la web <https://www.vmware.com/resources/compatibility/search.php> como se observa en la Figura 43 se debe ingresar los datos disponibles del nodo para evaluar su compatibilidad con la última versión de Esxi como resultado nos da que si es compatible nuestro nodo y la versión Esxi a instalar sería la 7.0.



The screenshot shows the VMware Compatibility Guides search interface. The search criteria are as follows:

- Product Release Version:** ESXi 7.0 U2
- System Type:** Blade
- Partner Name:** Lenovo
- Additional Criteria:**
 - Min Certified Memory: All
 - Max Certified Memory: All
 - Sockets: All
 - Max Cores per Socket: All
 - CPU Series: Intel Xeon Gold 6200/5200 (Cascade-Lake-SP/Refresh) Series
 - Posted Date Range: All
- Enhanced vMotion Capability Modes:** Intel® Nehalem Generation
- Fault Tolerant Compatible Sets:** Intel® Sandy-Bridge Generation

The search results table shows the following entry:

Lenovo	ThinkSystem SR630	Intel Xeon Gold 6200/5200 (Cascade-Lake-SP/Refresh) Series	ESXi	7.0 U2

Figura 43.- Análisis de Compatibilidad VMware.
Fuente: (VMware, 2021b)

Para la instalación del hipervisor, VMware nos exige requerimientos mínimos que deben cumplir los nodos de cómputo que se detallan en la Tabla 22, haciendo un match con las características de los nodos verificamos que la elección de los nodos de cómputo es acertada.

Ítem	Descripción.
Procesador	Al menos dos núcleos de CPU, gama de procesadores x86 de 64 bits de varios núcleos.
Memoria	Tener como mínimo 4 GB de RAM, es recomendable asignar 8 GB de RAM para un máximo rendimiento del Esxi.
VT-x	QLogic 16Gb habilitar la compatibilidad para virtualización Dual-port HBA Adaptador.
Disco	Disco de arranque de 32 GB para tipos de dispositivos, como HDD, SSD o NVMe.
Seguridad y disponibilidad	LED's de diagnóstico Light Path; unidades hot-swap, LED's de diagnóstico Light Path; ventiladores y PSU's redundantes; PFA; diagnóstico de acceso frontal con puerto USB exclusivo. TPM 1.2/2.0; funcionamiento continuo a 45°C.
Interfaz de red.	Colocar la red de administración y la red de producción de las máquinas virtuales en tarjetas de red distintas.

Tabla 22.-Requisitos Mínimos.
Fuente:(VMware, 2021b)

Antes de iniciar la instalación del hipervisor en los nodos de cómputo, es importante configurar la BIOS para sacarle el mejor rendimiento al hardware, es posible que la configuración predeterminada del BIOS de hardware en los servidores no siempre sea la mejor opción para un rendimiento óptimo. Esta sección enumera algunas de las configuraciones del BIOS que se debe verificar, particularmente cuando configure por primera vez un nuevo servidor. (VMware, 2021a)

- Se debe validar que se debe estar ejecutando la última versión del BIOS disponible para el sistema.
- BIOS esté configurada para activar todos los zócalos de procesador ocupados y para activar todos los núcleos en cada zócalo
- Activar "Turbo Boost" en el BIOS si sus procesadores lo admiten.
- Activar Hyper-Threading esté activado en el BIOS.
- Activar NUMA (Non-Uniform Memory Access)
- Activar funciones de virtualización asistida por hardware VT-x de Intel.
- Desactive desde el BIOS cualquier dispositivo que no vaya a utilizar. Esto puede incluir, por ejemplo, un puerto de red, USB o serie innecesario.

En función a los requerimientos solicitados, Lenovo nos brinda perfiles de configuración para sacarle el máximo provecho a los nodos de cómputo, y establece los parámetros que los debemos configurar como se considera en la Figura 44.

```
ThinkSystem Processors.CStates=Disable  
Processors.C1EnhancedMode=Disable  
Processors.EnergyEfficientTurbo=Disable  
Processors.MONITORMWAIT=Enable  
Power.PowerPerformanceBias=Platform Controlled  
Power.PlatformControlledType=Maximum Performance  
DevicesandIOPorts.PCI64BitResourceAllocation=Auto or Disable (*)  
DevicesandIOPorts.MMConfigBase=3GB
```

Figura 44.-Configuración UEFI Lenovo
Fuente:(Lenovo, 2021)

De esta manera y apegándose a las recomendaciones y buenas prácticas se tiene configurados los nodos a nivel de hardware, se debe garantizar la alta disponibilidad a nivel de las comunicaciones, justamente por esa razón se tiene múltiples puertos tanto como

para LAN y SAN, a continuación, se considera el diagrama de las conexiones de los nodos de cómputo tanto con la LAN y la SAN véase Figura 45.

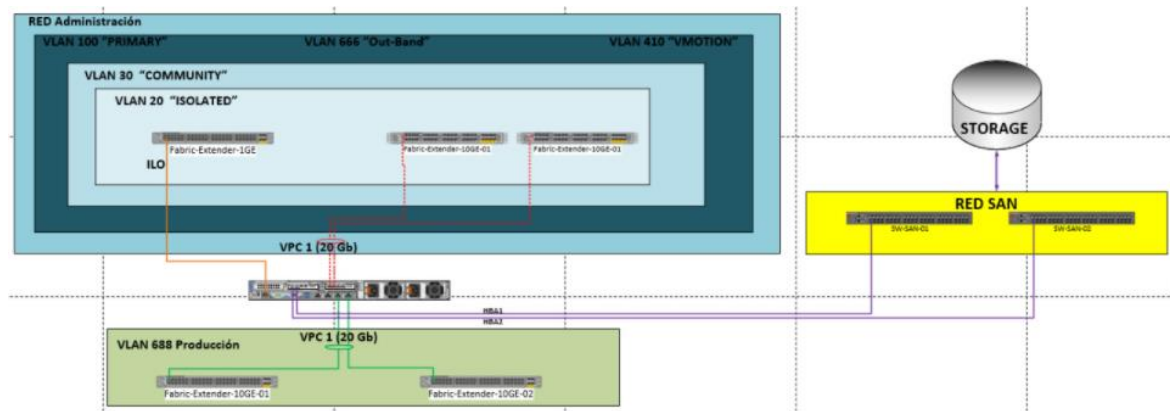


Figura 45.- Diagrama de Conexión de Host.
Fuente: Propia.

2.6.2 Diseño y Configuración de la Red de Datos.

La familia de productos Cisco Nexus se ha vuelto extremadamente popular en centros de datos pequeños y grandes gracias a su capacidad para unificar los servicios de almacenamiento, datos y redes. Gracias a Cisco Fabric Interconnect, no solo son compatibles con todos estos servicios, sino que también proporcionan una plataforma programable sólida como una roca que es totalmente compatible con cualquier entorno virtualizado. La familia Cisco Nexus incluye una generosa cantidad de diferentes modelos Nexus para satisfacer las demandas de cualquier entorno de centro de datos. (BOULOUKO, 2020)

En relación con el criterio descrito en la parte superior. La capa física de la red LAN es fundamental para el funcionamiento de la infraestructura y será el que nos proporcione las comunicaciones entre los nodos, la tecnología ideal para el Centro de Datos es Cisco en su gama Nexus que son diseñados para redes de alta velocidad con alta disponibilidad. En esta sección se verá los equipos elegidos y sus características, así como el criterio para la planificación y configuración siendo nuestra guía de consulta las mejores prácticas recomendadas por el fabricante Cisco.

Para el diseño se debe cumplir con los requisitos que la capa virtual exige en sus recomendaciones, tanto para mantener la seguridad y el funcionamiento de esta, en ese caso tenemos la necesidad de crear una red en múltiples Vlan (virtual LAN) y con jerarquía, se creará una Vlan de administración la misma que se usará para el acceso y

administración de las múltiples máquinas virtuales, Vcenter, asimismo la administración en los nodos de cómputo.

En esta Vlan al contener tráfico tanto de las máquinas virtuales como de los nodos de cómputo, se tiene la necesidad de ofrecer seguridad en la red, con el objetivo que este tráfico no sea fácilmente capturado por las máquinas virtuales ni por los nodos de cómputo, en ese caso emplearemos la técnica conocida como “Private Vlan”.

Las Private Vlan crea Vlan en subdominios dividiendo el dominio de broadcast de una Vlan, gracias esto se puede aislar el tráfico y por ende los puertos de un switch. Private Vlan se compone de una Vlan llamada Primaria y varias Vlans llamadas secundarias como se muestra en la Figura 46, todas las Vlan en un dominio de Private Vlan comparten la misma Vlan primaria, las Vlans secundarias pueden ser configuradas como Vlans aisladas y Vlans de comunidad(CISCO SYSTEM, 2019).

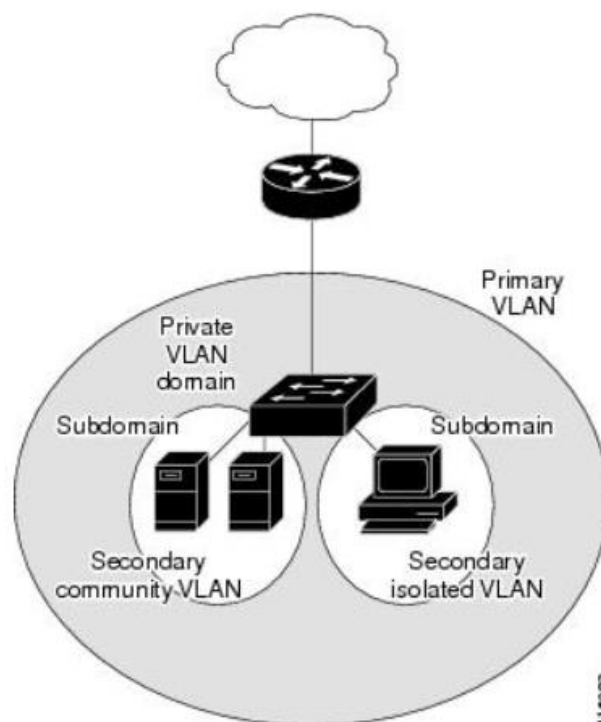


Figura 46.- Private Vlans
Fuente: (CISCO SYSTEM, 2019)

A continuación, el concepto de Vlans Primarias y secundarias de acuerdo con su configuración:

- Vlan Primaria: tiene la capacidad de transportar el tráfico a los puertos de host, desde puertos promiscuos, aislados o comunitarios hacia otros puertos promiscuos.

- Vlan Aislada: pertenece a una Vlan secundaria solo tiene la capacidad de transportar tráfico Unidireccional ascendente desde los hosts hacia puertos promiscuos, el tráfico aislado se mantiene aislado en el host, es decir los puertos aislados solo se comunican con ellos mismos y con los puertos promiscuos.
- Vlan Comunitaria: pertenece a una Vlan secundaria tiene la capacidad de transportar tráfico tanto a los puertos promiscuos como a los de su misma comunidad. Puede existir varias Vlans de comunidad en el un dominio de Vlan privada, no se permite el tráfico entre comunidades ni puertos aislados.

A continuación, en la Figura 47 se diagrama el funcionamiento de las Vlans privadas y su comunicación en su dominio (CISCO SYSTEM, 2019):

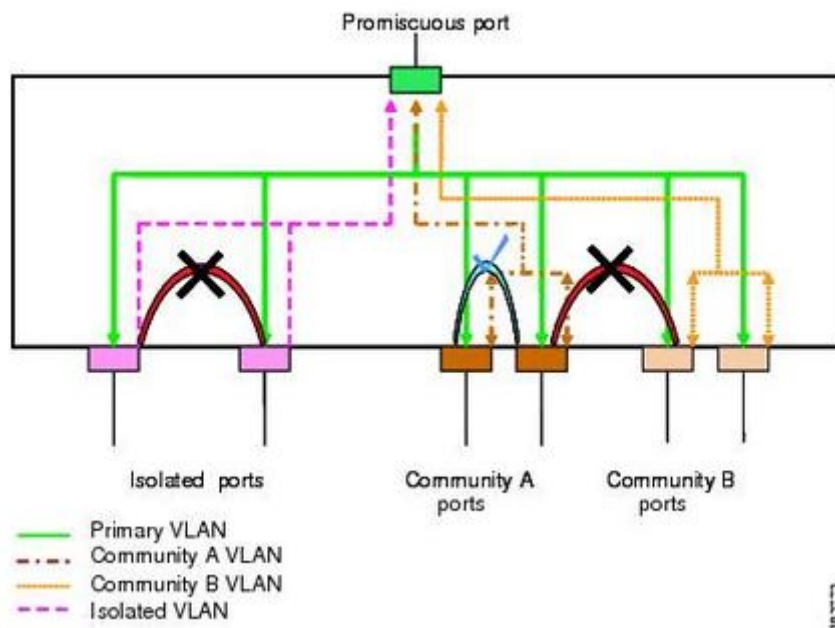


Figura 47.- Trafico de Vlans Privadas
Fuente: (CISCO SYSTEM, 2019)

En provecho de las características de esta técnica se consigue dar aislamiento y seguridad el tráfico de las máquinas virtuales, nodos de cómputo, Vcenter, etc. y componentes que pertenece a la Vlan de administración. Seguidamente con el diseño se necesita una Vlan que brinde el servicio vSphere VMotion técnica que ayuda a la migración dinámica de las cargas de trabajo de un nodo de cómputo a otro sin indisponibilidad de estos, técnica que la toparemos más adelante en el desarrollo de la capa virtual.

Es fundamental nunca perder la gestión de los equipos ante un inconveniente en la red sobre la Vlan de administración, de esta necesidad nos vemos en la obligación de implementar una Vlan para administración fuera de banda de los nodos de cómputo, a esta Vlan se conectara los equipos mediante su interfaz de administración fuera de banda en nuestro caso en los equipos Lenovo se la conoce como BMC (Baseboard Management Controller) hace posible acceder al servidor a revisar si algún inconveniente se presenta en la red de administración. Por último, se necesitará una Vlan de carga de trabajo por la misma se trasportará todo el tráfico de producción de las máquinas virtuales. A continuación, en la Tabla 23 un resumen de las Vlans que se van a utilizar en el ambiente.

Item Vlan ID	Descripción.
100, 20, 30	Vlan ID 100 Vlan Primaria, Vlan ID 20 Vlan de comunidad, Vlan ID 30 es la Vlan aislada.
410	Vlan ID 410 Vlan de vMotion.
666	Vlan ID 666 Administración fuera de banda.
688	Vlan ID 688 Vlan de producción.

Tabla 23.- Asignación de Vlans.

Fuente: Propia.

Al momento de diseñar una red existen varios aspectos que considerar, cisco tiene muchas alternativas y modelos de diseño de acuerdo a su necesidad, pero todas ellas se basan en una base fundamental al momento de presentar un modelo, independientemente de la familia de equipamiento que se utilice, en base en este principio para cisco es fundamental que se respete la arquitectura de red para que un diseño de red sea confiable, que conlleva a considerar cuatro aspectos fundamentales que se exponen en la Figura 48.



Figura 48.- Arquitectura de Red
Fuente:(Angel Perez, 2019)

En la red, por la criticidad de los servicios brindados debe estar siempre disponible, justamente la tolerancia a fallas pretende garantizar la operación ante un fallo ya sea en un equipo o interfaz, en los servicios del Centro de Datos es una obligación que los equipos se configuren el alta disponibilidad, los enlaces sean redundantes y los protocolos garanticen el correcto funcionamiento ante una falla, justamente por esa razón se debe considerar en la implementación equipos con redundancia no solamente a nivel de red sino de todos los componentes que intervienen en el diseño.

La escalabilidad también es un elemento importante, la red debe crecer y adaptarse a los cambios fácilmente sin la necesidad de una reconfiguración o mucho menos indisponibilidad en los servicios ante estos cambios, la configuración y los protocolos deben tener la capacidad de converger ante los cambios en caliente.

La calidad de servicio QoS (Quality of Service) es un término que cada vez se vuelve más trascendente en el diseño de la red, la multimedia demanda de un gran ancho de banda que los datos tradicionales, al existir encolamiento en las redes de datos esta técnica nos permite dar prioridad de atención en los sistemas, es importante que QoS se implemente en una red del Centro de Datos, ya que múltiples servicios atraviesan por ella y es necesario se establezca la calidad de servicio.

Algo muy crítico es la red es la seguridad de esta, se debe garantizar el blindaje de los equipos de red, para evitar accesos no autorizados y puedan realizar cambios e incluso robar información, los datos deben ser protegidos de tal manera que los mismos mantengan su confidencialidad, disponibilidad e integridad.

En función a esto cisco nos proporciona modelos topológicos recomendados para el Centro de Datos que cumplen este principio, y con en base al prototipo y requerimiento se va a adoptar un modelo básico que se basa en capas de red, este modelo ha sido validado y probado por varios centros de datos en el mundo. Las capas para implementar son: una capa de núcleo, una capa de agregación y una capa de acceso como se muestra en la Tabla 24. (CISCO SYSTEM, 2016)

Capa	Descripción.
CORE	Capa donde se ejecuta el BackPlane y los protocolos de capa 3 de alta velocidad, es el que controla el flujo de datos que ingresan y salen del Centro de Datos, es el cerebro de la red. Debe ser un equipamiento de alta gama con alta disponibilidad, es el que ejecuta los protocolo ospf o bgp y el equilibrio de la carga de tráfico.
Agregación	Es el nivel que proporciona el trabajo más duro, tiene funciones muy importantes como la integración de los módulos de servicio, el control de la capa 2. Y la redundancia de los enlaces. El flujo de los datos de punto inicial a punto final a traviesa por esta capa y debe tener la capacidad de mantener un flujo controlado para proteger la red y las aplicaciones, es esta capa esta la demarcación de la capa2 y capa 3 tanto para el tráfico de norte a sur y el tráfico de este a oeste.
Acceso	La capa de acceso es donde se conectan todos los dispositivos con distinto propósito como: servidores blade con conmutadores integrales, servidores blade con cableado de paso, servidores agrupados, posiblemente mainframes. En los centros de datos modernos, esta capa se divide además en una capa de acceso virtual que utiliza redes basadas en hipervisores.

Tabla 24.- Redes Cisco basado en capas.
Fuente: (CISCO SYSTEM, 2016)

En función a este principio en la Figura 49 se exhibe la topología teórica que cisco nos recomienda para implantarlo en el Centro de Datos.

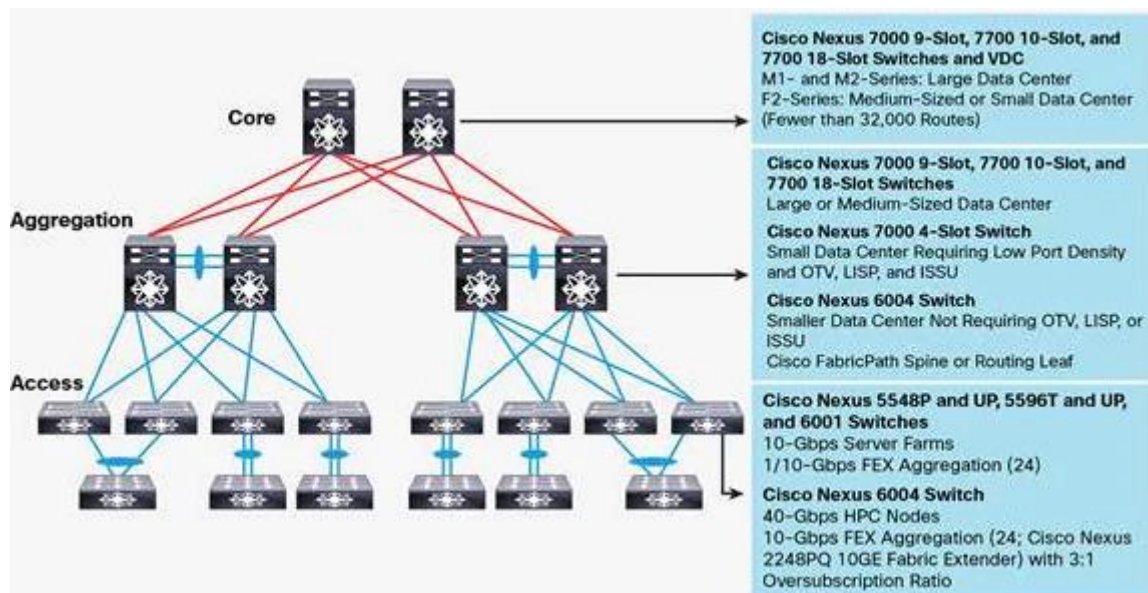


Figura 49.- Topología de Red Recomendada por Cisco.
Fuente:(CISCO SYSTEM, 2016)

En la teoría lo ideal sería implementar las tres capas, pero para en este caso de estudio y por las particularidades de los equipos es posible realizar una modificación en la red donde cisco lo llama núcleo colapsado, en esta variante lo que se hace es unir las capas de Core y agregación en una sola, así optimizando los recursos y manteniendo la jerarquía de la red.

Como se lo viene mencionado todas las tecnologías implementadas en el Centro de Datos deben constar de alta disponibilidad y enlaces redundantes, eso nos obliga a que se debe implementar todo en pares, es decir dos equipos en la capa de Core-agregación, dos equipos en la capa de Acceso a donde se debe conectar los equipos en Alta redundancia, la tecnología Nexus nos proporciona la característica de crear enlaces redundantes virtuales que se la conoce como vPC (Virtual Portchannel).

Aprovechando esta propiedad se acogerá uno de los modelos topológicos que cisco ofrece que se lo llama como "Single-Homed Server and Host vPC", esta topología es ideal para los servidores que se tiene disponible, ya que maneja varias NIC's y mediante la capa virtual se puede configurar el protocolo 802.3ad. En la Figura 50 se puede observar en la parte superior el par de Switch`'s Nexus que conforma el Core-agregación, y en la parte inferior la capa de acceso, los enlaces se conforman de manera redundante tanto los equipos de red como los nodos de cómputo.

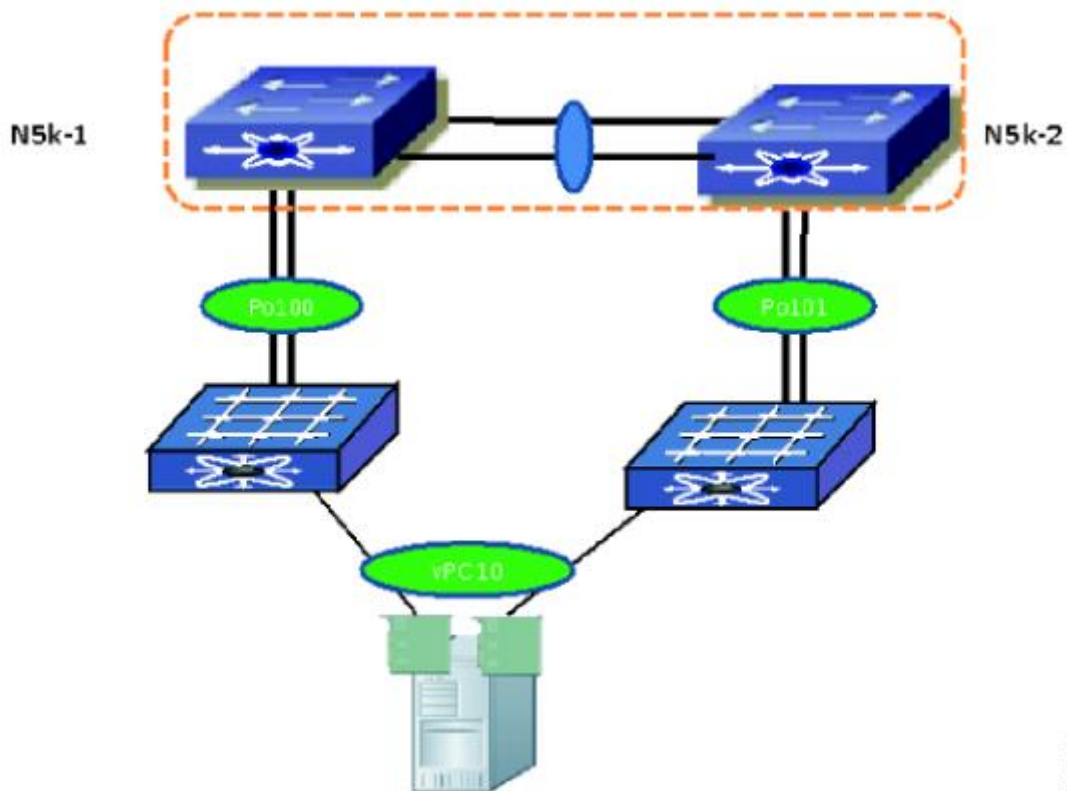


Figura 50.-Single-Homed FEX Topology
 Fuente: (cisco, 2018)

De acuerdo con la teoría revisada y a los recursos disponibles a continuación en la Tabla 25 se en lista los equipos que se usarán en la implementación del prototipo. Se contempla cuatro equipos N5K que conformaran el Core-agregación de la solución tanto para la red de Administración como la red de Producción, cuatro equipos N2K con interfaces de 10 Gbps que conformaran la capa de acceso a la red de Administración y la red de Producción, adicional un N2K con interfaces de 1 Gbps que conformara la capa de acceso a donde se conectará una interfaz en nuestros nodos de cómputo para la administración fuera de banda.

Capa	Modelo	Cantidad	DESCRIPCION
CORE-AGREGACION	N5K-C5672UP	4	Cisco Nexus 5672UP: 48 puertos SFP+ de 10 GE, 6 puertos QSFP+ de 40 GE compatibles con FCoE de 10 y 40 GE y 2/4/8/16 Gbps Fibra Canal. Los primeros 24 puertos pueden admitir 1GE.
ACCESO	N2K-C2232PP-10GE	4	32 interfaces de host 1/10 GE y FCoE (SFP+) y 8 interfaces de 10 GE y FCoE (SFP+)
ACCESO OUT-BAND	N2K-C2248TP-E-1GE	1	48 interfaces de host 100/1000BASE-T y 4 interfaces de estructura de 10 Gigabit Ethernet (SFP+)

Tabla 25.- Equipamiento Adquirido.
Fuente: Propia

De lo estudiado, se tiene las condiciones para diseñar la topología necesaria para implementar el prototipo planteado, en la Figura 51 y Figura 52 se presenta el diagrama en bajo nivel de nuestro diseño tanto para la red de Administración como para la red Producción, adicional se puede observar un switch de acceso en la red de Administración que se va usar para la administración fuera de banda de los nodos de cómputo en la Vlan 666 como se explicó en la parte de asignación de Vlans, cabe recalcar que las Vlans 100,20,20,410 conviven en la red de Administración y la Vlan 688 convive en la red de Producción. La conexión de los nodos de cómputo se puede visualizar en la Figura 50 de la sección anterior.

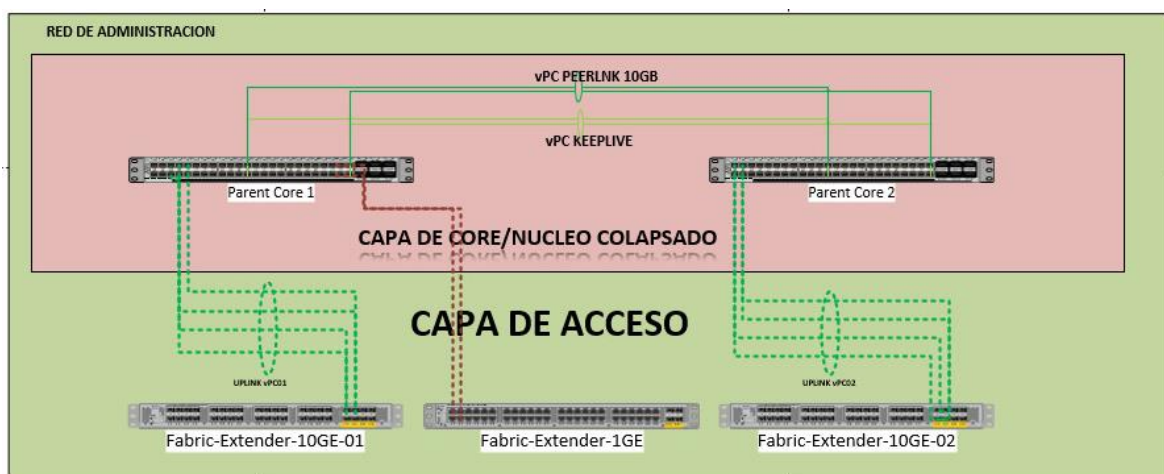


Figura 51.- Red De Administración
Fuente: Propia

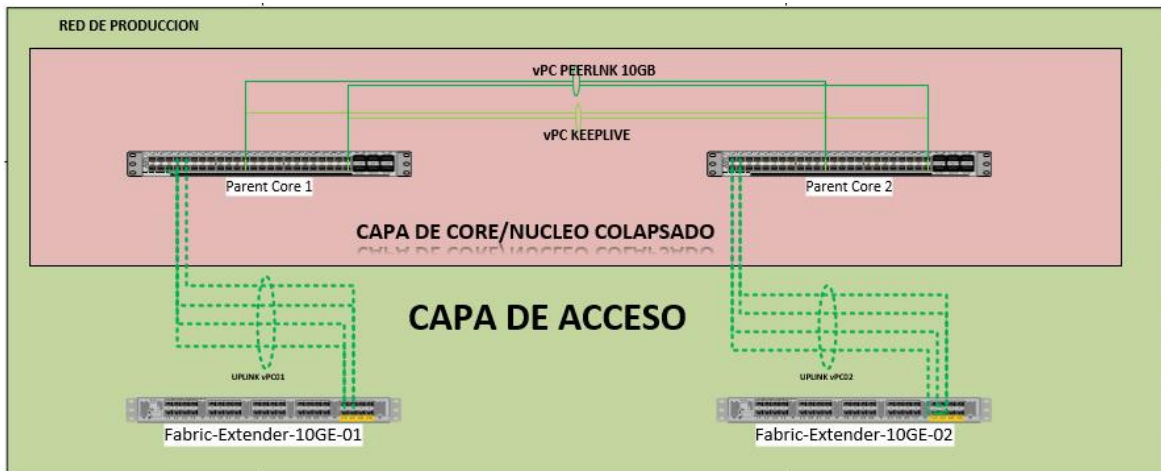


Figura 52.- Red de Producción.
Fuente: Propia.

Una vez definida la topología es momento de asignar los recursos de red y establecer el direccionamiento que será utilizada en cada uno de las Vlans asignadas previamente, a continuación, en la Tabla 26 se expone el direccionamiento IP.

ítem Vlan ID	Red IP
100, 20, 30	10.225.48.0/21
410	10.250.8.1/21
666	10.40.40.1/23
688	172.10.10.1/24

Tabla 26.- Recursos IP.
Fuente: Propia.

2.6.3 Diseño y Configuración de la Red SAN Fibra Canal.

En esta sección se verá el diseño y configuración de la red SAN (Storage Area Network), el acceso al espacio de almacenamiento es algo fundamental para el funcionamiento de la infraestructura, ya que es donde se alojará los datos siendo estos de operación de las máquinas virtual como data activa para la producción de una aplicación, por ende, el acceso al almacenamiento debe estar siempre disponible.

Por esa razón el diseño de la Red SAN es fundamental para un correcto funcionamiento, esta debe contar con alta disponibilidad y con equipos de alta gama que logre transacciones o velocidades muy altas, en nuestro caso todo el diseño se basa en fibra canal con interfaces de 16 Gbps con conexiones de alta redundancia en fibra multimodo del cual la desarrollaremos más adelante.

De la misma manera como en las secciones anteriores el diseño y la configuración será apegada a las buenas prácticas que el fabricante nos recomienda, el objetivo es traducir estas recomendaciones a los recursos que disponemos en el Centro de Datos, VMware nos permite usar una Red SAN como una alternativa de acceso a los sistemas de almacenamiento la idea es emular un espacio de almacenamiento como si estuvieran estos conectados directamente a los equipos de cómputo, otra ventaja que da una Red SAN es que se puede tener centralizada toda la gestión de almacenamiento, permitiéndonos el movimiento de datos entre servidores de una manera más rápida y efectiva así mismo copias de seguridad o restauración de los datos.

Esta técnica es muy recomendada y usada en los centros de datos por su elasticidad y escalabilidad que nos ofrece, los equipos se pueden agrupar en uno o varios racks y por sus interfaces de fibra conectar a largas distancias, puede crecer en cualquier momento con la agregación de más equipamiento, la Red SAN maneja el protocolo FC (Fibra Canal) que es mucho más rápido que manejar almacenamiento por IP.

Cuando se agrupa varios equipos de red San que trabajan en fibra canal a esta topología se la conoce como Fabric FC, son múltiples conmutadores interconectados mediante fibra óptica y dependiendo del fabricante el número de conmutadores pueden ir de 2 a 239 conmutadores, el protocolo FC es el que comanda todas las órdenes de la red. Existen componentes básicos que nos recomienda VMware el momento de diseñar una red San como son: Servidores, Storage Array, SAN Switches, Host Bus Adapters, Storage Processor que los vamos revisando más adelante. (VMware, 2010).

Lo que exige la recomendación del fabricante es que el equipo de cómputo tenga un hardware dedicado de dual FC HBA, los arreglos de los conmutadores y almacenamiento tengan múltiples rutas de acceso al espacio de discos. Para el diseño lo dividiremos en 3 grupos de componentes: Componentes de Host, Componentes Fabric, Componentes de Storage, como se muestra en la Figura 53.

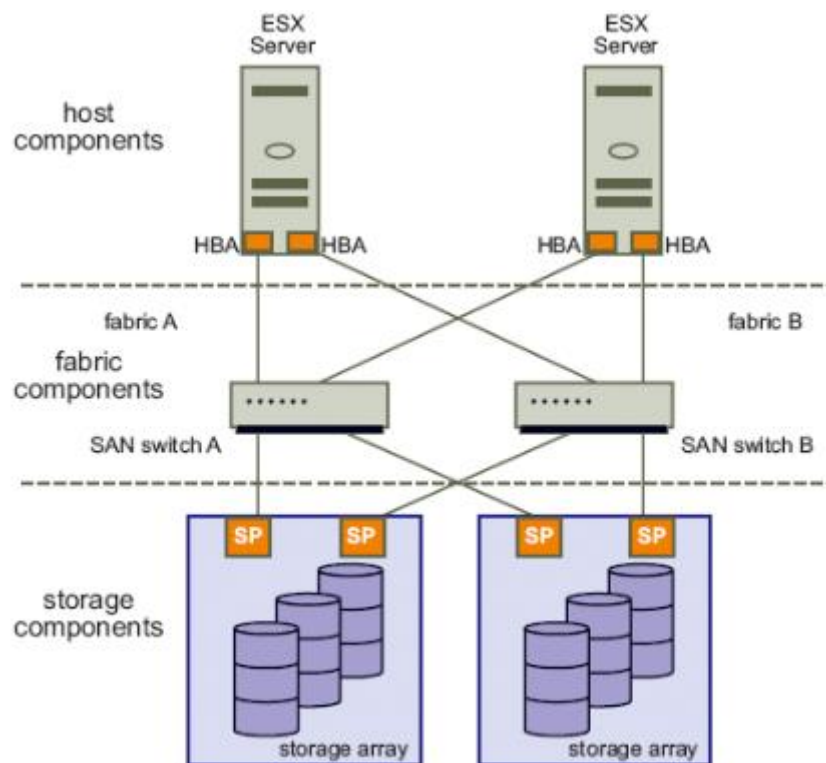


Figura 53.- Componentes SAN
Fuente:(VMware, 2010)

De esta manera se garantiza la continuidad de acceso al almacenamiento, el servidor de cómputo cuando necesita acceder a un espacio de disco envía tramas de acceso directamente desde el servidor a la controladora de arreglo de discos. El Storage responde a través de comandos SCSI que se encapsulan en tramas de FC, el host a través de sus HBAs des encapsulan la trama y acepta los datos que previamente fueron codificados en 8 o 10 bits que se transmiten por la fibra óptica.

De acuerdo con las guías revisadas y a los recursos proporcionados por el Centro de Datos, en la Tabla 27 se detalla el hardware que se va a utilizar para la implementación de la Red SAN.

Equipo	Modelo	Cantidad	DESCRIPCION
Switch SAN	Brocade 6510	2	Brocade 6510, 48P, PORT SIDE EXHAUST AIR FLOW, 16 Gbps: requiere SFP+ conectable en caliente de Brocade, conector LC; 16 Gbps/32Gbps.
HBA`s	QLogic 2692	6	Dual-Port 16Gb Fibra Canal (FC) Host Bus Adapters (HBAs)
Almacenamiento	VSP-G700	1	Equipped with 32/16-Gbps Fibre Channel or 10-Gbps iSCSI interface facilitates high-speed data transfer, RAID 1, RAID 5, and RAID 6 support (RAID 6 including 14D+2P), Equipped with high capacity cache to provide a total of 512 GB of high-speed processing.

Tabla 27.- Equipos Red SAN.

Fuente: (Brocade, 2018; Dell, 2020; Hitachi., 2021)

De esta manera en la Figura 54 se exhibe el esquema en bajo nivel de la red SAN para el funcionamiento del prototipo planteado. De esta manera cumplimos con las recomendaciones como se puede visualizar todas las interfaces se encuentran en redundancia, la VSP (Virtual Storage Platform) consta de 2 controladoras con 4 Drivers cada una, cada controladora se interconecta con los conmutadores SAN a una velocidad de 16Gbps en fibra multimodo, el nodo de cómputo se interconecta con dos Hba`s una a cada Conmutador igualmente en Fibra multimodo a 16 Gbps, en la imagen consta un solo nodo pero en realidad es la misma configuración de interfaces para los otros dos nodos.

Cada camino es determinado como una ruta, cada nodo de cómputo tiene acceso a un espacio de disco o volumen a través de estas rutas o caminos, cuando un equipo tiene más de una ruta de acceso se lo conoce como multipathing. Por defecto VMware Esxi maneja un control de las rutas por falla, es decir accede al disco por una ruta y si llega a falla esta conmuta a la otra que se encuentra en espera, en nuestro caso se sacará el máximo provecho a las múltiples rutas y se manejará un esquema de balanceo por algoritmo Roud-

Robin que consiste en balancear las cargas en todas las rutas disponibles esta configuración ya se revisará más adelante en la capa virtual.

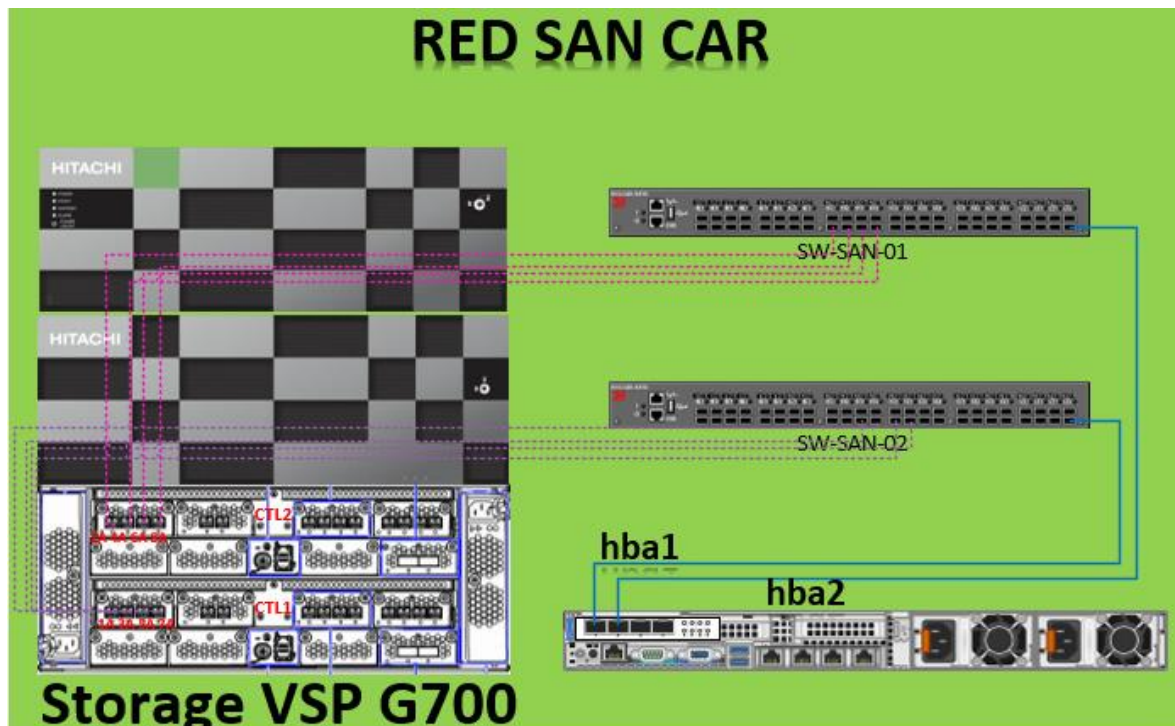


Figura 54.- Topología de la RED SAN.
Fuente: Propia.

Se ha revisado lo que corresponde a los componentes físicos e interconexiones de la red SAN, pero como toda red ya sea de datos o en este caso SAN se debe realizar una configuración lógica, en este caso el término con el que se familiarizará es la zonificación, que básicamente consiste en agrupar todos los componentes en una zona para que exista la comunicación entre ellos, dentro de la zonificación se usará la técnica de Alias, que quiere decir que a cada WWN (World Wide Node Name) específico se nombrará para que pueda ser reconocido a donde pertenece. Los WWN es la huella digital del puerto esta es única en el mundo.

La zonificación es un método seguro y muy común usado en los centros de datos, ya que las tramas y protocolos son solo difundidos a los elementos dentro de la zona y a los elementos que lo conforman, el tráfico de estas zonas está totalmente aislado de las otras, en nuestro caso se definen los Alias y el nombre de la zona y se detallan en la Figura 55.

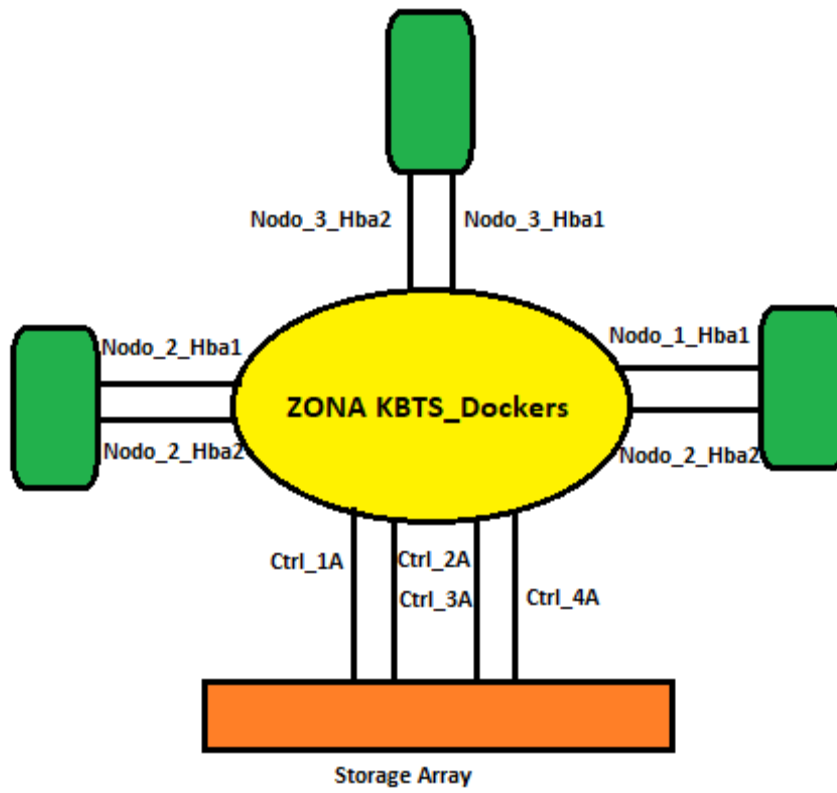


Figura 55.- Asignación de Zonas y Alias.
Fuente: Propia

CAPÍTULO III

3. FUNCIONAMIENTO.

En el capítulo III se encontrará las pruebas del funcionamiento de la plataforma, para realizar las pruebas se ejecutarán uno o dos aplicaciones para validar el funcionamiento y experimentar el uso de la consola gracia, así como la ejecución de los contenedores en el clúster Swarm, por último, se evidenciará los resultados de las aplicaciones ejecutadas en los contenedores.

3.1 Pruebas del funcionamiento Prototipo

Se va a realizar el despliegue de un contenedor en modo servicio, es decir mediante Portainer se va a ejecutar un contendor, la aplicación se trata de la simulación de una lista de compras, en la misma se puede agregar productos y eliminar los mismos, el puerto en el que atiende la aplicación es port:3000:4000.

Como se ha explicado cuando se ejecuta un contendor, en primer lugar, Docker intenta ubicar la imagen en el sitio público de Docker, en este caso se va a registrar un sitio en Docker-Hub para descargarlo desde ese sitio privado y publicándolo para la prueba, la idea es probar el funcionamiento del registro de sitios para la descarga de imágenes.

En primer lugar, la Figura 56 muestra la consola de bienvenida en donde se deberá ingresar el usuario y contraseña, cada cliente se le proporcionara su usuario correspondiente en este caso el que se va a utilizar es el de administración global de la plataforma.

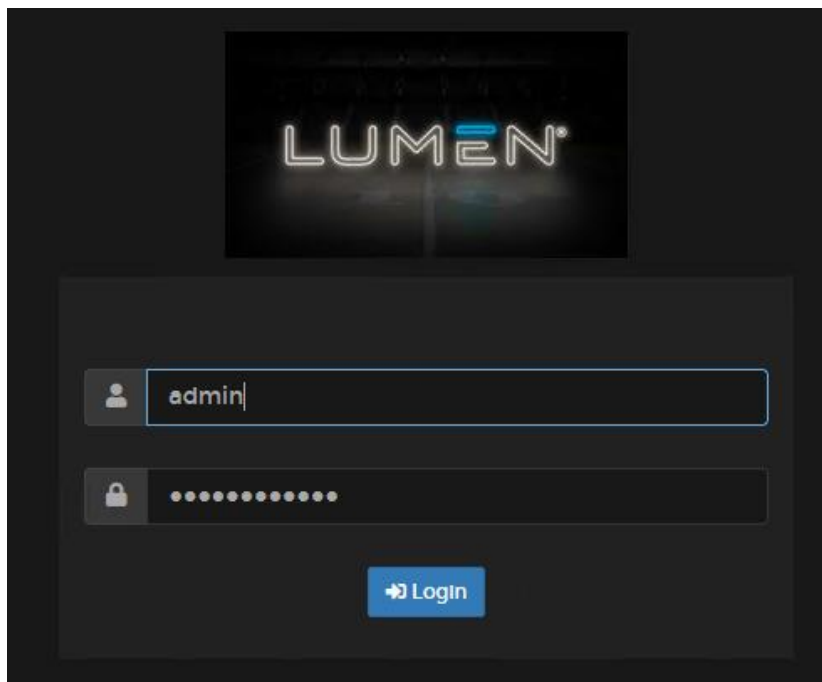
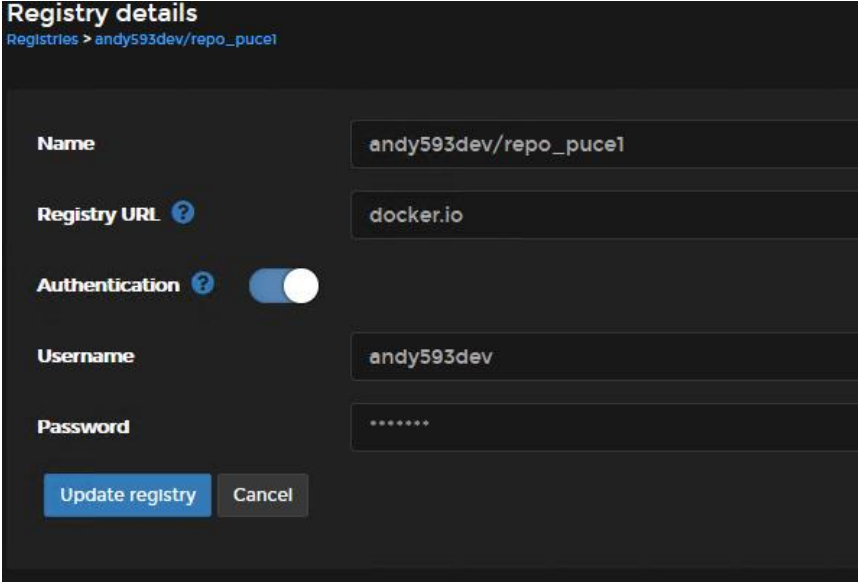



Figura 56.- Inicio de Plataforma Portainer.
Fuente: Propia.


Se realiza el registro del sitio en Docker-Hub, como se observa en la Figura 57 agregamos el sitio en Portainer como un sitio alternativo para la descarga de la imagen del contenedor que lo vamos a correr. El repositorio lo llamaremos "repo_puce1". Este paso no es obligatorio como lo explicamos al ejecutar un contenedor Docker la imagen es buscada en el sitio público, en general los desarrolladores tienen sus propios repositorios donde guardan sus aplicaciones y contenedores, los sitios soportados son Docker-Hub, AWS-ECR, Quay.io, ProGet, Azure, GitLab, y una opción para personalizar y agregar cualquier sitio.



Registry details
Registries > andy593dev/repo_puce1

Name andy593dev/repo_puce1

Registry URL  docker.io

Authentication 

Username andy593dev

Password

Figura 57.- Registro de sitio Docker-Hub en Portainer.
Fuente: Propia.

Para el proceso como se aprecia en la Figura 58, en la parte izquierda de la consola se elige la forma de despliegue como un servicio, igualmente como se ha venido hablando los contenedores se los puede desplegar como contenedores, como un servicio, o como un Stack, en este caso es un servicio.

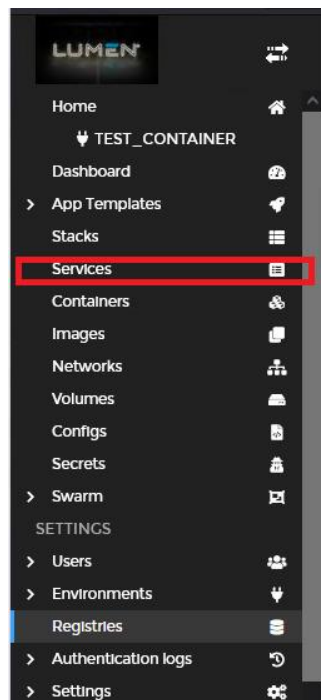


Figura 58.- Despliegue de contenedores como un servicio.
Fuente: Propia.

Como se puede observar en la Figura 59, se configura un nombre al servicio y el modo de despliegue, un contenedor se lo puede desplegar en modo global o en réplica, en modo global los servicios se despliegan en todo el clúster incluyendo los másteres, en modo de réplica los servicios se ejecutan dependiendo el número configurado, en este caso la réplica es de 1, es decir; el servicio va a estar corriendo en un nodo Worker.

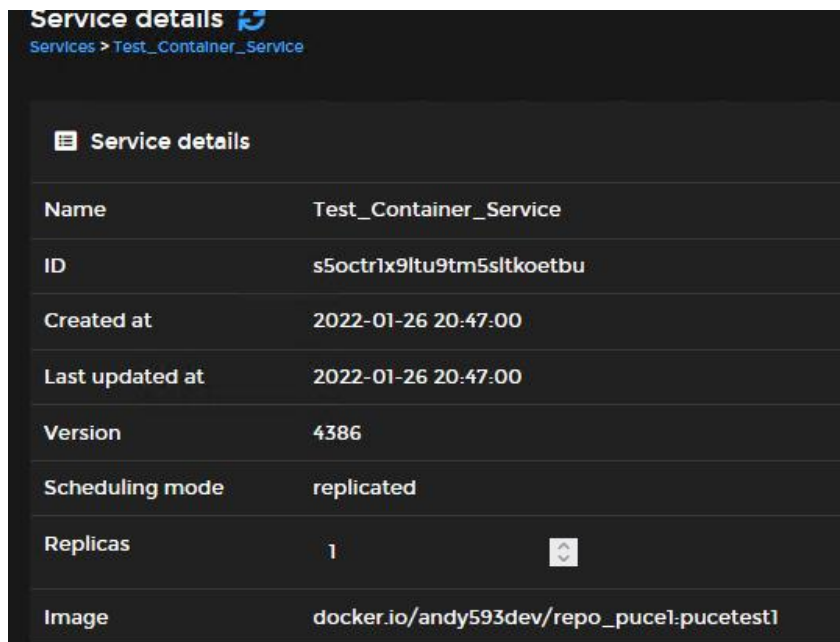


Figura 59.- Detalles del despliegue de servicios.
Fuente: Propia.

Como se muestra en la Figura 60, la imagen del contenedor la descarga del registro que fue creado.

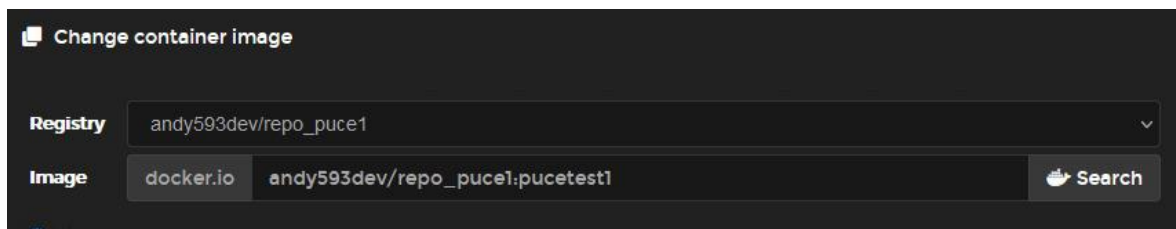


Figura 60.- Descarga de la imagen del contenedor.
Fuente: Propia.

Se configura por el puerto que va a atender al servicio como se aprecia en la Figura 61, en el caso sería el puerto 4000.

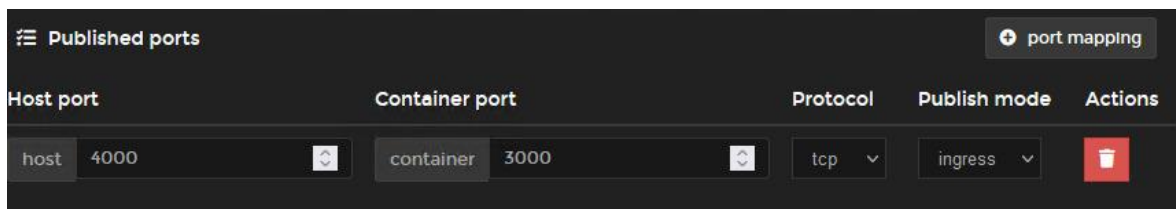


Figura 61.- configuración del Puerto del servicio de Prueba.
Fuente: Propia.

Como opcional se puede calibrar el uso de la CPU y RAM como se exhibe en la figura 62, para el caso se lo deja por defecto, sin límites.

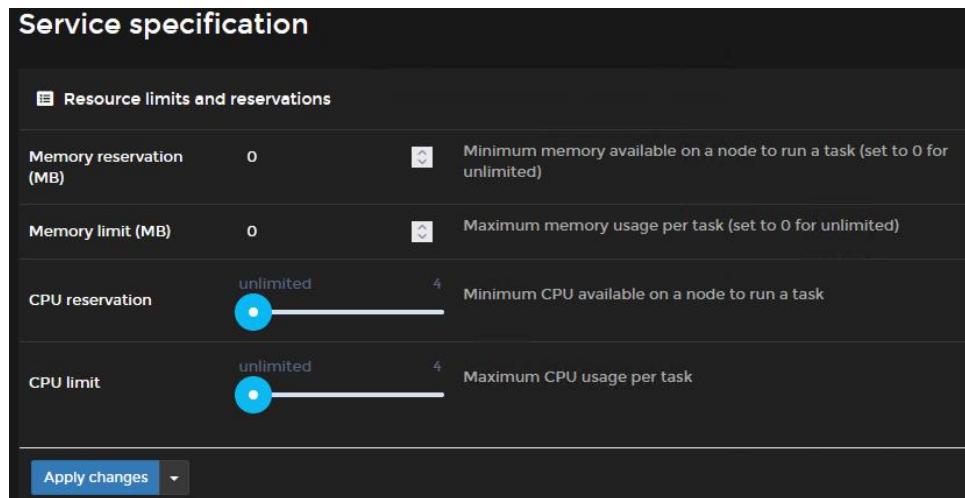


Figura 62.- Calibración de límites de consumo de CPU y RAM de la aplicación de prueba.
Fuente: Propia.

En este caso al ser el despliegue de un contenedor de prueba del ambiente la mayoría de las opciones se los dejo por defecto, ya en un entorno un poco más real los parámetros las debería establecer el desarrollador de la aplicación. Como se observa en la Figura 63 el servicio se encuentra corriendo, el nombre del servicio, desde que sitio fue obtenida la imagen, el número de réplicas y en que nodo se está ejecutando, el servicio estaría lista para su uso, el tiempo que demoro en el despliegue fue aproximadamente 2 min.

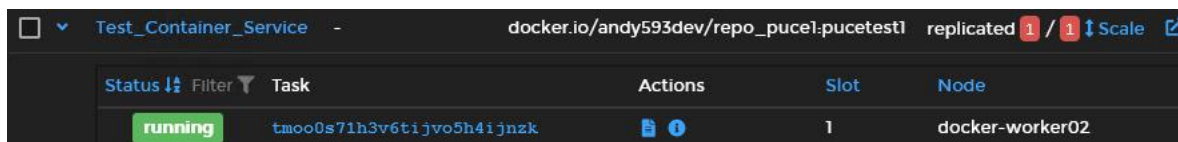


Figura 63.- Ejecución de contenedor como Servicio.
Fuente: Propia.

Por último, en la Figura 64 se usa la opción de visualización del clúster para determinar el estado del contenedor y la ejecución de este de una manera más visual y comprensible.

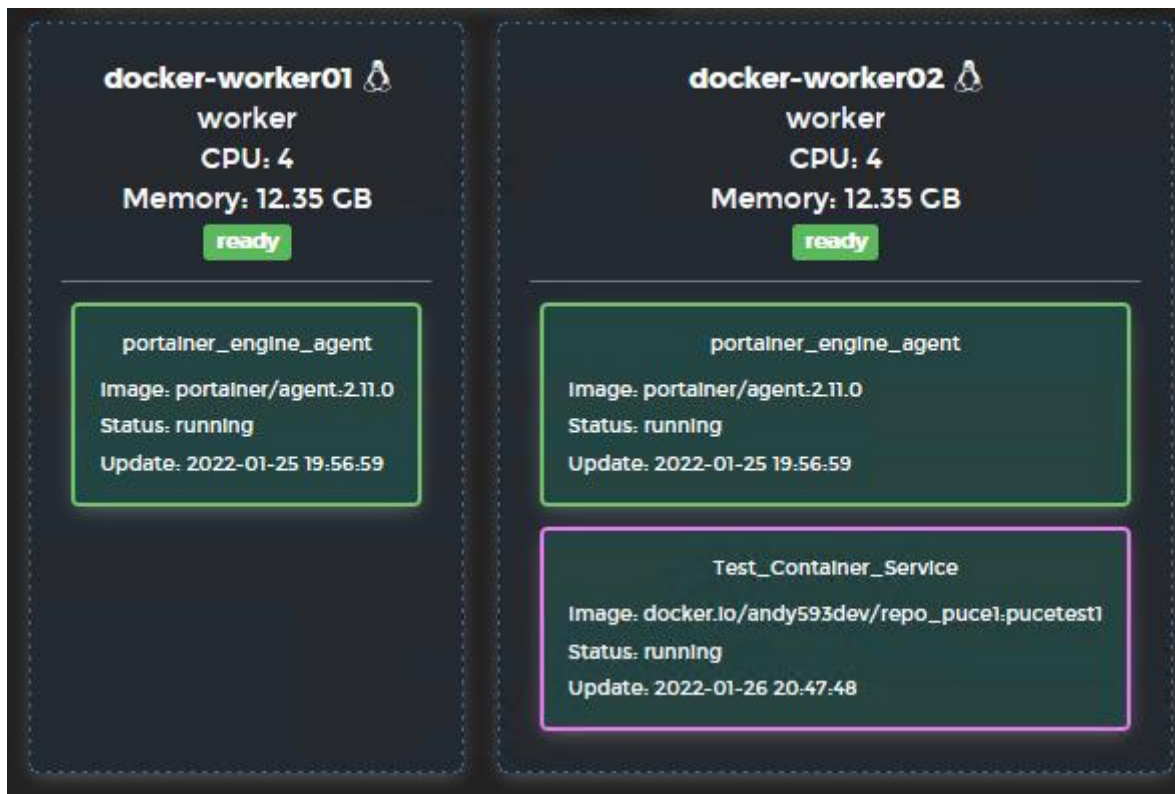


Figura 64.- Vista Grafica de la ejecución del contenedor de Prueba.
Fuente: Propia.

3.2 Resultados Obtenidos.

En la Figura 65 se evidencia la aplicación en ejecución, mediante un navegador Web y apuntando en este caso al Worker 2 con la IP: 172.10.10.20 en el puerto 4000 se despliega el servicio, se trata de una lista para el supermercado.

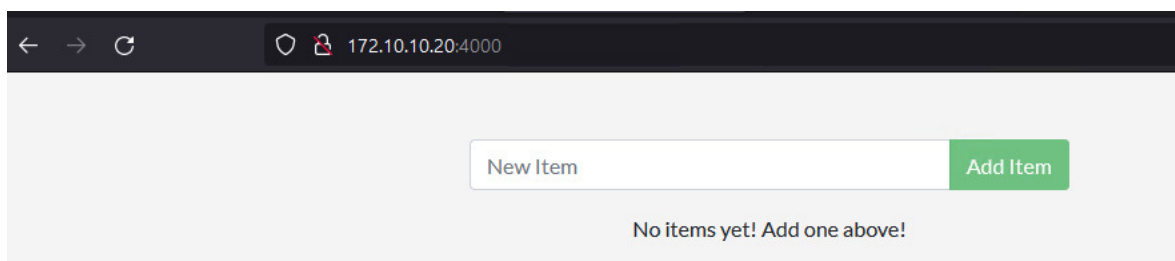
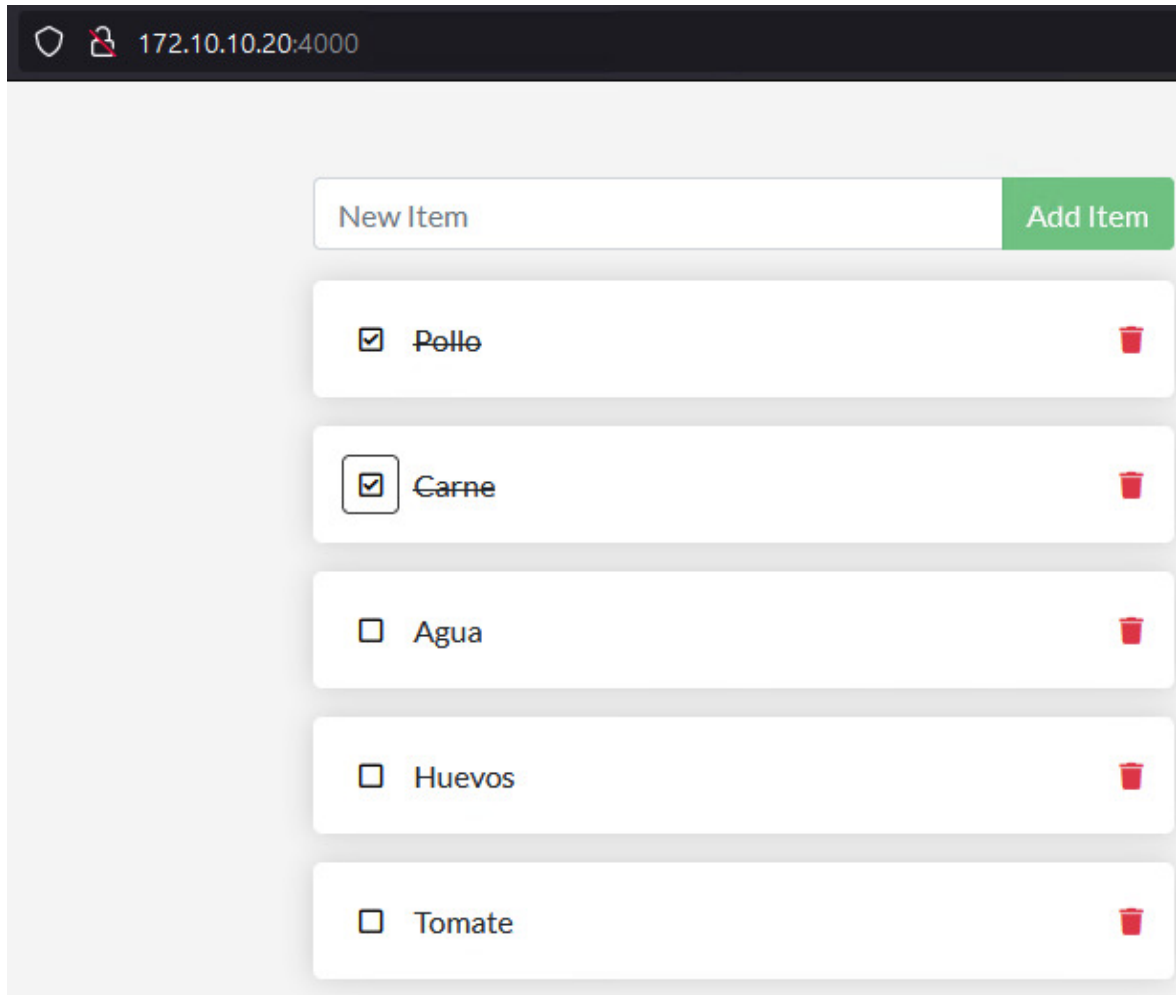


Figura 65.- Interfaz Web de la aplicación de Prueba.
Fuente: Propia.

Según la Figura 66 en la aplicación se puede agregar productos mediante el botón “Add Item”, igualmente nos permite hacer un “Check” de lo que se va comprando, y también se puede eliminar los productos, todo esto ejecutándose desde un servicio a través de un contenedor.



*Figura 66.- Uso de la aplicación de Prueba.
Prueba: Propia.*

CAPÍTULO IV

4. Análisis Financiero, Caso de Negocio para el modelo de CaaS (Container as a Service).

En el capítulo IV se encontrará un análisis del mercado, su actualidad y competencia, también se analizará los recursos desde la perspectiva de inversión y ganancia, por último, se revisará el plan y la proyección financiera para establecer costos de inversión y operación.

4.1 Análisis de Mercado

Dentro del mercado ecuatoriano existen dos tipos de proveedores que podrían brindar servicios de CaaS, los cuales son: Local Cloud Service Providers (CSP) y Public Cloud Service Provider (CSP).

Los Local Cloud Service Providers son proveedores de recursos de computación en la nube en territorio ecuatoriano, generalmente son proveedores de servicios de telecomunicaciones que han realizado inversiones en Centros de Datos y brindan modelos de IaaS gestionado.

A breves rasgos se podría citar los siguientes proveedores de capa o tier 1 de esta sección, quienes podrían sobre su esquema IaaS entregar servicios de Contenedores (CaaS).







Claro	Telconet	Lumen	Telefónica	PuntoNet	CNT
 <p>Estrategia Ampliar base de clientes Foco: IaaS Nacional y desde Colombia Oferta de Plataforma de BDD Licenciamiento Oracle y Microsoft Oferta de Facturación On Demand Complemento con BaaS</p>	 <p>Estrategia es mantener el market share de IaaS. Crecimiento: Su crecimiento es en base a Precio. Oferta de servicio local con Seguridades. Complemento con BaaS, DRaaS</p>	 <p>Estrategia de prestar servicios IaaS, SAP Hosting, Oracle Hosting y Contenedores Foco: Big Deals. Oferta con servicios Gestionados y SAP Basis Complemento con BaaS, DRaaS Gestión MultiCloud</p>	 <p>Estrategia de Ampliar Base de Cliente Venta especializada IT con Seguridades - Foco: Analítica, IaaS desde Perú y Miami</p>	 <ul style="list-style-type: none"> Estrategia en clientes medianos y grandes IaaS y Hosting SAP - Foco: IaaS + DRaaS 	 <p>Estrategia de Crecimiento en Sector Público Oferta de servicios Hosting Foco: Capacidad de Computo IAAS</p>

Figura 67.- Local CSP-Tier1
Fuente: Propia.

Como se puede apreciar en la Figura 67, Lumen Technologies forma parte de este conjunto de Cloud Service Providers locales, y el resto de los proveedores citados estarían en posibilidad de desarrollar un servicio de CaaS sobre sus entornos IaaS.

Sin embargo, también se debería tomar en cuenta los proveedores de nube pública que colocan su oferta en territorio ecuatoriano. Entre ellos se puede citar a los de la Figura 68.



Figura 68.- Public Cloud CSP
Fuente: Propia.

De toda la lista de los vendedores de Computación en la Nube Pública, Se puede destacar que los tres principales HyperScalers: AWS, Azure y GCP tienen presencia creciente en el Ecuador, por lo que podrían constituir competencia para el servicio CaaS diseñado.

En particular, se va a listar aquí a dos servicios que entregan tanto contenedores basados en Docker como RunTime como servicio bajo el mismo concepto revisado en el presente documento.

AWS Elastic Bean Stalk, este servicio contempla la entrega ya sea de un ambiente basado en Docker o inclusive ya el run time adecuado ejecutándose sobre Docker. El servicio por detrás utiliza servicios IaaS de AWS como AWS EC2, AWS EBS, AWS VPC entre otros, sin embargo, de cara al cliente todo es gestionado por AWS, y el cliente solo es responsable de los despliegues y la aplicación. (AWS, 2021)

AWS Elastic Bean Stalk, permite escoger entre ejecutar Docker, o inclusive ya un run time definido como se aprecia en la figura 69 y 70.

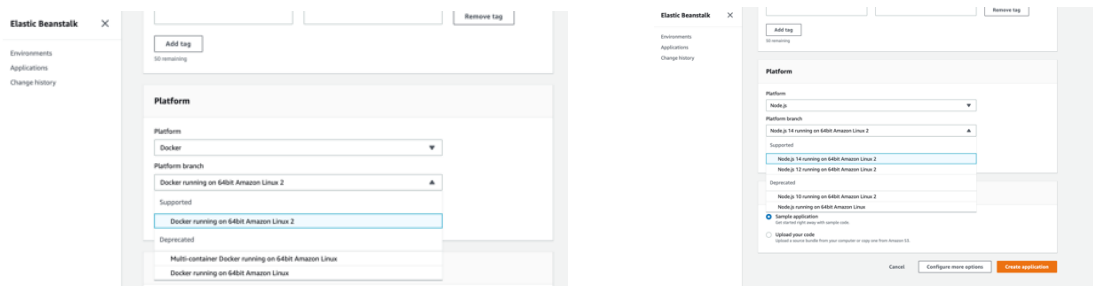


Figura 69.- Panel de configuración AWS Elastic BeanStalk.
Fuente: (AWS, 2021)

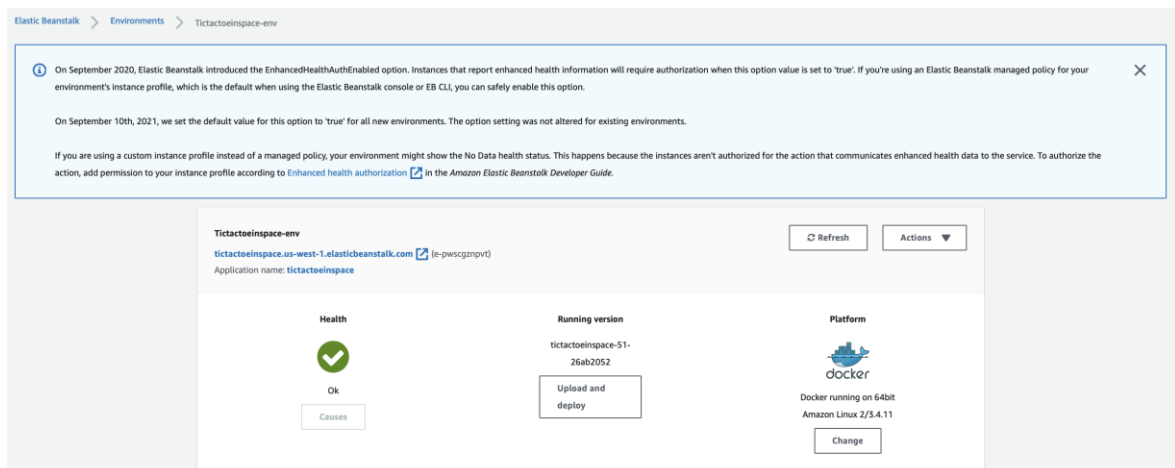


Figura 70.- Docker ejecutándose con Elastic BeanStalk
Fuente: (AWS, 2021)

Azure App Service, este servicio permite desplegar contenedores y entornos web, soporte integración con Docker, Kubernetes, y todo el toolchain de herramientas de CI/CD. De igual manera, permite que se ejecuten contenedores o runtime sobre IaaS. En este caso la gestión de todo el IaaS, y el orquestador de contenedores lo hace Azure. A diferencia de AWS Elastic BeanStalk, no se puede revisar que servicios de IaaS se usan en backend para entregar el servicio. (Microsoft, 2021).

De igual manera, permite la carga de Docker con sus respectivas imágenes de contenedores, o el run time del Stack de aplicación que se desee ejecutar como se aprecia en la Figura 71.

[Home](#) > [Create a resource](#) > [Marketplace](#) > [Web App](#) >

Create Web App ⋮

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * 

Resource Group * 
[Create new](#)


Instance Details

Need a database? [Try the new Web + Database experience.](#) 

Name * .azurewebsites.net

Publish * Code Docker Container Static Web App

Operating System * Linux Windows

Region *
 Not finding your App Service Plan? Try a different region or select your App Service Environment.

*Figura 71.- Panel de Configuración de Azure App Service.
Fuente:(Microsoft, 2021).*

Como conclusión, si bien se pueden identificar los competidores locales que podría tener el servicio diseñado en el presente documento, con respecto a Ecuador se carece de una competencia directa lo que es ventajoso, sin embargo, hay que tomar en cuenta que servicios como AWS ElasticBeanStalk, Azure App Service puede brindar competencia con la desventaja de que se ejecutan en centros de datos fuera de territorio ecuatoriano.

De igual manera para cerrar el presente análisis se consideran soluciones que son de nicho, o ya contemplan otro nivel de servicio en donde la ejecución de contenedores no es sobre IaaS, sino sobre PaaS o en esquemas serverless.



Figura 72.- Cuadrante de clasificación de servicios CaaS.
Fuente: Propia.

4.2 Organización y Recursos

Como parte de la factibilidad de entrega del presente diseño, se considera un Pod de computación para desplegar el servicio, este Pod debe considerar toda la infraestructura de computación, almacenamiento, red, backup, virtualización, ejecución de run time, operación, administración y orquestación. El Pod considera el diseño presentado en el capítulo III, ver Figura 73.

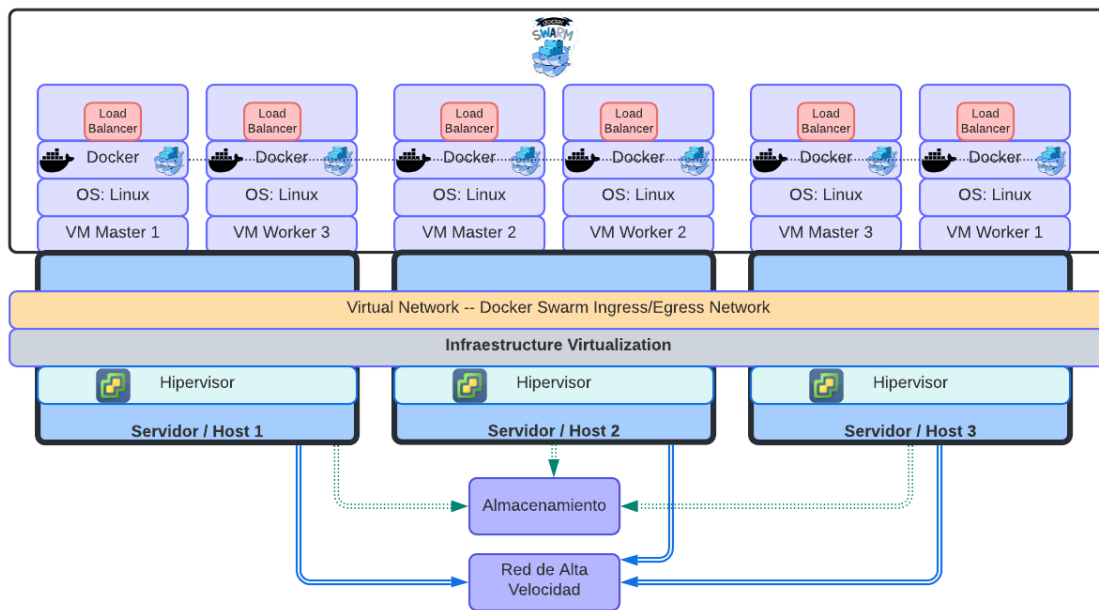


Figura 73.- POD para el Servicio CaaS.
Fuente: Propia.

Para poder brindar el servicio, Lumen Technologies debe considerar los siguientes recursos, los cuales se clasifican como Capital Expenditure (Capex) es decir la inversión de capital y como Operational Expenditures (Opex), es decir gastos operativos recurrentes en el tiempo.

Para el presente análisis se considera un caso de negocio por treinta seis (36) meses equivalentes a tres (3) años.

4.2.1 Capex.

4.2.1.1 Servidores

Se consideran los tres servidores físicos de acuerdo con el modelo seleccionado, estos servidores deben tener las especificaciones técnicas del presente diseño, además de contemplar un contrato de soporte y garantía por el plazo definido para el caso de negocio.

4.2.1.2 Almacenamiento.

Se considera la provisión de un espacio de 40TB en un esquema de discos multicapa (multi-tiering), considerando discos FMD (discos flash) y discos de estado sólido (SSD).

4.2.1.3 *Networking.*

El servicio se conectará a las redes de alta velocidad del Centro de Datos de Lumen, sin embargo, se considera la cantidad de puertos de red Loss Less Ethernet tanta para conexiones de cliente como para propósito de monitoreo y backup para el POD.

4.2.1.4 *Seguridad de Red.*

El servicio considera la colocación de un esquema de alta disponibilidad de firewalls de siguiente generación el cual será utilizado por la totalidad de los clientes del Pod.

4.2.1.5 *Hardware para Backup.*

Se considera que se debe obtener backup para la totalidad de infraestructura virtual utilizada, por lo que se considera 40TB de capacidad de almacenamiento.

4.2.1.6 *Alojamiento de Infraestructura.*

Se considera colocar el Pod para servicio CaaS dentro de un Full Rack de 19 pulgadas con capacidad para 42 unidades de rack y 4KVA de energía.

4.2.2 *Opex.*

4.2.2.1 *Licencia de Backup.*

Se considera el licenciamiento de la herramienta Veritas NetBackup para respaldar todas las VMs que son parte de la solución.

4.2.2.2 *Licencia de Monitoreo.*

Se considera licenciamiento de HP Operations Manager – SiteScope, para monitorear todas las máquinas virtuales a nivel de sistema operativo.

4.2.2.3 *Licenciamiento de Sistemas Operativos.*

Se considera las suscripciones para los Sistemas Operativos RHEL (Red Hat Enterprise Linux). Para los S.O CentOS, se considera un esquema de soporte brindado por un tercero en este caso IBM.

4.2.2.4 *Licenciamiento de virtualización de servidores.*

Se considera el licenciamiento vSphere Enterprise Plus para el clúster de virtualización. De igual manera se considera el licenciamiento de vSphere vCenter que es el plano de control del esquema de virtualización e IaaS.

4.2.2.5 *Licenciamiento de Docker Engine.*

Se pueden usar dos (2) opciones, la primera opción utilizar Docker CE (Community Edition) y adquirir soporte brindado por un tercero en este caso IBM, o la segunda opción adquirir Docker Enterprise (Mirantis), para contar soporte de marca. Ambos escenarios son válidos. Docker CE tiene reléase community cada 3 meses, mientras que Docker Enterprise (Mirantis) libera versiones cada 6 meses.

4.2.2.6 *Energía.*

Se considera, el gasto de 4KVA por mes, del rack donde funcionará la solución.

4.2.2.7 *Gestor de Networking.*

Como recurso humano, se considera tiempo del actual gestor de networking de la operación del Centro de Datos y Cloud de Lumen.

4.2.2.8 *Gestor de Virtualización.*

Como recurso humano, se considera tiempo del actual gestor de virtualización de la operación del Centro de Datos y Cloud de Lumen.

4.2.2.9 *Gestor de Sistemas Operativos.*

Como recurso humano, se considera un recurso dedicado para este servicio.

4.2.2.10 *Gestor de Contenedores y Devops.*

Como recurso humano, se considera un recurso dedicado para este servicio.

4.2.2.11 Gestor de Seguridad

Como recurso humano, se considera tiempo del actual gestor de seguridad de la operación del Centro de Datos y Cloud de Lumen.

4.2.2.12 Operación Base.

Se considera la agregación de un operador al esquema de operación 24x7 para poder abastecer este servicio.

4.2.2.13 Servicios Profesionales.

Se considera la contratación de un servicio de implementación de Docker Swarm para el arranque del Pod de servicio.

4.2.3 Plan y Proyección Financiera.

Una vez considerados todos los componentes del caso de negocio, se procede con la generación de la lista de materiales necesarios para el despliegue del Pod del servicio diseñado. La misma puede ser revisada en las siguientes tablas 28, 29 y 30.

CAPEX				
Categoría	Descripción	Cantidad	Precio Unit	Precio Total
Servidor	Servidor- Lenono- ThinkSystem SR630 - 2 x 16 Cores 2,3GHZ, 512GB RAM, 2x480 SSD Disk, 6x10GE, 2 x HBA 16GB, 2 x PS	3	\$ 9.904,00	\$ 29.712,00
Almacenamiento	Discos SSD - Raid 5 - GB RAW	26624	\$ 1,30	\$ 34.598,00
Almacenamiento	Disco FMD - Raid 5 - GB - RAW	26624	\$ 1,43	\$ 37.939,20
Networking	12 puertos de Red Privada + 12 puertos de Red Monitoreo & Backup	1	\$ 5.250,00	\$ 5.250,00
Security	1 Fortigate 500E	2	\$ 6.045,00	\$ 12.090,00
Backup	Hardware para Backup de 40TB - Veritas NetBackup Appliance	40	\$ 931,72	\$ 37.268,80
Housing	1 full Rack	1	\$ 2.400,00	\$ 2.400,00
TOTAL, CAPEX				\$ 159.258,00

Tabla 28.- Lista de Materiales 1 CAPEX.

Fuente: Propia.

OPEX ÚNICO				
Categoría	Descripción	Cantidad	Precio Unit	Precio Total
Servicios Profesionales	Servicios de Implementación de Docker Swarm	1	\$ 25.000,00	\$ 25.000,00
TOTAL, OPEX ÚNICO				\$ 25.000,00

Tabla 29.- Lista de Materiales 2 OPEX.
Fuente: Propia.

OPEX RECURRENTE - MONTH PRORATE				
Categoría	Descripción	Cantidad	Precio Unit	Precio Total
Backup	Licencia para 40 TB	40	\$ 23,00	\$ 920,00
Monitoreo	Licencia de monitoreo	6	\$ 36,00	\$ 216,00
OS	Licencia de OS RHEL - Año y/o Soporte con Tercero para CentOS (IBM)	3	\$ 137,75	\$ 413,25
Docker	Licencia de Docker Enterprise - Mirantis y/o Soporte con Tercero para Docker CE (IBM)	3	\$ 291,67	\$ 875,00
Energía	4 KVA	4	\$ 90,00	\$ 360,00
Recurso Humano	Recursos compartidos: Networking, Virtualización, Seguridad	4	\$ 333,00	\$ 1.332,00
Recurso Humano	Recurso dedicado Sistemas Operativos	1	\$ 2.500,00	\$ 2.500,00
Recurso Humano	Recurso dedicado Contenedores y Devops	1	\$ 3.000,00	\$ 3.000,00
Operación	Operador de Cloud y DC	1	\$ 850,00	\$ 850,00
TOTAL, OPEX RECURRENTE				\$ 10.466,25

Tabla 30.- Lista de Materiales 3 OPEX.
Fuente: Propia.

Se identifican tres grandes grupos, el Opex relacionado con las inversiones de capital, el opex único relacionado con los servicios de implementación necesarios para desplegar los servicios de gestión de contenedores con Docker Swarm y opex recurrentes relacionados con los contratos de soporte, licencias, energía y recursos humanos necesarios para poder brindar el servicio.

Con estos valores, se procede a generar un análisis del coste de la solución también conocido como Total Cost of Owership (TCO), con el que se puede entender la conducta de los costos directos e indirectos en el tiempo. Mediante este análisis se puede presentar el flujo de caja a la empresa y pueda considerar este servicio.

A continuación, se presenta el TCO con corte anual, ver Tabla 31.

		Containers as a Service					
COMPONENTES		VPN	Cargo Inicial	Total, Año 1	Total, Año 2	Total, Año 3	TOTAL, PROYECTO 3 AÑOS
1	CAPEX	\$ 159.258,00	\$ 159.258,00	\$ 0,00	\$ 0,00	\$ 0,00	\$ 159.258,00
2	OPEX UNICO	\$ 25.000,00	\$ 25.000,00	\$ 0,00	\$ 0,00	\$ 0,00	\$ 25.000,00
3	OPEX RECURRENTE	\$ 326.428,93	\$ 0,00	\$ 125.595,00	\$ 125.595,00	\$ 125.595,00	\$ 376.785,00
Total		\$ 510.686,93	\$ 184.258,00	\$ 125.595,00	\$ 125.595,00	\$ 125.595,00	\$ 561.043,00

Tabla 31.- TCO del Proyecto.

Fuente: Propia.

El valor presente neto – VPN contabiliza la variación del valor en el tiempo, en este caso se considera que la tasa mensual de interés efectivo es del 0,797%, mientras la tasa de retorno de inversión o interés nominal anual es del 10%, lo que permite evidenciar que el proyecto con una totalidad de USD 561.043,00 a treinta y seis meses, tiene un valor en tiempo presente de USD 510.686,93. Esto implica que el componente de inversión de capital que se realiza al inicio del proyecto, se amortiza con una ganancia adecuada obteniendo ventajas con el interés de retorno de inversión.

Ahora bien, de cara a monetizar y comercializar el servicio CaaS, se realiza el siguiente análisis de caso de negocio ver Tabla 32.

Tiempo de Contrato	36	meses
Payback Meses	24	meses
Margen de Costos	25%	
Opex Recurrente	\$	10.466,25
Opex Único	\$	25.000,00
Opex Mensualizado	\$	11.507,92
Capex Total	\$	159.258,00
Capex Mensualizado	\$	6.635,75
MRC	\$	24.191,56
Total, Revenues Teórico	- \$	870.896,00
NPV	\$	510.686,93
Revenues / NPV		1,71

Tabla 32.- Caso de Negocio.
 Fuente: Propio

Se considera, un tiempo de ejecución y/o contrato del servicio CaaS de 36 meses, con un tiempo de recuperación de inversión (payback) de veinte y cuatro meses, considerando un 25% de margen sobre los costos de acuerdo con los parámetros financieros que maneja Lumen Technologies.

Se realiza un prorrateo mensual de los valores de CAPEX y OPEX para construir el precio necesario para el servicio, y a la suma de los componentes de CAPEX y OPEX se le aplica el margen de costos establecido. En relación con a eso, para poder brindar el servicio de CaaS con una rentabilidad adecuada y recuperar la inversión, se necesitaría recibir USD 24191,56 mensuales. Calculando la ejecución de los treinta y seis meses del servicio, se tendría un revenue o ganancia total de USD 870.896,00, los cuales son superiores al valor presente neto, porque el servicio brindaría una ganancia de USD 360.209,07 correspondiente al 70% de la inversión, por lo que brindar un servicio de esta naturaleza permite recuperar la inversión y obtener ganancias considerables.

Por último, en base al valor mensual necesario, se realiza la planificación de la capacidad y plan de comercialización del servicio CaaS como se evidencia en la Tabla 33.

Service	Capacity	Unit / Per Customer	Customer Qty	Availability to Scale	Type of Revenue	Unit Price	Full Capacity Price	Comments	
Storage	40TB	250 GB	160	0	MRC	\$ 40,00	\$ 6.400,00		
Compute	192 vCPUs	1 vCPU/1000 milicores	160	32	Usage MRC	\$ 10,00	\$ 1.600,00	2 centavos por milicore	
Compute	1024 GB	8GB	160	96	Usage MRC	\$ 32,50	\$ 5.200,00	0,004 dolares por MB	
Security	N/A	N/A	160	40	MRC	\$ 6,50	\$ 1.040,00		
Internet	N/A	10 Mbps	160	N/A	MRC	\$ 32,50	\$ 5.200,00		
Management	N/A	N/A	160	40	MRC	\$ 29,70	\$ 4.751,56		
Total							\$	24.191,56	

Description	Abono Recurrente mensual
1 Ambiente de Computación con Docker: 250GB Disco, Hasta 16 Subnets, 1 IP Pública, Monitoreo, Backup, Servicio Administrado, 10 Mbps de Internet	\$ 108,70
Uso de Cómputo hasta 1 VCPU (1000 milicores)	\$ 20,00
Uso de cómputo hasta 8GB RAM (8192 MB)	\$ 32,50
Total	\$ 161,20

Tabla 33.- Comercialización del Servicio.
Fuente: Propia.

En donde, se mapea las capacidades del servicio en base al diseño realizado, y su monetización.

CONCLUSIONES

El uso de los contenedores cada vez se va popularizando más en el mundo por su facilidad de despliegue y administración, para una compañía que se dedica a la renta de servicios en la nube es una obligación de adaptarse a estos cambios, el objetivo principal de la elaboración de este proyecto es brindarle a la academia un sustento técnico para futuros proyectos en lo que concierne al uso de contenedores en la nube. Y, por otro lado, es proporcionarle de tecnologías nuevas y estén disponibles para el negocio de la empresa, el efecto de la implementación de este nuevo servicio es muy positivo para la empresa, ya que expande su portafolio de servicios y se postula como una opción más en Ecuador como distribuidor de servicios en la nube orientado a la ejecución de contenedores, a su vez posicionándose como una de las mejores marcas de tecnologías en la nube, siendo esta una respuesta positiva a la interrogante principal planteada en la identificación del problema.

Para el despliegue de una máquina virtual entran en juego muchas consideraciones el momento de implementarla, los elementos como la cantidad de RAM, CPU, disco, Sistema operativo, particionamiento y aplicaciones deben ser considerados al momento de crearla, a esto sumarle la cantidad de máquinas virtuales por cada aplicación que se ejecuta en su correspondiente Sistema Operativo, en sí la virtualización nos da herramientas y prestaciones para hacer de una manera fácil y rápida, pero aún siguen siendo tiempos elevados y molestoso para el cliente, como se pudo observar el desplegar un contenedor, servicio o un Stack completo es mucho más rápido, práctico y sencillo si lo comparamos con la creación de una nueva máquina virtual. En conclusión, el uso de contenedores optimiza de una manera rotunda el uso de recursos informáticos beneficiando directamente a la empresa, un clúster puede albergar muchos contenedores de varios clientes, siendo muy dinámico la creación y eliminación de estos, ganado tiempo y operación al cliente final.

Una vez creada la plataforma la ejecución de los contenedores se convierte en al muy práctico, y como se observó el DevOps puede crear ambientes de Desarrollo, Test y Producción solo usando contenedores, así como un grupo de contenedores de Frontend y otro grupo de contenedores de Backend y desplegarlos de inmediato, los contenedores llegaron para quedarse y ayudar a los desarrolladores de apps, esto no quiere decir que las máquinas virtuales están quedando obsoletas creo son importantes en la pila de la infraestructura, concluyendo que los contenedores no reemplazan a las máquinas virtuales más bien se complementan para subir al siguiente nivel de las Tecnologías de la

Información. En cuanto al cliente final liberándolo del mantenimiento y operación de la infraestructura su única preocupación es el desarrollo y ejecución de los contenedores.

La creación del clúster a nivel de contenedores y virtualización son cosas muy diferentes, pero son muy importantes en el funcionamiento, hacer una correcta configuración del clúster virtual garantiza la alta disponibilidad, mediante reglas de afinidad se consiguió que cada máquina se ejecute en diferentes equipos de cómputo y jamás estén juntas en el clúster, y, por otro lado, el clúster de contenedores nos da la alta disponibilidad que se necesita. Un fallo a nivel físico o un fallo a nivel lógico con esta configuración siempre va a estar disponible la aplicación.

Existen varios clientes en el Centro de Datos que ejecutan todo un pool de máquinas virtuales donde se corren aplicaciones del negocio, en base de la experiencia obtenida mediante el presente trabajo, al cliente le vendría bien migrar a contenedores, desde la perspectiva de la operación su carga operativa disminuiría, ya que solo dependería de sus DevOps para el desarrollo con esto también se haría una reducción de las máquinas virtuales al mínimo posible.

Gracias a la virtualización se pudo crear un pool de máquinas virtuales donde se implementó toda la solución del prototipo, pero se le puede sacar más beneficios a la infraestructura, es decir; en un pool se puede crear las máquinas virtuales para la solución, y también se puede crear muchos más pools para seguir entregando máquinas virtuales como el servicio tradicional, todo corriendo en la misma infraestructura. EL cliente puede tener su servicio de contenedores y también sus máquinas virtuales para su producción.

RECOMENDACIONES

En el desarrollo de este prototipo se centra especialmente en el desarrollo y configuración de la plataforma para ejecutar contenedores como se observó en el documento, si bien todas las configuraciones se apegan a las buenas prácticas, especialmente en la red se implementan técnicas de seguridad, se debe tomar en cuenta que toda esta solución debe estar detrás de un firewall para proteger las aplicaciones, contenedores, servidores, etc., se recomienda hacer el uso de un firewall de borde que administre las comunicaciones y puertos con las que se enlazan los clientes a las aplicaciones.

Como se observó en el desarrollo de este proyecto, detrás de la solución final este comprende de todo un desarrollo de ingeniería en la capa física y como se lo denominó es el músculo donde funcionan las aplicaciones, la elección de los diferentes componentes fueron clave para el funcionamiento, la red es vital, ya que por ahí se transportan todos los datos de producción y administración con el uso de 10 Gbps en los enlaces ascendentes esta dinámica de datos es muy fluida. De la misma manera el acceso a los discos mediante la SAN no se evidencia latencias en las aplicaciones que se ejecutan.

En el prototipo se hace uso del orquestador Swarm para la creación de los clústeres, para una investigación y desarrollo como es el caso la solución funciona correctamente con aplicativos Open Source, se pudo evidenciar que se ejecutan contenedores sin problema, pero para ambientes ya productivos la mejor recomendación es usar soluciones de paga que ya cuentan con un soporte de fábrica que podrían ser, OpenShift, Tanzu Kubernetes, soluciones completas que ya pueden ser alternativas para un ambiente productivo.

BIBLIOGRAFÍA

- Abdelbaky, M., Diaz-Montes, J., Parashar, M., Unuvar, M., & Steinder, M. (2015). Docker Containers across Multiple Clouds and Data Centers. *Proceedings - 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing, UCC 2015*, 368–371. <https://doi.org/10.1109/UCC.2015.58>
- Angel Perez. (2019). *Exploración de la red*. Web. https://www.uv.mx/personal/angelperez/files/2019/02/CCNA_ITN_Chp1.pdf
- Asad Ali. (2020). *orquestración de contenedores para DevOps*. Web. <https://geekflare.com/es/container-orchestration-software/>
- AWS. (2021). *AWS Elastic Beanstalk*. Web. <https://aws.amazon.com/es/elasticbeanstalk/>
- AWS. (2022). *Modelo de responsabilidad compartida*. Web. <https://aws.amazon.com/es/compliance/shared-responsibility-model/>
- ayudaley. (2020). *Software de virtualización*. Web. https://ayudaleyprotecciondatos.es/2021/05/17/software-de-virtualizacion/#Que_es_la_virtualizacion
- Azure. (2022). *Qué es la nube pública, privada e híbrida*. Web. <https://azure.microsoft.com/es-es/overview/what-are-private-public-hybrid-clouds/#overview>
- Berry, T. (2022). *Shared responsibility in the cloud*. Web. <https://docs.microsoft.com/en-us/azure/security/fundamentals/shared-responsibility>
- BOULOUKO, V. (2020). *Cisco Nexus*. Web. <http://www.firewall.cx/cisco-technical-knowledgebase/cisco-data-center/1201-introduction-nexus-family-nx-os-ios-differences.html>
- Brocade. (2018). *Brocade 6510 Switch*. Web. https://www.dataswitchworks.com/datasheets/switches/6510_ds.pdf
- Chowdhury, F. H. (2021). *The Docker Handbook*. Web. <https://www.freecodecamp.org/news/the-docker-handbook/#how-to-compose-projects-using-docker-compose>
- cisco. (2018). *Using Enhanced vPC*. Web. https://www.cisco.com/en/US/docs/switches/datacenter/nexus5000/sw/mkt_ops_guid

es/513_n1_1/n5k_enhanced_vpc.pdf

- CISCO SYSTEM. (2016). *Classic Network Design Using Cisco Nexus 9000 Series Switches*. Web. <https://www.cisco.com/c/en/us/products/collateral/switches/nexus-9000-series-switches/guide-c07-730115.html>
- CISCO SYSTEM. (2019). *Configuring Private VLANs*. Web. https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus3000/sw/layer2/503_U2_1/b_Cisco_n3k_layer2_config_guide_503_U2_1/b_Cisco_n3k_layer2_config_guide_503_U2_1_chapter_0101.pdf
- Cresswell, N. (2021). *Container Management Solution Portainer.io*. Web. <https://www.portainer.io/blog/container-management-solution-portainer.io-raises-6m-series-a-round-to-accelerate-global-expansion>
- Datos101. (2022). *Qué es un Datacenter*. Web. <https://www.datos101.com/blog/que-es-un-data-center/>
- Dell. (2020). *QLE2690/2690L*. Web. https://i.dell.com/sites/csdocuments/Shared-Content_data-Sheets_Documents/en/QLE269x_269xL_DEL_Product_Brief.pdf
- Dockers Doc. (2022). *Dockers Manuals*. Web. <https://docs.docker.com>
- Dua, R., Raja, A. R., & Kakadia, D. (2014). Virtualization vs containerization to support PaaS. *Proceedings - 2014 IEEE International Conference on Cloud Engineering, IC2E 2014*, 610–614. <https://doi.org/10.1109/IC2E.2014.41>
- Furukawa. (2021). *Topologías de conexión en Data Centers*. Web. <https://www.furukawalatam.com/es/conexion-furukawa-detalles/topologias-de-conexion-en-data-centers-que-considerar-en-la-infraestructura-de-redes>
- Gonzalez, J. M. (2015). *Qué tamaño dar a mis LUNs de almacenamiento en VMware*. Web. <https://www.josemariagonzalez.es/video-tutoriales-trucos/que-tamano-dar-a-mis-luns-de-almacenamiento-en-vmware.html>
- Hitachi. (2021). *Overview of Hitachi Virtual Storage Platform G700*. Web. https://knowledge.hitachivantara.com/Documents/Storage/VSP_G130_GF350_GF370_GF700_GF900/88-07-0x/About_Your_System/VSP_G700_Hardware/01_Overview_of_Hitachi_Virtual_Storage_Platform_G700
- IBM. (2020). *IaaS, PaaS y SaaS – Modelos de servicio de IBM Cloud*. Web. <https://www.ibm.com/es-es/cloud/learn/iaas-paas-saas>

- IDCMINI. (2022). *the-physical-savings-of-virtualization*. Web. <https://idcmini.com/the-physical-savings-of-virtualization/>
- Kharb, D. L. (2015). Automated Deployment of Software Containers Using Dockers. *Ijeter*, 4(10), 4–6.
- kionetworks. (2019). *Descubre las diferencias entre almacenamiento NAS y SAN*. Web. <https://www.kionetworks.com/blog/data-center/almacenamiento-nas-y-san>
- kubernetes. (2014). *Going back in time*. Web. <https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>
- Kumar, K., & Kurhekar, M. (2017). Economically Efficient Virtualization over Cloud Using Docker Containers. *Proceedings - 2016 IEEE International Conference on Cloud Computing in Emerging Markets, CCEM 2016*, 95–100. <https://doi.org/10.1109/CCEM.2016.025>
- Lenovo. (2021). *ThinkSystem SR630 Datasheet*. Web. <https://lenovopress.com/datasheet/es/ds0031-lenovo-thinksystem-sr630>
- Lenovo. (2021). *System tuning for VMware on x86 Servers and ThinkSystem - Lenovo ThinkSystem and Lenovo Server*. Web. <https://support.lenovo.com/us/en/solutions/ht115952>
- Lingayat, A., Badre, R. R., & Gupta, A. K. (2018). Performance Evaluation for Deploying Docker Containers on Baremetal and Virtual Machine. *Proceedings of the 3rd International Conference on Communication and Electronics Systems, ICCES 2018*, 1019–1023. <https://doi.org/10.1109/CESYS.2018.8723998>
- Mauricio Collazos. (2019). *Una guía rápida de Docker*. Web. <https://medium.com/contraslashsas/una-guía-no-tan-rápida-de-docker-2fab9243762a>
- Meysa I.T. Solutions. (2019). <http://www.meysa.es/virtualizacion-e-infraestructura1/>. Web. <http://www.meysa.es/virtualizacion-e-infraestructura1/>
- Microsoft. (2021). *App Service*. Web. <https://azure.microsoft.com/es-es/services/app-service/#overview>
- Nakagawa, G., & Oikawa, S. (2017). Behavior-Based Memory Resource Management for Container-Based Virtualization. *Proceedings - 4th International Conference on Applied Computing and Information Technology, 3rd International Conference on Computational Science/Intelligence and Applied Informatics, 1st International Conference on Big Data, Cloud Computing, Data Science*, 213–217.

<https://doi.org/10.1109/ACIT-CSII-BCD.2016.049>

- NetApp. (2021). *Qué son los contenedores*. Web. <https://www.netapp.com/es/devops-solutions/what-are-containers/#virtual-machine>
- Panic, D. (2016). *Anatomía de un vmotion*. Web. <https://dailypanic.wordpress.com/category/vmware/page/2/>
- Portainer. (2022). *Portainer architecture*. Web. <https://docs.portainer.io/v/ce-2.11/start/architecture>
- Preeth, E. N., Mulerickal, J. P., Paul, B., & Sastri, Y. (2016). Evaluation of Docker containers based on hardware utilization. *2015 International Conference on Control, Communication and Computing India, ICCCI 2015, November*, 697–700. <https://doi.org/10.1109/ICCC.2015.7432984>
- Red Hat. (2018). *Qué es cloud computing*. Web. <https://www.redhat.com/es/topics/cloud>
- Red Hat. (2020). *¿Qué es IaaS?* Web. <https://www.redhat.com/es/topics/cloud-computing/what-is-iaas>
- RedHat. (2018). *Qué es la virtualización*. Web. <https://www.redhat.com/es/topics/virtualization/what-is-virtualization>
- RODRÍGUEZ, T. (2019). *Contenedores*. Web. <https://www.xataka.com/otros/docker-a-kubernetes-entendiendo-que-contenedores-que-mayores-revoluciones-industria-desarrollo>
- Singh, S., & Singh, N. (2017). Containers & Docker: Emerging roles & future of Cloud technology. *Proceedings of the 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology, ICATccT 2016*, 804–807. <https://doi.org/10.1109/ICATCCT.2016.7912109>
- SNIA. (2020). *What Is a Storage Area Network (SAN)?* Web. https://www.snia.org/education/storage_networking_primer/san/what_san
- Speroni, R. (2018). *Alta disponibilidad con Docker Swarm*. Web. <https://medium.com/@raulsperoni/alta-disponibilidad-con-docker-swarm-traefik-y-volumenes-compartidos-22ac7d876fdf>
- Veracierta, O. R. (2018). *Estrategia para Despliegue de Contenedores*. Web. <https://es.slideshare.net/olafrv/estrategia-para-despliegue-de-contenedores-agiledevops>

- VMware. (2010). *SAN_Design_and_Deployment_Guide*. Web.
https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/SAN_Design_and_Deployment_Guide.pdf
- VMware. (2021a). *Performance Best Practices for VMware vSphere 7.0*. Web.
<https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/performance/vsphere-esxi-vcenter-server-70-performance-best-practices.pdf>
- VMware. (2021b). *VMware Compatibility Guide*. Web.
<https://www.vmware.com/resources/compatibility/search.php>
- VMware vSphere. (2019). *vsphere-esxi-vcenter-server-65-networking*. Web.
<https://docs.vmware.com/es/VMware-vSphere/6.5/vsphere-esxi-vcenter-server-65-networking-guide.pdf>
- VMware vSphere. (2021). *vCenter Server Installation and Setup*. Web.
<https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vcenter.install.doc/GUID-88571D8A-46E1-464D-A349-4DC43DCAF320.html>