

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE SISTEMAS INFORMÁTICOS



**PROTOTIPO DE DESPLIEGUE AUTOMATIZADO DE REGIONES PARA SU
INTEGRACIÓN A UNA INFRAESTRUCTURA CENTRALIZADA CON UN CASO
DE USO EN UNA ESCUELA DE UNA UNIVERSIDAD**

PRESENTADO POR:

CASTRO SÁNCHEZ JOSÉ ANDRÉS

MÁRQUEZ VENTURA JONATHAN ENRIQUE

MEDINA HUEZO GERARDO ANTONIO

MEJÍA ALVARADO JORGE OVIDIO

PARA OPTAR AL TÍTULO DE:

INGENIERO DE SISTEMAS INFORMÁTICOS

CIUDAD UNIVERSITARIA, NOVIEMBRE DE 2023

RECTOR:

MSc. JUAN ROSA QUINTANILLA

SECRETARIO GENERAL:

LIC. PEDRO ROSALÍO ESCOBAR CASTANEDA

FACULTAD INGENIERÍA Y ARQUITECTURA

DECANO:

ING. LUIS SALVADOR BARRERA MANCÍA

SECRETARIO:

ARQ. RAUL ALEXANDER FABIÁN ORELLANA

ESCUELA DE INGENIERÍA DE SISTEMAS INFORMATICOS

DIRECTOR:

ING. CÉSAR AUGUSTO GONZÁLEZ RODRÍGUEZ

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE INGENIERÍA DE SISTEMAS INFORMÁTICOS

Trabajo de Graduación previo a la opción al Grado de:

INGENIERO DE SISTEMAS INFORMÁTICOS

Título:

**PROTOTIPO DE DESPLIEGUE AUTOMATIZADO DE REGIONES PARA SU
INTEGRACIÓN A UNA INFRAESTRUCTURA CENTRALIZADA CON UN CASO
DE USO EN UNA ESCUELA DE UNA UNIVERSIDAD**

Presentado por:

CASTRO SÁNCHEZ JOSÉ ANDRÉS
MÁRQUEZ VENTURA JONATHAN ENRIQUE
MEDINA HUEZO GERARDO ANTONIO
MEJÍA ALVARADO JORGE OVIDIO

Trabajo de Graduación Aprobado por:

Docente Asesor:

MSc. DAMIÁN MORALES

SAN SALVADOR, NOVIEMBRE 2023

Trabajo de Graduación Aprobado por:

Docente Asesor:

MSc. Damián Morales

Índice de contenido

1.	Introducción	1
2.	Descripción del proyecto	2
3.	Concepción del proyecto	3
3.1.	Antecedentes	3
3.2.	Justificación	5
3.3.	Pregunta de investigación	6
3.4.	Objetivos	7
3.4.1.	Objetivo General	7
3.4.2.	Objetivos Específicos	7
3.5.	Limitaciones	8
4.	Marco teórico	9
4.1.	Virtualización	9
4.1.1.	Introducción de los hipervisores	10
4.1.1.1.	Hipervisor tipo 1	11
4.1.1.2.	Hipervisor tipo 2	13
4.2.	Cloud Computing	14
4.2.1.	Nube	15
4.2.2.	Tipos de nubes	15
4.2.2.1.	Nube Pública	16
4.2.2.2.	Nube Privada	17
4.2.2.3.	Nube Híbrida	18
4.2.3.	Tipos especializados de nube	20
4.2.3.1.	Nube comunitaria	20
4.2.4.	Gestión y Seguridad de la Infraestructura en la Nube	22
4.2.4.1.	Región	22
4.2.4.2.	Zona de disponibilidad	22
4.2.4.3.	Dominio	23
4.2.4.4.	Proyecto	23
4.2.4.5.	Rol	25
4.2.4.6.	Usuarios	25
4.2.4.7.	RBAC	26
4.3.	Soluciones de despliegue en la nube	27

4.3.1.	Servicios esenciales en la nube.....	29
4.3.1.1.	Servicio de cómputo.....	30
4.3.1.2.	Servicio de red.....	31
4.3.1.3.	Servicio de identidad.	33
4.3.1.4.	Servicio de imágenes.....	34
4.3.1.5.	Servicio de panel de control (Dashboard).	36
4.3.1.6.	Servicio de gestión de claves.	37
4.4.	Clúster	38
4.4.1.	Cluster de base de datos.	39
4.4.1.1.	Mariadb - Galera.....	39
4.4.2.	Cluster de almacenamiento.....	42
4.4.2.1.	Almacenamiento en bloques.	42
4.4.2.2.	Almacenamiento de objetos.	43
4.4.2.3.	Soluciones de almacenamiento distribuido.	44
4.4.2.3.1.	<i>Glusterfs</i>	44
4.4.2.3.2.	<i>Cephfs</i>	45
4.5.	Automatización.....	48
4.5.1.	Terraform.....	50
4.5.2.	SaltStack.	51
4.5.3.	Ansible.....	52
5.	Metodología de investigación.....	55
5.1.	Metodología de la investigación	55
5.1.1.	Enfoque cualitativo utilizado.	55
5.2.	Población y Muestra.....	56
5.3.	Técnica de recopilación de datos utilizada.....	57
5.3.1.	Entrevista estructurada.....	57
5.4.	Análisis y Resultados.....	58
6.	Factibilidad.....	59
6.1.	Factibilidad técnica	59
6.1.1.	Comparación de soluciones de nube.	60
6.1.2.	Soluciones de almacenamiento.....	62
6.1.2.1.	MinIo.....	62
6.1.2.2.	HekaFS (LizardFS).....	62
6.1.2.3.	MooseFs.	63

6.1.3.	Comparativa de soluciones de almacenamiento.....	63
6.2.	Factibilidad económica	65
7.	Prototipo.....	70
7.1.	Construcción del prototipo.....	71
7.1.1.	Tecnologías utilizadas.....	71
7.1.2.	Diseño del prototipo.....	71
7.1.3.	Desarrollo del prototipo.	72
7.1.3.1.	Infraestructura.....	72
7.1.3.2.	Región.	73
7.1.3.3.	Redes empleadas.	74
7.2.	Prueba del prototipo	76
7.2.1.	Funcionamiento de OpenStack.....	76
7.2.2.	Funcionamiento de Ceph.....	81
8.	Caso de uso	86
8.1.	Implementación de Regiones para una Infraestructura Centralizada de nube comunitaria para la Escuela de Ingeniería de Sistemas Informáticos	86
9.	Conclusiones.....	91
10.	Recomendaciones	92
11.	Referencias Bibliográficas.....	94
12.	Anexo.....	100

1. Introducción

La computación en la nube ha ocasionado un cambio profundo en la manera en que las entidades organizativas diseñan, ejecutan y supervisan sus aplicaciones y servicios. Su implementación abarca un espectro que va desde Software como Servicio (SaaS) hasta la Infraestructura como Servicio (IaaS), ha otorgado a las compañías una mayor efectividad, flexibilidad y capacidad de escalabilidad en sus operaciones de Tecnologías de la Información. Además, esta transformación tecnológica ha brindado una alternativa para la adopción de esquemas innovadores de negocios digitales, lo que ha conducido a una competencia más intensa entre diversos proveedores. Esta competencia, a su vez, ha enfatizado la necesidad de aprovechar al máximo las ventajas ofrecidas por dichos proveedores en beneficio de los usuarios.

El objetivo esencial de este proyecto se centra en la creación de un despliegue automatizado de una entidad de nube denominada "Región", que se configurará de tal manera que se integre sin mayor complejidad a un nodo de infraestructura central ya existente, al que le cederá la gestión de servicios esenciales, tales como almacenamiento, conectividad de red y capacidad de cómputo. Esta región asumirá un rol crítico en la infraestructura global, aportando significativamente a la disponibilidad y eficiencia de los recursos en beneficio de todos los usuarios que la utilicen.

La tarea de implementar esta región representa un desafío importante para la computación en la nube, puesto que se abordan aspectos cruciales de la administración de dichos entornos, como lo son la escalabilidad, disponibilidad y segmentación de los servicios que la conforman. A lo largo de este reporte, se detallarán los aspectos técnicos sobre cómo llevar a cabo este proyecto, desde la instauración de la nube, su configuración y su interacción posterior con la interfaz de programación de aplicaciones (API) de la infraestructura subyacente.

2. Descripción del proyecto

El proyecto propuesto presenta como tarea el desarrollo de un despliegue automatizado de una nube, configurada como una región, que pueda ser perfectamente integrada en una infraestructura centralizada. Sin embargo, para asegurar su viabilidad y eficacia, también se llevará a cabo un despliegue de tipo infraestructura con los componentes mínimos necesarios, que permitirá poner a prueba el proceso de integración y todas las configuraciones correspondientes.

Esta nube contendrá un conjunto de servicios críticos que proveerán las funcionalidades esenciales para su pleno funcionamiento. Es importante destacar que los servicios de almacenamiento de la nube se integrarán a una solución de almacenamiento distribuida, lo que garantizará una gestión eficiente de los datos y la posibilidad de visualizarlos a través de un panel gráfico.

Un aspecto relevante de este proyecto es su aplicación práctica. Se plantea un caso de uso específico en el contexto de la Escuela de Ingeniería de Sistemas Informáticos de la Facultad de Ingeniería y Arquitectura de la Universidad de El Salvador. En este caso, la Escuela asumirá el papel de infraestructura para la nube , mientras que el rol de las regiones puede ser tomado por los cuatro departamentos que la componen. Esto demuestra cómo el concepto de regiones en una nube puede ser aplicado de manera efectiva en un entorno educativo, facilitando la gestión y distribución de recursos de manera más eficiente y escalable.

3. Concepción del proyecto

3.1. Antecedentes

La evolución tecnológica en el ámbito de la informática ha transformado la manera en que las organizaciones gestionan sus recursos y servicios. Anteriormente, las universidades, ministerios y empresas de diferentes tamaños enfrentaban una variedad de desafíos en la gestión de sus infraestructuras y servicios, lo que limitaba su capacidad para innovar y responder de forma eficiente a las demandas del mercado y la sociedad. Antes del surgimiento de soluciones de la nube, estas organizaciones a menudo enfrentaban los siguientes problemas:

1. Las universidades, ministerios, pequeñas y grandes empresas tenían sus propios centros de datos para gestionar sus recursos informáticos y servicios en sus instalaciones de forma física.
2. Cada organización tenía que invertir en hardware, software y personal para mantener sus infraestructuras de TI.
3. El mantenimiento y la administración de la infraestructura eran costosos y requerían de mucho esfuerzo.
4. La colaboración y el intercambio de recursos entre organizaciones eran complicados debido a que la infraestructura en muchos casos estaba fragmentada por ejemplo entre los distintos departamentos dentro de la organización y no se tenía de forma centralizada.

La aparición de soluciones en la nube marcó un antes y un después en la forma en que estas organizaciones abordaban sus necesidades tecnológicas. Por medio de la creación de una "Nube Comunitaria" y la implementación de "Regiones", se abre la puerta a solucionar estos problemas de manera integral ya que brinda las siguientes soluciones:

1. Se establece una infraestructura unificada basada en soluciones en la nube que permite la creación de múltiples regiones interconectadas.
2. Cada organización puede aprovechar recursos compartidos, almacenamiento y red a través de la nube comunitaria.
3. Las universidades pueden colaborar de forma más eficiente en proyectos de investigación y educación al compartir recursos de manera más sencilla.
4. Los ministerios encontraron una forma más eficiente de gestionar sus aplicaciones y servicios gubernamentales, optimizando costos.
5. Las medianas y grandes empresas han podido escalar según sus necesidades comerciales y desplegar servicios de manera más ágil.
6. La administración centralizada y la automatización redujeron la carga de trabajo del personal de TI y los costos de mantenimiento.
7. Las organizaciones pudieron consumir servicios a través de un API, lo que facilitó la integración y la implementación.

Las soluciones en la nube brindan a las universidades, ministerios y empresas, tanto medianas como grandes, una solución integral y escalable que ha venido a transformar la forma en que estas operan y aprovechan la tecnología. El proyecto busca aprovechar estos beneficios al crear una Nube Comunitaria con Regiones que proporcionen servicios de manera eficiente y colaborativa, que les permita a las organizaciones aprovechar al máximo la tecnología para lograr sus objetivos.

3.2. Justificación

La implementación de soluciones escalables, flexibles y altamente disponibles representa un desafío significativo para muchas organizaciones. La infraestructura propuesta busca abordar este desafío mediante la creación de una región que se integre de manera transparente con una nube centralizada. Esto permitirá a las empresas expandir sus servicios y aplicaciones de manera efectiva, evitando los costos operativos y de capital elevados asociados con la adquisición, mantenimiento y escalabilidad de infraestructuras tradicionales.

Estos costos incluyen la inversión inicial en hardware, la necesidad de personal especializado para la administración y el mantenimiento continuo, así como la dificultad para adaptarse rápidamente a cambios en la demanda. La inversión inicial en hardware abarca desde servidores hasta equipos de red, generando una carga financiera considerable. La contratación de personal especializado para la administración y el mantenimiento continuo añade costos operativos a largo plazo. Además, la complejidad asociada con la escalabilidad en infraestructuras tradicionales puede resultar en dificultades para ajustarse eficientemente a las fluctuaciones de la demanda del mercado. La nueva infraestructura propuesta busca ofrecer una alternativa más rentable al eliminar la necesidad de grandes inversiones iniciales en hardware y simplificar la gestión operativa. Al hacerlo, las empresas pueden reducir los costos a largo plazo, concentrándose en la expansión eficiente de sus servicios. Además, al garantizar la redundancia y la continuidad operativa en caso de fallos, la propuesta busca mitigar riesgos y asegurar la estabilidad en el rendimiento de las operaciones empresariales.

3.3. Pregunta de investigación

¿Es factible llevar a cabo la implementación automatizada de una nube configurada como una región a forma de prototipo, que pueda integrarse con éxito en una infraestructura centralizada, incluyendo servicios de gestión de recursos, cómputo, redes, imágenes y almacenamiento, y que ofrezca dichos servicios de manera eficiente a todos los usuarios a través de una interfaz web centralizada proporcionada por dicha infraestructura?

3.4. Objetivos

3.4.1. *Objetivo General.*

Diseñar, implementar y validar un prototipo de despliegue automatizado de una nube, enfocada en la configuración de una región, con el propósito de lograr su integración eficiente en una infraestructura centralizada, facilitando los puntos de acceso de los servicios desplegados para su respectiva interacción mediante una interfaz web provista por dicha infraestructura.

3.4.2. *Objetivos Específicos.*

- Configurar los nodos necesarios para el despliegue de una nube configurada como una región, considerando los recursos computacionales, de red y de almacenamiento.
- Implementar los servicios esenciales de la nube, abarcando cómputo, redes, almacenamiento en bloques, almacenamiento de objetos y asignación de recursos, asegurando su interacción y funcionamiento sin conflictos dentro de la misma región y la infraestructura centralizada.
- Integrar la región en la infraestructura centralizada, configurando los componentes respectivos para lograr una comunicación eficiente entre ambas entidades.
- Configurar la región de tal manera que se garantice la escalabilidad horizontal y vertical, permitiendo el aumento eficiente de recursos según las demandas de los servicios.
- Documentar de manera exhaustiva el proceso de diseño, configuración y despliegue, así como los desafíos enfrentados y las soluciones implementadas, para futuras referencias y replicaciones.

3.5. Limitaciones

- La capacidad del equipo de cómputo actual de los miembros del grupo no cumple con los requisitos de hardware necesarios para llevar a cabo el prototipo sin enfrentar obstáculos en cuanto a capacidad de almacenamiento, ancho de banda y potencia de cómputo.
- La implementación y gestión eficaz del prototipo puesto en producción requerirán personal capacitado en OpenStack.
- La implementación exitosa del prototipo en la escuela de sistemas depende críticamente de una conexión a internet robusta con un ancho de banda adecuado debido a que la infraestructura diseñada requiere una transmisión eficiente de datos entre las regiones y nodos, lo que podría afectar negativamente el rendimiento y la experiencia del usuario en caso de limitaciones en la conexión.

4. Marco teórico

4.1. Virtualización

La virtualización es una tecnología que se puede usar para crear representaciones virtuales de servidores, almacenamiento, redes y otras máquinas físicas. El software virtual imita las funciones del hardware físico para ejecutar varias máquinas virtuales a la vez en una única máquina física. Las empresas recurren a la virtualización para utilizar sus recursos de hardware de manera eficiente y obtener retornos mayores de sus inversiones. También potencia los servicios de computación en la nube que ayudan a las organizaciones a administrar la infraestructura de manera más eficaz (RedHat, 2023).

La virtualización proporciona varios beneficios a cualquier organización:

- Utilización eficiente de los recursos.
- Administración automatizada de las TI.
- Recuperación de desastres más rápida.

Existen diferentes tipos de virtualización en los cuales permite obtener las funciones de distintos tipos de infraestructura física y todas las ventajas de un entorno virtualizado. (AWS, s.f.) explica algunos de ellos :

- **Virtualización de servidores:** es un proceso que particiona un servidor físico en múltiples servidores virtuales.
- **Virtualización de almacenamiento:** combina las funciones de los dispositivos de almacenamiento físico, como el almacenamiento conectado a la red (NAS) y la red de área de almacenamiento (SAN).
- **Virtualización de red:** es un proceso que combina todos estos recursos de red para centralizar las tareas administrativas.

- **Virtualización de datos:** crea una capa de software entre estos datos y las aplicaciones que los necesitan.
- **Virtualización de aplicaciones:** extrae las funciones de las aplicaciones de modo que se ejecuten en sistemas operativos distintos de aquellos para los que fueron diseñadas.
- **Virtualización de escritorios:** permite administrar los escritorios de forma eficiente y segura, con lo que se ahorra dinero en hardware de escritorio.

4.1.1. Introducción de los hipervisores.

La virtualización utiliza un software especializado, llamado hipervisor, para crear varias instancias en la nube o máquinas virtuales en un solo equipo físico (AWS, s.f.).

Hipervisores: El hipervisor es el software de virtualización que se instala en la máquina física. Es una capa de software que actúa como intermediario entre las máquinas virtuales y el hardware subyacente o el sistema operativo del host. El hipervisor coordina el acceso al entorno físico de manera que varias máquinas virtuales tengan acceso a su propia cuota de recursos físicos (AWS, s.f.).

Instancias en la nube o máquinas virtuales: Después de instalar el software de virtualización en la computadora, podrá crear una o más máquinas virtuales. Se puede acceder a las máquinas virtuales de la misma manera que se accede a otras aplicaciones en la computadora. La computadora se llama host y la máquina virtual se llama huésped. Varios huéspedes se pueden ejecutar en el host. Cada huésped tiene su propio sistema operativo, que puede ser el mismo o diferente del sistema operativo del host (AWS, s.f.).

Según Gerald J. Popek y Robert P. Goldberg (1974), afirman que hay dos tipos de hipervisores. Aunque la distinción entre ambos tipos no siempre es completamente clara.

4.1.1.1. Hipervisor tipo 1.

Los hipervisores de tipo 1 o bare-metal, también conocidos como hipervisores nativos, se ejecutan directamente en el servidor. Los hipervisores bare-metal gestionan el sistema o sistemas operativos invitados. Según (Stackscale, s. f.) algunos ejemplos de hipervisores bare-metal de código abierto y comerciales son:

- **Hipervisores bare-metal de código abierto:** KVM, Proxmox y Xen.
- **Hipervisores bare-metal comerciales:** Red Hat Enterprise Virtualization (RHEV), Citrix XenServer, Hyper-V y VMware ESXi.

Este tipo de hipervisores es el que se suele desplegar para necesidades de computación de centros de datos. Por lo general, este tipo de hipervisores ofrecen un rendimiento mejor y más eficiente que los hipervisores alojados. (Stackscale, s. f.) menciona un listado de algunos hipervisores tipo 1:

1. **VMware ESXi:** es un hipervisor bare-metal de nivel empresarial que forma parte de la suite de virtualización vSphere de VMware. Este cuenta con su propio kernel y otros componentes esenciales del sistema operativo.
2. **OpenStack:** es una plataforma cloud de código abierto y gratuita, que se despliega tanto en nubes públicas como privadas. Fue desarrollado inicialmente por Rackspace Hosting y la NASA en 2010.
3. **LXC:** abreviatura de Linux Containers, es una plataforma de virtualización a nivel del sistema operativo, de código abierto, diseñada para desarrollar tecnologías de contenedores para Linux.
4. **Proxmox VE:** abreviatura de Proxmox Virtual Environment, es una plataforma de virtualización de código abierto basada en Debian. Soporta tanto la virtualización basada en contenedores con LXC y virtualización completa con KVM.

5. **RHEV:** abreviatura de Red Hat Enterprise Virtualization, es una plataforma de virtualización de nivel empresarial basada en KVM.
6. **Hyper-V:** también llamado Microsoft Hyper-V, es un hipervisor bare-metal, desarrollado por Microsoft.
7. **XEN:** es un hipervisor bare-metal desarrollado inicialmente por el departamento de tecnología y ciencias de la computación de la Universidad de Cambridge, liderado por el académico senior Ian Pratt y su estudiante de doctorado Keir Fraser, en 2003 (Stackscale, s. f.).
8. **KVM:** son una tecnología de virtualización open source integrada a Linux. Con ellas, se puede transformar Linux en un hipervisor que permite que una máquina host ejecute varios entornos virtuales aislados llamados máquinas virtuales (VM) o guests (RedHat, s. f.).

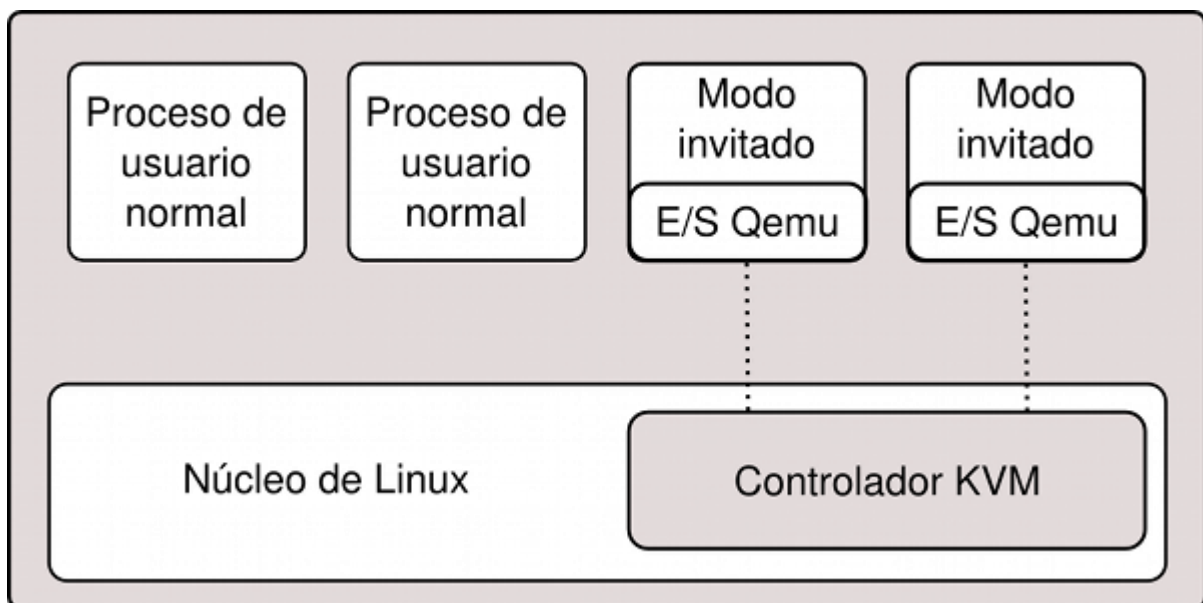


Figura 1: Talens-Oliag, Sergio. (2010). *Arquitectura de KVM* [Imagen]. Herramientas de virtualización libres para sistemas GNU/Linux. https://www.uv.es/sto/charlas/2010_CIM/hv1-cim-2010.html/index.html

En la Figura 1 se muestra la estructura de KVM, en este modelo las máquinas virtuales son procesos normales del sistema (por esto la gestión de memoria y la planificación de

procesos son el estándar del sistema) a los que se añade un modo de ejecución adicional (invitado) a parte de los modos de ejecución estándar de Linux (usuario y núcleo). (ITI, 2010) menciona que una máquina virtual tendrá tres modos de ejecución:

- **Modo invitado:** será el modo de ejecución normal para el código del sistema invitado siempre que no tenga operaciones de entrada/salida.
- **Modo usuario:** sólo se usa para ejecutar las operaciones de entrada/salida del sistema invitado, permite gestionar dispositivos virtuales a nivel de usuario.
- **Modo núcleo:** se usa para entrar a trabajar en modo invitado y para gestionar las salidas desde modo usuario causadas por operaciones especiales o de entrada/salida.

En cuanto a la implementación del sistema, el KVM está formado por dos componentes:

- Un controlador de dispositivos (Controlador KVM) para gestionar el hardware de virtualización, accesible desde el dispositivo `/dev/kvm` (incluido en el núcleo de Linux desde la versión 2.6.20, con soporte para microprocesadores Intel y AMD).
- Un programa de usuario (Proceso de usuario normal) que emula el hardware del PC (actualmente se usa una versión modificada de `qemu`) que se encarga de reservar la memoria de la máquina virtual y llamamiento al controlador anterior para ejecutar código en modo invitado (ITI, 2010).

4.1.1.2. Hipervisor tipo 2.

Los hipervisores de tipo 2 o hipervisores alojados, del inglés *hosted hypervisors*, se ejecuta como una capa de software por encima del sistema operativo de la *host machine*. Se usan para abstraer los sistemas operativos invitados del sistema operativo principal. Algunos ejemplos de hipervisores alojados de código abierto y comerciales son (Stackscale, s. f.):

- **Hipervisores alojados de código abierto:** QEMU y VirtualBox.

- **Hipervisores alojados comerciales:** Parallels Desktop, VMware Workstation Player y VMware Fusion.

Listado de algunos hipervisores tipo 2:

1. **VMware Workstation Player:** es un hipervisor alojado para ejecutar máquinas virtuales en Microsoft Windows o Linux PC. VMware Player y VMware Workstation fueron integrados dentro de un mismo paquete en 2015.
2. **VMware Fusion:** es un hipervisor de escritorio para ejecutar máquinas virtuales en Mac OS.
3. **VirtualBox:** conocido anteriormente como Innotek VirtualBox, Sun VirtualBox y Sun xVM VirtualBox, es un hipervisor alojado de código abierto. Inicialmente fue creado por Innotek, que en 2008 fue adquirido por Sun Microsystems (que Oracle compró a su vez en 2010).
4. **Parallels Desktop:** es un hipervisor alojado que permite ejecutar Windows en Mac sin necesidad de reiniciar.
5. **QEMU:** es un hipervisor alojado gratuito y de código abierto, escrito originalmente por Fabrice Bellard y disponible con una licencia pública general de GNU. Actualmente su desarrollo y mantenimiento lo lleva a cabo el equipo de QEMU. QEMU se puede usar como emulador y como virtualizador (Stackscale, s. f.).

4.2. Cloud Computing

Cloud computing es la disponibilidad bajo demanda de recursos de computación como servicios a través de Internet. La computación de la nube evita que las empresas tengan que encargarse de aprovisionar, configurar o gestionar los recursos y permite que paguen únicamente por los que usen (Google, s. f.).

4.2.1. Nube.

Los servicios de nube son infraestructuras, plataformas o sistemas de software que los proveedores alojan y ponen a disposición de los usuarios a través de una red, como Internet. Dichos servicios facilitan el flujo de datos de los usuarios hasta los sistemas de los proveedores, y viceversa. Además, fomentan el diseño de aplicaciones en la nube y la flexibilidad de trabajar en ella. Para acceder a estos servicios, los usuarios solo necesitan una computadora, un sistema operativo y conexión de red (Red Hat, 2023).

La nube permite a los usuarios acceder a los mismos archivos y aplicaciones casi desde cualquier dispositivo, ya que los procesos informáticos y de almacenamiento tienen lugar en servidores en un centro de datos, y no de forma local en el dispositivo del usuario. Por ejemplo, un usuario de redes sociales puede iniciar sesión en su cuenta de Instagram con un teléfono nuevo después de que se le haya averiado el anterior y seguirá teniendo acceso a su cuenta de siempre, con sus fotos, vídeos y el historial de conversación. Funciona igual con proveedores de correo electrónico en la nube como Gmail o Microsoft Office 365, y con proveedores de almacenamiento en la nube como Dropbox o Google Drive (Cloudflare. s.f.).

4.2.2. Tipos de nubes.

Los tipos de nube en informática representan una evolución fundamental en la forma en que las organizaciones abordan la gestión de datos y servicios informáticos. La elección de un modelo de nube adecuado depende de una serie de factores, incluidas las necesidades específicas de la organización, los requisitos de seguridad y cumplimiento, así como las metas de eficiencia y escalabilidad. Con una comprensión adecuada de los diferentes tipos de nube y sus aplicaciones, las organizaciones pueden aprovechar al máximo los beneficios que la nube nos ofrece (VMware, 2022).

4.2.2.1. Nube Pública.

Una nube pública es un modelo de TI cuyos servicios informáticos e infraestructura a la carta los gestiona un proveedor externo y se comparten con varias organizaciones a través de la red pública de Internet. Los proveedores de nube pública pueden prestar servicios de nube de tipo infraestructura como servicio (IaaS), plataforma como servicio (PaaS) o software como servicio (SaaS) a los usuarios por una tarifa mensual o de pago por uso. Esto elimina la necesidad de que los usuarios alojen estos servicios de forma local en su propio centro de datos (VMware, 2022).

Los proveedores de servicios de nube utilizan grupos de centros de datos que están divididos en máquinas virtuales y compartidos por los clientes. Los clientes pueden sencillamente alquilar el uso de las máquinas virtuales o pagar por servicios adicionales basados en la nube, como aplicaciones de software, herramientas de desarrollo de aplicaciones o almacenamiento. Las empresas suelen utilizar los servicios de nube pública para aplicaciones menos confidenciales que tienen picos de uso impredecibles o para almacenar datos que no requieren un acceso frecuente (VMware, 2022).

(Red Hat, 2023) hace mención de las siguientes características para una nube pública:

1. **Asignación de los recursos:** los usuarios externos al firewall del proveedor comparten los recursos virtuales y los servicios de nube del conjunto de infraestructuras, plataformas y sistemas de software del proveedor.
2. **Acuerdos de uso:** los recursos se distribuyen en función de las necesidades, pero los modelos de pago según el consumo no son imprescindibles. Algunos clientes usan las nubes públicas sin costo, como las instituciones de investigación que utilizan Massachusetts Open Cloud.
3. **Gestión:** como mínimo, el proveedor se encarga del mantenimiento del hardware subyacente a la nube, brinda soporte para la red y gestiona el software de virtualización.

Proveedores de nubes públicas

Existe variedad en cuanto a los proveedores de nubes públicas. Entre los de mayor cuota de mercado se encuentran:

- Google Cloud
- Amazon Web Services
- Microsoft Azure
- Oracle Cloud Infrastructure
- IBM Cloud

4.2.2.2. Nube Privada.

La nube privada se define como los servicios informáticos que se ofrecen a través de Internet o de una red interna privada solo a algunos usuarios y no al público general. También denominada nube interna o corporativa, la informática en nube privada aporta a las empresas gran parte de las ventajas de la nube pública (como autoservicio, escalabilidad y elasticidad), pero con el control y la personalización disponibles en los recursos dedicados a través de una infraestructura informática hospedada en el entorno local. Además, las nubes privadas ofrecen un nivel más alto de seguridad y privacidad con firewalls de la compañía y hospedaje interno, con el fin de garantizar que las operaciones y los datos confidenciales no estén accesibles para proveedores externos. Un inconveniente es que el departamento de TI de la compañía es responsable de la administración de la nube privada y el coste que conlleva (Microsoft Azure, s. f.).

En una nube privada, al igual que en una nube pública o de otro tipo, se pueden ofrecer dos modelos de servicios en la nube. El primero es infraestructura como servicio (IaaS), que permite a una compañía utilizar recursos de infraestructura, como proceso, red y almacenamiento, como servicio. El segundo es plataforma como servicio (PaaS), que permite a una compañía ofrecer todo, desde aplicaciones basadas en la nube sencillas hasta aplicaciones

empresariales sofisticadas. Las nubes privadas se pueden combinar también con nubes públicas para crear una nube híbrida, que permite a las empresas aprovechar la extensión a la nube para liberar más espacio y escalar servicios informáticos en la nube pública cuando aumenta la demanda de recursos informáticos (Microsoft Azure, s. f.).

Comparación entre la nube privada y la nube pública

La diferencia principal entre la nube pública y la nube privada es donde se aloja dicha nube y quién es responsable de la administración de ella. La nube pública usa una infraestructura compartida, mientras que las nubes privadas usan la infraestructura dedicada de una organización. Las plataformas de nube pública, agrupan recursos en centros de datos distribuidos en todo el mundo a los que pueden acceder varias empresas y usuarios desde Internet. En lugar de un equipo interno, los proveedores de la nube pública son responsables de administrar y mantener la infraestructura subyacente. Como resultado, aprovechar los servicios de nube pública reduce los costos operativos de TI y libera tiempo para que los equipos se enfoquen en trabajo valioso que beneficia directamente a la empresa (Google Cloud, s. f.).

4.2.2.3. Nube Híbrida.

La nube híbrida es un diseño de infraestructura de TI que integra los recursos de TI internos de una empresa con la infraestructura y los servicios de proveedores de nube de terceros. Con la nube híbrida, puede almacenar sus datos y ejecutar sus aplicaciones en varios entornos, ayudando así a aprovisionar, escalar y administrar de forma centralizada los recursos informáticos con los que se disponga.

(Amazon Web Services, s.f.) proporciona más detalles acerca de las ventajas de utilizar una infraestructura de nube híbrida:

1. **Mayor agilidad en el desarrollo:** Un enfoque de nube híbrida permite acelerar el lanzamiento de nuevos productos al mercado. Esto se logra al personalizar la infraestructura para experimentar con nuevas ideas de manera más segura y eficiente.
2. **Mayor capacidad de adaptación:** A medida que una empresa crece, aumenta la demanda de recursos informáticos, como memoria, capacidad de procesamiento y ancho de banda de la red, conocidos como cargas de trabajo. La nube híbrida brinda la flexibilidad de mover estas cargas de trabajo entre diferentes entornos, permitiendo una escalabilidad eficiente y rentable.
3. **Continuidad empresarial:** La continuidad empresarial se refiere a la capacidad de mantener las operaciones sin interrupciones, incluso en situaciones de fallos, mantenimiento u otros cambios. Los modelos de nube híbrida reducen los tiempos de inactividad y mejoran la experiencia del usuario. Por ejemplo, si la infraestructura de nube privada está en proceso de actualización, es posible trasladar sus cargas de trabajo a entornos de nube pública para garantizar la continuidad de las operaciones de la organización.

¿Nube pública, nube privada o nube híbrida?

Una empresa que requiere servicios informáticos basados en la nube tiene la opción de utilizar una nube pública (donde los servicios son proporcionados por un proveedor externo y compartidos con otros usuarios), una nube privada (donde la empresa aloja sus propios servicios en la nube) o una combinación de ambas, llamada "nube híbrida". La nube pública ofrece la ventaja de una fácil escalabilidad, mientras que una nube privada, aunque inicialmente más económica después de invertir en infraestructura, carece de esta facilidad de escalabilidad, ya que la expansión puede requerir la adquisición de hardware adicional. (VMware, 2022) afirma que esto puede resultar en la infrutilización de recursos costosos si el uso de la nube privada disminuye.

La opción de nube híbrida puede ser un compromiso beneficioso para quienes consideran la nube pública frente a la nube privada. La nube híbrida implica cualquier combinación de soluciones de nube pública y privada. Mediante el entorno de nube híbrida, las organizaciones pueden aprovechar las ventajas de ambos tipos de plataformas de nube y elegir la más adecuada según sus necesidades de datos específicas. Por ejemplo, la nube híbrida también ofrece una alternativa para el almacenamiento de datos confidenciales, permitiendo a las empresas ofrecer servicios a través de una nube pública mientras protegen la seguridad de los datos sensibles en una nube privada (Microsoft Azure, s.f.).

4.2.3. Tipos especializados de nube.

4.2.3.1. Nube comunitaria.

Una nube comunitaria es una variante modificada de una nube privada. Esta plataforma de tecnología virtual multiusuario permite que distintas organizaciones de una comunidad, usualmente con objetivos metas afines o que solucionan una necesidad específica, puedan trabajar de manera colaborativa en una misma plataforma con infraestructuras y recursos compartidos. Estas nubes comunitarias se alojan en proveedores de servicios en la nube de terceros. El alojamiento se encuentra en un centro de datos que puede ser local o externo. A diferencia de otras nubes, como las privadas, públicas e híbridas, las nubes comunitarias son un modelo relativamente novedoso. Durante la pandemia, debido a la necesidad de trabajar en línea, las nubes comunitarias han ganado popularidad. (Capterra, s.f.) menciona algunas ventajas y desventajas de una nube comunitaria:

Ventajas de una nube comunitaria:

1. **Costos compartidos:** Al compartir recursos de infraestructura y servicios en la nube, los participantes pueden reducir sus costos operativos, ya que se comparten gastos de mantenimiento y gestión.

2. **Personalización:** Las nubes comunitarias a menudo permiten una mayor personalización de los servicios para satisfacer las necesidades específicas de los miembros de la comunidad, lo que puede conducir a soluciones más adecuadas.
3. **Mayor seguridad:** Al compartir una infraestructura común, las organizaciones pueden colaborar en la implementación de medidas de seguridad, lo que puede llevar a una mejor protección de los datos y sistemas.
4. **Cumplimiento normativo:** En algunos casos, las nubes comunitarias pueden facilitar el cumplimiento normativo compartido, ya que las organizaciones pueden trabajar juntas para garantizar que sus operaciones cumplan con los requisitos legales y regulatorios específicos de su industria.
5. **Escalabilidad:** Al compartir recursos, las organizaciones pueden aprovechar la capacidad de escalabilidad de la nube para satisfacer las demandas cambiantes sin incurrir en costos excesivos.

Desventajas de una nube comunitaria:

1. **Complejidad de gestión:** La gestión compartida de recursos puede ser compleja, ya que las organizaciones deben coordinar políticas, configuraciones y requisitos de seguridad, lo que puede dar lugar a desafíos operativos.
2. **Dependencia de otros miembros:** Si un miembro de la comunidad experimenta problemas o incumple sus responsabilidades, esto puede afectar a otros usuarios de la nube comunitaria.
3. **Conflictos de intereses:** Pueden surgir conflictos entre los miembros de la comunidad debido a diferencias en las necesidades y objetivos de cada organización.
4. **Limitaciones en la personalización:** Aunque las nubes comunitarias ofrecen cierta flexibilidad en la personalización, es posible que no sean tan adaptables como las soluciones de nube privada, lo que puede limitar las opciones de configuración.

5. **Seguridad compartida:** Aunque se mencionó anteriormente como una ventaja, la seguridad también puede ser una desventaja si no se gestiona adecuadamente, ya que un fallo en la seguridad de un miembro puede poner en riesgo a toda la comunidad.

4.2.4. Gestión y Seguridad de la Infraestructura en la Nube.

4.2.4.1. Región.

Una región es un conjunto de zonas de disponibilidad en un área geográfica designada. Cada zona de disponibilidad dentro de una región está conectada de forma independiente a todas las demás zonas de disponibilidad dentro de la misma región mediante enlaces de fibra óptica privados redundantes y de baja latencia, lo que garantiza una comunicación y replicación confiables y de alta velocidad. Estas interconexiones independientes evitan que cualquier zona de disponibilidad se convierta en un único punto de error para la comunicación entre otras zonas de disponibilidad (Dgtl Infra, 2023).

Todas las regiones están aisladas para la tolerancia a fallos, pero interconectadas entre sí e Internet a través de redes de fibra óptica de alta velocidad. Se recomienda seleccionar una región y zonas de disponibilidad más cercanas a las operaciones comerciales para lograr el mayor rendimiento posible. Sin embargo, no todas las regiones o zonas de disponibilidad son iguales, y es posible que algunos servicios solo estén disponibles en regiones o zonas de disponibilidad específicas (Dgtl Infra, 2023).

4.2.4.2. Zona de disponibilidad.

Según (Educative, s. f.) una zona de disponibilidad o AZ es una ubicación aislada dentro de una región geográfica que sirve como centro de datos físico. Estas ubicaciones son donde residen los centros de datos y los servicios de los proveedores de servicios en la nube se originan y operan. Las zonas de disponibilidad sirven para varios propósitos, tales como:

- Conformidad
- Proximidad al cliente
- Disminución de la latencia
- Protección de recursos, etc.

4.2.4.3. Dominio.

Los dominios son entornos o segmentos de recursos y servicios que se organizan de manera lógica o virtual en una infraestructura de nube. Estos dominios pueden estar relacionados con la gestión de recursos, la seguridad, la administración de identidades y accesos, etc. Por ejemplo, en un entorno de nube empresarial, puede haber dominios que representan diferentes departamentos o equipos, y se pueden configurar políticas y permisos específicos para cada dominio (Tecnozero 2023).

4.2.4.4. Proyecto.

Según (Aitor Medrano, 2022) un proyecto se refiere a un conjunto de recursos y configuraciones específicas que se crean y gestionan dentro de ese servicio para llevar a cabo una iniciativa o una serie de tareas relacionadas. (Aitor Medrano, 2022) considera lo siguiente:

Organización y aislamiento: Un proyecto en la nube es una entidad organizativa que permite agrupar y aislar recursos, aplicaciones y datos dentro de un entorno de nube. Esto facilita la gestión y el control de recursos específicos por parte de equipos, departamentos o aplicaciones, lo que mejora la seguridad y la eficiencia operativa.

Recursos y servicios: Los proyectos en la nube permiten la creación y gestión de una amplia variedad de recursos y servicios en función de las necesidades de la organización. Estos recursos pueden incluir máquinas virtuales, bases de datos, redes, almacenamiento, servicios de inteligencia artificial, contenedores, entre otros.

Control de acceso: Cada proyecto tiene su propio conjunto de políticas de control de acceso que determinan quién tiene permisos para acceder y gestionar los recursos dentro de ese proyecto. Esto garantiza que solo las personas autorizadas puedan realizar acciones específicas en los recursos de un proyecto.

Facturación y gestión de costos: Los proyectos en la nube también se utilizan para fines de facturación y gestión de costos. Cada proyecto puede tener su propio presupuesto y límites de gasto, lo que facilita el seguimiento de los costos asociados con cada iniciativa o departamento.

Aislamiento de recursos: Los recursos en un proyecto de nube están aislados de otros proyectos, lo que significa que las configuraciones y las acciones realizadas en un proyecto no afectarán a otros proyectos. Esto es fundamental para evitar interferencias y garantizar la seguridad y la disponibilidad de los recursos.

Gestión y monitoreo: La infraestructura de nube generalmente proporciona herramientas de gestión y monitoreo para supervisar el rendimiento, la disponibilidad y la salud de los recursos en cada proyecto. Esto permite identificar problemas y realizar ajustes según sea necesario.

Escalabilidad: Los proyectos en la nube pueden escalar vertical u horizontalmente según la demanda. Esto significa que se pueden agregar o reducir recursos de manera flexible para adaptarse a las necesidades cambiantes de la organización.

Seguridad: La seguridad es una consideración fundamental en los proyectos de nube. Los proveedores de servicios en la nube suelen ofrecer herramientas y servicios de seguridad que permiten proteger los datos y las aplicaciones en la nube. También es responsabilidad de la organización implementar prácticas sólidas de seguridad en sus proyectos de nube.

4.2.4.5. Rol.

Según (ITILCOM, 2020) el administrador de la plataforma es el encargado de crear los diferentes usuarios que tendrán acceso a la misma. También, deberá incluir en cada uno de estos perfiles los accesos adecuados y necesarios según las funciones que tengan la responsabilidad de realizar. (ITILCOM, 2020) distingue cuatro tipos de roles.

- **Rol de acceso a datos:** Este rol asignado a un usuario permite que el mismo tenga acceso a los datos . Normalmente con fines transaccionales o información mediante tablas. Al mismo tiempo, es posible que estos roles concretos tengan asignada una tarea particular que le da acceso a la gestión de unos datos específicos.
- **Rol de acceso a tareas:** Se encargan de definir las tareas que se tendrán que realizar. También, pueden establecer derechos para que otros usuarios puedan llevar a cabo una acción concreta.
- **Privilegios de sistema:** Se trata de un rol de usuario que otorga la capacidad de hacer tareas en las que participe el propio sistema. Se incluiría la propia administración de la configuración de la plataforma. También, la gestión que se hace de los datos y de la seguridad.
- **Función del usuario:** Los roles enfocados a asignar una determinada función a un usuario son también conocidos como roles de trabajo. Se trata de la asignación de un determinado rol a un usuario en la que se establece un determinado rango, que, a su vez, tiene asignado una serie de tareas y acceso a una serie de funcionalidades determinadas.

4.2.4.6. Usuarios.

Usuario de la nube significa cualquier persona individual que haga uso de los Servicios en la nube de un CSP (Políticas de Seguridad de Contenido) proporcionados a un cliente de la

nube, en función de una relación entre ese cliente de la nube y el usuario de la nube. Una persona individual puede ser tanto un usuario de la nube como un cliente de la nube si el contrato de la nube se concluye para la prestación de Servicios de computación en la nube a un solo usuario de la nube (Cloud Computing Regulatory Framework, s. f.).

4.2.4.7. RBAC.

El control de acceso basado en roles (RBAC) es una capacidad crítica para las organizaciones que implementan aplicaciones en la nube. Con RBAC, los analistas de seguridad y operaciones de TI obtienen visibilidad y supervisión completas de los permisos de las aplicaciones y la capacidad de administrar fácilmente quién tiene acceso a los recursos basados en la nube, a qué áreas de la red pueden acceder los usuarios y qué tipos de acciones pueden realizar los usuarios con los recursos que se les permite usar (sumo logic, s. f.). Según (sumo logic, s. f.) la metodología RBAC se basa en un conjunto de tres reglas principales que rigen el acceso a sistemas seguros:

- **Asignación de roles:** Cada transacción u operación sólo se puede llevar a cabo si el usuario ha asumido el rol apropiado. Una operación se define como cualquier acción realizada con respecto a un objeto de sistema o red que está protegido por RBAC. Los roles pueden ser asignados por una parte separada o seleccionados por el usuario que intenta realizar la acción.
- **Autorización de rol:** El propósito de la autorización de roles es garantizar que los usuarios solo puedan asumir un rol para el que se les haya otorgado la autorización adecuada. Cuando un usuario asume un rol, debe hacerlo con la autorización de un administrador.
- **Autorización de transacción:** una operación solo se puede completar si el usuario que intenta completar la transacción posee el rol apropiado.

4.3. Soluciones de despliegue en la nube

El despliegue en la nube según (Kinsta, 2021) se refiere al proceso de implementar aplicaciones o servicios en entornos de alojamiento basados en la nube, como Software como Servicio (SaaS), Plataforma como Servicio (PaaS) e Infraestructura como Servicio (IaaS). Este proceso abarca desde el diseño y la planificación hasta la implementación y ejecución de cargas de trabajo en la nube.

(Kinsta, 2021) define los siguientes beneficios del despliegue en la nube:

- 1. Despliegues eficientes y ágiles:** La nube permite automatizar el despliegue de código, bases de datos y versiones de aplicaciones, incluyendo la provisión de recursos. Esto acelera el tiempo necesario para llevar nuevas funcionalidades al mercado.
- 2. Ahorro de costos:** Con el despliegue en la nube, las organizaciones pueden controlar los costos al pagar sólo por los recursos que utilizan. Esto elimina la necesidad de invertir en costosas infraestructuras locales.
- 3. Escalabilidad y crecimiento:** Aprovechando la infraestructura global de proveedores de servicios en la nube, las empresas pueden expandirse fácilmente en otras geografías y aumentar sus capacidades según las necesidades cambiantes.
- 4. Innovación continua:** Los proveedores constantemente actualizan y mejoran sus servicios, lo que permite a las organizaciones adoptar nuevas tecnologías y modelos de negocio digitales de manera más rápida.
- 5. Resiliencia empresarial:** El diseño de alta disponibilidad y la tolerancia a fallos ofrecidos aseguran la continuidad de las aplicaciones y la recuperación frente a desastres.
- 6. Agilidad y escalabilidad:** Los recursos en la nube se pueden escalar automáticamente para responder a picos de demanda sin necesidad de una inversión previa en capacidad adicional.

7. Acceso global: Las aplicaciones se pueden acceder desde cualquier lugar o dispositivo, aprovechando la red global.

(Revista Byte, 2021) define los siguiente desafíos en la implementación de aplicaciones en la nube:

- 1. Costos iniciales elevados:** Aunque la nube puede ahorrar dinero a largo plazo, la migración inicial puede requerir inversión en capacitación y ajustes.
- 2. Migraciones complejas:** Mover aplicaciones existentes a la nube puede ser un proceso complicado, especialmente para sistemas heredados.
- 3. Monitorización insuficiente:** La supervisión y gestión de recursos en la nube puede ser compleja y requiere herramientas adecuadas.
- 4. Integraciones complejas:** La integración con sistemas existentes puede ser un desafío, especialmente si se utilizan múltiples servicios en la nube.
- 5. Cargas de trabajo dinámicas y estacionales:** Gestionar la escalabilidad y recursos para picos de demanda puede ser complejo.
- 6. Vulnerabilidades de aplicaciones:** La seguridad sigue siendo una preocupación, y se requieren medidas adicionales para proteger las aplicaciones en la nube.

La elección de la solución adecuada depende de las necesidades específicas de cada proyecto. Las Soluciones de despliegue en la nube ofrecen diversas formas de abordar las necesidades de infraestructura y aplicaciones en entornos digitales. Según (Ambit BST) las principales categorías se encuentran:

1. PaaS (Platform as a Service):

En PaaS, se proporciona una plataforma completa para desarrollar, ejecutar y administrar aplicaciones. Esto incluye entornos de desarrollo, herramientas de programación y servicios de alojamiento. Los usuarios de PaaS pueden centrarse en el desarrollo de aplicaciones sin preocuparse por la infraestructura subyacente, ya que el proveedor de PaaS se

encarga de la gestión de servidores y recursos. Ejemplos de plataformas PaaS incluyen Heroku, Google App Engine y Microsoft Azure App Service.

2. SaaS (Software as a Service):

SaaS ofrece aplicaciones y servicios completos a través de la nube. Los usuarios pueden acceder a estas aplicaciones a través de un navegador web o una aplicación cliente. No es necesario instalar ni mantener el software localmente, ya que todo se ejecuta en la infraestructura del proveedor de servicios en la nube. Ejemplos de aplicaciones SaaS incluyen Gmail, Microsoft 365, Salesforce y Dropbox.

3. IaaS (Infrastructure as a Service):

IaaS proporciona recursos de infraestructura en la nube, como máquinas virtuales, redes, almacenamiento y sistemas operativos. Los usuarios de IaaS tienen un mayor control sobre la configuración y administración de estos recursos, lo que les permite crear y gestionar su propia infraestructura de TI en la nube. Ejemplos de proveedores de IaaS incluyen Amazon Web Services (AWS), Microsoft Azure y Google Cloud Platform (GCP).

4.3.1. Servicios esenciales en la nube.

En el contexto de la evolución digital y empresarial, las soluciones de despliegue en la nube se han convertido en un pilar fundamental para la agilidad y la escalabilidad. Este tema se enfoca en la gestión y automatización de recursos en la nube, explorando servicios esenciales como cómputo, identidad, imágenes, mensajería, orquestación, panel de control y telemetría, que desempeñan un papel crucial en la implementación exitosa de infraestructuras en la nube modernas y eficientes (Conzultek, s.f).

4.3.1.1. Servicio de cómputo.

Es el encargado de las máquinas virtuales en la nube. Su trabajo es crear, gestionar y supervisar las instancias de máquinas virtuales, como administrar un edificio lleno de apartamentos virtuales para tus aplicaciones (Instituto Tecnológico de Costa Rica, 2014). A continuación, se proporciona información detallada sobre cuál de los servicios asume la función designada en las diversas soluciones en la nube disponibles (Instituto Tecnológico de Costa Rica, 2014):

- 1. OpenStack (Nova):** En OpenStack, Nova es el encargado de las máquinas virtuales en la nube. Su trabajo es crear, gestionar y supervisar las instancias de máquinas virtuales, como administrar un edificio lleno de apartamentos virtuales para tus aplicaciones. Nova es el servicio de cómputo en OpenStack que se encarga de la creación, programación y gestión de instancias de máquinas virtuales. Permite a los usuarios lanzar y administrar máquinas virtuales de manera eficiente en una nube OpenStack (IONOS, 2023).
- 2. Microsoft Azure (Azure Virtual Machines):** En Microsoft Azure, Azure Virtual Machines son los servidores en la nube. Puedes crear y administrar máquinas virtuales con diferentes sistemas operativos y capacidades para ejecutar tus aplicaciones. Azure Virtual Machines es el servicio de cómputo en la nube de Azure que permite crear y administrar máquinas virtuales escalables y flexibles en función de tus necesidades (TIC Portal, s.f.).
- 3. Google Cloud Platform (Google Compute Engine):** En Google Cloud Platform, Google Compute Engine es el centro de cómputo en la nube. Proporciona máquinas virtuales personalizables y potentes para ejecutar aplicaciones y cargas de trabajo. Google Compute Engine es el servicio de cómputo en la nube de Google Cloud que

permite crear y gestionar máquinas virtuales con diversas configuraciones y sistemas operativos (Beservices, s.f.).

4. **Amazon Web Services (AWS - Amazon EC2):** En Amazon Web Services (AWS), Amazon EC2 es la granja de servidores virtual en la nube. Se pueden lanzar instancias de máquinas virtuales con una variedad de configuraciones y sistemas operativos según las necesidades. Amazon EC2 (Elastic Compute Cloud) es el servicio de cómputo en la nube de AWS que permite el despliegue de instancias de máquinas virtuales escalables y personalizables (Rootstack, 2022).
5. **OpenNebula:** OpenNebula ofrece capacidades de cómputo en la nube a través de su servicio de gestión de máquinas virtuales, que se asemeja a Nova en OpenStack. Los usuarios pueden crear, administrar y supervisar instancias de máquinas virtuales en sus entornos de nube de acuerdo con sus requisitos específicos (Wikipedia, s.f).

4.3.1.2. Servicio de red.

Es el maestro de las conexiones en la nube. Imagina que es el ingeniero de tráfico de una ciudad digital. Su trabajo es asegurarse de que todas las máquinas virtuales (como edificios en la ciudad) puedan comunicarse entre sí y con el mundo exterior. Gestiona las carreteras virtuales y los semáforos para que los datos fluyan de manera segura y eficiente. En resumen, es el encargado de mantener todas las comunicaciones en la nube funcionando sin problemas (Instituto Tecnológico de Costa Rica, 2014). A continuación, se proporcionará información detallada sobre cuál de los servicios asume la función designada en las diversas soluciones en la nube disponibles:

1. **OpenStack (Neutron):** En OpenStack, Neutron es el arquitecto de la red de la nube. Se asegura de que las máquinas virtuales puedan comunicarse, cómo construir carreteras virtuales y configurar las señales de tráfico en tu nube para que todo fluya sin

problemas. Neutron es el servicio de red, que proporciona una API para la gestión de redes virtuales, subredes, routers y otras entidades de red. Permite la creación y configuración de redes, así como el control de la conectividad y las reglas de seguridad en una nube OpenStack (IONOS, 2023).

2. **Microsoft Azure (Azure Virtual Network):**En Microsoft Azure, Azure Virtual Network es el sistema de carreteras de la nube. Permite la creación de redes virtuales aisladas y la configuración de la conectividad entre máquinas virtuales y recursos en la nube de Azure. Azure Virtual Network es el servicio de red en Azure que facilita la creación y administración de redes virtuales, subredes y reglas de seguridad para tus aplicaciones y máquinas virtuales (TIC Portal, s.f.).
3. **Google Cloud Platform (Google Virtual Private Cloud - VPC):**En Google Cloud Platform, Google VPC es la estructura de red de la nube. Permite crear y gestionar redes virtuales privadas, subredes y reglas de firewall para tus recursos en la nube de Google. Google VPC (Virtual Private Cloud) es el servicio de red de Google Cloud que permite definir y administrar tu propia red virtual, configurar reglas de firewall y controlar la conectividad de tus recursos en la nube (Beservices, s.f.).
4. **Amazon Web Services (AWS - Amazon VPC):**En Amazon Web Services (AWS), Amazon VPC es el diseñador de la red de la nube. Permite crear redes virtuales aisladas, subredes y configurar reglas de seguridad para las instancias de Amazon EC2 y otros recursos en la nube de AWS. Amazon VPC (Virtual Private Cloud) es el servicio de red en AWS que permite crear y gestionar tu propia red virtual, definir subredes y controlar el acceso a instancias y recursos en la nube de AWS mediante grupos de seguridad y listas de control de acceso (Rootstack, 2022).
5. **Open nebula:** OpenNebula proporciona capacidades de red a través de sus recursos de red definidos por el usuario en sus entornos de nube, aunque no tiene un servicio

específico equivalente a Neutron. Los usuarios pueden definir y configurar redes virtuales y topologías de red personalizadas según sus necesidades en OpenNebula (Wikipedia, s.f).

4.3.1.3. Servicio de identidad.

Es el guardián de la puerta de la nube. Su tarea es autenticar a los usuarios y servicios y proporcionar autorización para acceder a los recursos. Es el encargado de controlar quién tiene las llaves para entrar a diferentes áreas de la nube (Instituto Tecnológico de Costa Rica, 2014). A continuación, se proporcionará información detallada sobre cuál de los servicios asume la función designada en las diversas soluciones en la nube disponibles:

- 1. OpenStack (Keystone):** En OpenStack, Keystone es el guardián de la puerta de la nube. Su tarea es autenticar a los usuarios y servicios y proporcionar autorización para acceder a los recursos. Es el encargado de controlar quién tiene las llaves para entrar a diferentes áreas de la nube. Keystone es el servicio de identidad y autenticación en OpenStack, encargado de gestionar la autenticación de usuarios y servicios, así como de controlar el acceso y las políticas de autorización en una nube OpenStack (IONOS, 2023).
- 2. Microsoft Azure (Azure Active Directory):** En Microsoft Azure, Azure Active Directory (Azure AD) es el administrador de identidad. Proporciona la autenticación y la administración de identidades de usuarios y aplicaciones para acceder a recursos en Azure y otras aplicaciones de Microsoft. Azure Active Directory es el servicio de identidad y acceso en la nube de Azure que permite gestionar usuarios, grupos y aplicaciones, y controlar el acceso a recursos de manera segura (TIC Portal, s.f.).
- 3. Google Cloud Platform (Google Identity and Access Management - IAM):** En Google Cloud Platform, IAM es el conserje de la nube. Administra las identidades y

los permisos de acceso de usuarios y servicios a recursos en la nube de Google. Google Identity and Access Management (IAM) es el servicio de gestión de identidad y acceso en la nube de Google Cloud que permite controlar de manera precisa quién puede acceder y qué acciones pueden realizar en los recursos de la nube (Beservices, s.f.).

4. **Amazon Web Services (AWS - AWS Identity and Access Management - IAM):** En Amazon Web Services (AWS), IAM es el guardián de los recursos en la nube. Controla el acceso a servicios y recursos de AWS al gestionar las identidades y los permisos de los usuarios y servicios. AWS Identity and Access Management (IAM) es el servicio de gestión de identidad y acceso en la nube de AWS que permite definir políticas de acceso y controlar quién puede hacer qué en los recursos de AWS (Rootstack, 2022).
5. **OpenNebula:** OpenNebula ofrece capacidades de gestión de identidad y acceso a través de la configuración de usuarios y grupos en sus entornos de nube, aunque no tiene un servicio específico equivalente a Keystone. Los usuarios y servicios pueden ser autenticados y autorizados según las políticas definidas en OpenNebula (Wikipedia, s.f.).

4.3.1.4. Servicio de imágenes.

Es el archivero de la nube. Su tarea es almacenar y gestionar imágenes de máquinas virtuales, como si estuviera manteniendo una biblioteca de sistemas operativos y plantillas de aplicaciones para usar en la nube (Instituto Tecnológico de Costa Rica, 2014). A continuación, se proporcionará información detallada sobre cuál de los servicios asume la función designada en las diversas soluciones en la nube disponibles:

1. **OpenStack (Glance):** En OpenStack, Glance es el archivero de la nube. Su tarea es almacenar y gestionar imágenes de máquinas virtuales, como si estuviera manteniendo una biblioteca de sistemas operativos y plantillas de aplicaciones para usar en la nube.

Glance es el servicio de imágenes en OpenStack, que permite la gestión de imágenes de máquinas virtuales, incluyendo la carga, el almacenamiento y el registro de imágenes que pueden ser utilizadas para crear nuevas instancias de máquinas virtuales (IONOS, 2023).

2. **Microsoft Azure (Azure Marketplace):** En Microsoft Azure, Azure Marketplace es la tienda de aplicaciones de la nube. Ofrece imágenes preconfiguradas de máquinas virtuales y aplicaciones que puedes implementar directamente en Azure. Azure Marketplace es un catálogo de imágenes y soluciones preconfiguradas que se pueden utilizar para desplegar máquinas virtuales y aplicaciones en Azure (TIC Portal, s.f.).
3. **Google Cloud Platform (Google Cloud Marketplace):** En Google Cloud Platform, Google Cloud Marketplace es similar a una tienda de aplicaciones en la nube. Proporciona imágenes preconfiguradas de máquinas virtuales y aplicaciones listas para usar en la plataforma de Google Cloud. Google Cloud Marketplace es un catálogo de soluciones que incluye imágenes de máquinas virtuales y aplicaciones preconfiguradas que se pueden implementar en Google Compute Engine y otros servicios de Google Cloud (Beservices, s.f.).
4. **Amazon Web Services (AWS - AWS Marketplace):** En Amazon Web Services (AWS), AWS Marketplace es el centro comercial de imágenes de máquinas virtuales y aplicaciones. Ofrece una amplia selección de imágenes preconfiguradas y software listo para usar en la nube de AWS. AWS Marketplace es un mercado en línea que ofrece una variedad de imágenes de máquinas virtuales y software listo para usar que se pueden implementar en instancias de Amazon EC2 y otros servicios de AWS (Rootstack, 2022).
5. **OpenNebula:** OpenNebula permite la gestión de imágenes de máquinas virtuales a través de su servicio de almacenamiento de imágenes. Los usuarios pueden cargar y

utilizar imágenes personalizadas para crear y gestionar instancias de máquinas virtuales en sus entornos de nube (Wikipedia, s.f).

4.3.1.5. Servicio de panel de control (Dashboard).

Es el tablero de control de la nube. Proporciona una interfaz gráfica de usuario para ver y gestionar los recursos en la nube de manera visual, como si se estuvieran controlando los interruptores y botones de una sala de control (Instituto Tecnológico de Costa Rica, 2014). A continuación, se proporcionará información detallada sobre cuál de los servicios asume la función designada en las diversas soluciones en la nube disponibles:

- 1. OpenStack (Horizon):** En OpenStack, Horizon es el panel de control web que proporciona una interfaz gráfica de usuario para administrar y supervisar los recursos en una nube OpenStack. Permite a los administradores y usuarios gestionar máquinas virtuales, redes, almacenamiento y otros servicios de OpenStack de manera visual (IONOS, 2023).
- 2. Microsoft Azure (Azure Portal):** En Microsoft Azure, Azure Portal es el panel de control web que ofrece una experiencia unificada para administrar y supervisar los recursos en la nube de Azure. Proporciona una interfaz gráfica de usuario para gestionar máquinas virtuales, bases de datos, redes y otros servicios de Azure (TIC Portal, s.f.).
- 3. Google Cloud Platform (Google Cloud Console):** En Google Cloud Platform, Google Cloud Console es el panel de control web que permite administrar y supervisar los recursos en la nube de Google Cloud. Ofrece una interfaz gráfica de usuario para gestionar máquinas virtuales, bases de datos, servicios de almacenamiento y otros servicios de Google Cloud (Beservices, s.f.).
- 4. Amazon Web Services (AWS - AWS Management Console):** En Amazon Web Services (AWS), AWS Management Console es el panel de control web que

proporciona una interfaz gráfica de usuario para administrar y supervisar los recursos en la nube de AWS. Permite gestionar instancias de EC2, bases de datos, almacenamiento y otros servicios de AWS (Rootstack, 2022).

- 5. OpenNebula:** OpenNebula no tiene un panel de control web incorporado como los mencionados anteriormente. Sin embargo, los usuarios pueden desarrollar interfaces personalizadas o utilizar herramientas de terceros para gestionar y supervisar sus entornos de nube basados en OpenNebula (Wikipedia, s.f).

4.3.1.6. Servicio de gestión de claves.

Es el custodio de las llaves secretas en la nube. Su trabajo es crear, almacenar y proteger las claves criptográficas, certificados y otros secretos sensibles utilizados en la nube. Imagina que es el guardián de un cofre de tesoros digitales que asegura que solo las personas autorizadas tengan acceso a ellos (Instituto Tecnológico de Costa Rica, 2014). A continuación, se proporcionará información detallada sobre cuál de los servicios asume la función designada en las diversas soluciones en la nube disponibles:

- 1. OpenStack (Barbican):** En OpenStack, Barbican es el servicio de gestión de claves y secretos. Ofrece un lugar seguro para el almacenamiento y la gestión de claves de cifrado, certificados y otros secretos sensibles utilizados en una nube OpenStack. Barbican se encarga de proteger y gestionar estos secretos de manera centralizada (IONOS, 2023).
- 2. Microsoft Azure (Azure Key Vault):** En Microsoft Azure, Azure Key Vault es el servicio de gestión de claves y secretos que permite la creación, el almacenamiento y el control de claves criptográficas, certificados y secretos en la nube de Azure. Proporciona una capa adicional de seguridad para proteger datos sensibles (TIC Portal, s.f.).

- 3. Google Cloud Platform (Google Cloud Key Management Service - KMS):** En Google Cloud Platform, Google Cloud KMS es el servicio de gestión de claves que permite generar, almacenar y controlar claves de cifrado utilizadas en Google Cloud. También ofrece la gestión de secretos y acceso a datos sensibles (Beservices, s.f.).
- 4. Amazon Web Services (AWS - AWS Key Management Service - KMS):** En Amazon Web Services (AWS), AWS KMS es el servicio de gestión de claves que permite la creación y el control de claves de cifrado utilizadas en AWS. Además de la gestión de claves, AWS KMS proporciona la capacidad de cifrar y descifrar datos de manera segura (Rootstack, 2022).
- 5. OpenNebula:** OpenNebula no tiene un servicio de gestión de claves y secretos incorporado de forma nativa como los mencionados anteriormente. Sin embargo, los usuarios pueden implementar soluciones de gestión de claves y secretos de terceros en sus entornos de nube basados en OpenNebula según sus necesidades específicas (Wikipedia, s.f.).

4.4. Clúster

Los clusters son grupos de servidores que se gestionan juntos y participan en la gestión de carga de trabajo. Un clúster puede contener nodos o servidores de aplicaciones individuales. Un nodo suele ser un sistema físico con una dirección IP de host distinta que ejecuta uno o varios servidores de aplicaciones. Los clústeres se pueden agrupar bajo la configuración de una célula, que asocia lógicamente muchos servidores y clústeres con distintas configuraciones y aplicaciones entre sí en función de la discreción del administrador y de lo que tenga sentido en sus entornos organizativos (IBM, s.f.)

Los clústers son responsables de equilibrar la carga de trabajo entre los servidores. Los servidores que forman parte de un clúster se denominan miembros del clúster. Cuando se instala una aplicación en un clúster, la aplicación se instala automáticamente en cada miembro del clúster. Se puede configurar un clúster para proporcionar equilibrio de carga de trabajo con integración de servicios o con beans controlados por mensajes en el servidor de aplicaciones (IBM, s.f.).

4.4.1. Cluster de base de datos.

Un cluster de base de datos es una configuración avanzada en la que dos o más servidores de bases de datos se combinan para trabajar juntos como una única entidad de procesamiento de bases de datos. El objetivo principal de un cluster de base de datos es mejorar la disponibilidad, el rendimiento y la escalabilidad de una base de datos, lo que lo hace especialmente útil para aplicaciones empresariales críticas donde el tiempo de inactividad es costoso y el rendimiento es fundamental (IBM, 2015).

4.4.1.1. Mariadb - Galera.

Galera Cluster es un paquete de software para sistemas operativos Linux que permite la creación y gestión de clústeres de MySQL, XtraDB y MariaDB. Esta aplicación de clúster se basa en el motor de almacenamiento InnoDB o su derivado XtraDB. Experimentalmente es compatible con el motor MyISAM, que durante mucho tiempo se utilizó también en MySQL y MariaDB como predecesor de InnoDB. Galera Cluster aplica el principio de la replicación ya que sincroniza al almacenamiento de datos en los diversos nodos independientes del clúster. Todos los procesos de copia y modificación de los datos almacenados tienen lugar simultáneamente en todas las unidades de almacenamiento primarias y secundarias, de forma que los datos presentes en todos los nodos están siempre actualizados y no difieren entre ellos (IONOS Digital Guide, 2020).

La expresión mínima de un Galera Cluster consiste en tres nodos (los desarrolladores recomiendan en general usar un número impar). Si uno de los nodos falla durante la transacción de datos, por ejemplo, debido a problemas de la red o del sistema, los otros dos nodos siguen formando la mayoría necesaria para realizar la transacción correctamente (IONOS Digital Guide, 2020).

¿Cómo funcionan los clústeres MariaDB de Galera?

Un clúster MariaDB basado en Galera Cluster se caracteriza por el hecho de que todos los nodos de la red tienen los mismos datos en todo momento. Por tanto, en el software de clúster desaparece la distinción tradicional entre servidores de bases de datos maestros (servidores en los que se puede escribir) y esclavos (servidores de los que solo se puede leer). Los usuarios pueden escribir datos en cualquier nodo de almacenamiento y estos se reenviarán automáticamente a todos los demás participantes del clúster. Esta propiedad también se llama multimaster (IONOS Digital Guide, 2020).

Para garantizar este intercambio de datos flexible, Galera Cluster emplea un procedimiento de replicación síncrona para la administración, basado en el intercambio de certificados. Según (IONOS Digital Guide, 2020) si los datos se replican, es decir, se escriben en una de las bases de datos del clúster MariaDB de Galera, se aplican estos dos principios:

A cada transacción en la base de datos se le asigna un número de secuencia único. Antes de que un nodo del clúster apruebe los cambios respectivos en la base de datos, este compara el número secuencial con el número de la última transacción confirmada. En la comprobación, todos los nodos llegan a la misma conclusión (Transacción confirmada o Transacción rechazada). El nodo que inició la transacción transmite el resultado al cliente.

En cada transacción se actualizan todas las réplicas de la base de datos. De este modo, si una transacción se confirma después de la certificación, todos los nodos hacen los cambios

correspondientes en su registro de datos. Si los nodos de un Galera Cluster experimentan algún problema técnico, solo se podrán excluir de la replicación síncrona temporalmente.

¿Cómo es la estructura de un Galera Cluster?



Figura 2: IONOS. (2010). *Arquitectura interna de un Galera Cluster* [Imagen]. Galera Cluster: vista general de un clúster MariaDB. <https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/galera-cluster-para-mariadb/>

La Figura 2 muestra la arquitectura interna de un Galera Cluster que según (IONOS Digital Guide, 2020) está estructurada en torno a estos cuatro componentes:

- **Sistema de gestión de la base de datos:** el SGBD es la unidad central del clúster. En cada nodo se ejecuta el servidor de base de datos correspondiente. Como se mencionó anteriormente, Galera Cluster es compatible con MySQL y Percona XtraDB, además de con MariaDB.
- **API wsrep:** la API wsrep define e implementa la interfaz y las funciones de acceso a los servidores de bases de datos implicados. Asimismo, wsrep controla la replicación de datos. Entre otras cosas, la API proporciona los wsrep hooks (enlaces al servidor de la base de datos para su replicación) y la función dlopen(), que permite la comunicación con los hooks.

- **Plugin de replicación de Galera o Galera replication plug-in:** este plugin implementa la API de wsrep. Para esto, proporciona una capa para el proceso de certificación, una capa para la replicación (incluido el protocolo) y el marco de comunicación de grupo.
- **Plugins de comunicación de grupo o Group Communication plug-ins:** Galera Cluster proporciona varias extensiones para la implementación de sistemas de comunicación de grupo como el Spread Toolkit o gcomm. Estos plugins están basados en el marco de comunicación de grupo.

4.4.2. Cluster de almacenamiento.

Un cluster de almacenamiento de datos, también conocido como clúster de almacenamiento, se refiere a un grupo o conjunto de dispositivos de almacenamiento físicos o lógicos que están interconectados y operan de manera conjunta para proporcionar almacenamiento de datos de alto rendimiento, disponibilidad y confiabilidad. Estos clústeres se utilizan en entornos de centros de datos y redes empresariales para satisfacer las demandas de almacenamiento de datos cada vez más grandes y críticas (VMware, 2022).

Es una colección de almacenes de datos con recursos compartidos y una interfaz de administración compartida. Los clústeres de almacenes de datos son a los almacenes de datos lo mismo que los clústeres son a los hosts (VMware, 2022).

4.4.2.1. Almacenamiento en bloques.

El almacenamiento en bloque, también denominado almacenamiento a nivel de bloque, es una tecnología que se utiliza para almacenar archivos de datos en redes de área de almacenamiento (SAN) o entornos de almacenamiento basados en la nube. Los desarrolladores prefieren el almacenamiento en bloque para situaciones de cálculo en las que requieren un transporte de datos rápido, eficaz y fiable (IBM, s.f.).

El almacenamiento en bloque descompone los datos en bloques y, a continuación, los almacena como partes separadas, cada una con un identificador exclusivo. La SAN coloca esos bloques de datos donde sean más eficaces. Esto significa que puede almacenar esos bloques en distintos sistemas y cada bloque se puede configurar (o particionar) para que trabaje con diferentes sistemas operativos (IBM, s.f.).

Por ejemplo, el despliegue de máquinas virtuales en una empresa. Con el almacenamiento en bloque, se puede crear y formatear fácilmente un volumen de almacenamiento basado en bloque para almacenar los VMFS (Virtual Machine File System). A continuación, un servidor físico se puede conectar a dicho bloque y crear varias máquinas virtuales. Además, al crear un volumen basado en bloques, instalar un sistema operativo y conectarse a ese volumen, los usuarios pueden compartir archivos mediante ese sistema operativo nativo. Los despliegues de cloud privado son otro uso excelente del almacenamiento en bloque (IBM, s.f.).

4.4.2.2. Almacenamiento de objetos.

Según (AWS, s.f.) el almacenamiento de objetos es una tecnología que almacena y administra datos en un formato no estructurado denominado objetos. Cada objeto se etiqueta con un identificador único y contiene metadatos que describen el contenido subyacente. Por ejemplo, el almacenamiento de objetos para fotos contiene metadatos relacionados con el fotógrafo, la resolución, el formato y el tiempo de creación. Los desarrolladores utilizan el almacenamiento de objetos para almacenar datos no estructurados, como texto, vídeo e imágenes.

(AWS, s.f.) afirma que tanto las soluciones de almacenamiento en bloques como las de objetos son beneficiosas según el caso de uso. El almacenamiento en bloque proporciona valores de baja latencia y alto rendimiento en varios casos de uso. Sus funciones son útiles principalmente para el almacenamiento de bases de datos estructuradas, los volúmenes del

sistema de archivos de VM y los grandes volúmenes de cargas de lectura y escritura. El almacenamiento de objetos se aprovecha más para grandes cantidades de datos no estructurados, especialmente cuando la durabilidad, el almacenamiento ilimitado, la escalabilidad y la administración compleja de metadatos son factores relevantes para el rendimiento general.

4.4.2.3. Soluciones de almacenamiento distribuido.

Según (IONOS Digital Guide, 2020) las soluciones de almacenamiento distribuido son sistemas de almacenamiento que distribuyen datos en múltiples ubicaciones físicas o nodos de manera que los datos estén disponibles y sean accesibles desde diferentes ubicaciones geográficas o servidores. Estas soluciones son fundamentales para entornos empresariales donde se requiere escalabilidad, alta disponibilidad y acceso a datos en tiempo real.

Los sistemas de almacenamiento distribuido son la solución para almacenar y gestionar datos que no caben en un servidor convencional. En este sentido, el tamaño no es el único problema, sino que los sistemas de archivos clásicos, con su estructura en carpetas, tampoco son compatibles con datos sin estructurar (IONOS Digital Guide, 2020).

4.4.2.3.1. Glusterfs.

GlusterFS es un sistema de archivos en red escalable y distribuido, definido para ser utilizado en el espacio de usuario, es decir, sin utilizar el espacio de almacenamiento crítico del sistema, y de esta forma no compromete el rendimiento. Se utiliza en multitud de entornos, como en análisis de datos, streaming y otras tareas intensivas de almacenamiento. Lo mejor de todo es que GlusterFS es un software gratuito y Open Source (BITS, s.f.).

¿Cómo funciona GlusterFS?

(IONOS Digital Guide, 2020) afirma que sistema de archivos distribuido es conveniente sólo si se conectan varios ordenadores entre sí. Según la descripción oficial de GlusterFS, se

requieren al menos tres servidores, pero no se trata de servidores en sentido literal, sino que puede tratarse de prácticamente cualquier tipo de hardware físico o emulado. También se pueden usar, además de todo tipo de ordenadores, máquinas virtuales, que tienen muchas ventajas, en especial en materia de flexibilidad.

Los servidores integrados funcionan como nodes (nodos) y se conectan a través de la red TCP/IP. Los dispositivos conectados forman así lo que se denomina una trusted pool, es decir, un conjunto de servidores de confianza. Estos servidores ponen sus memorias a disposición del usuario en forma de bricks (ladrillos), en los cuales se crean, finalmente, los volúmenes (unidades de volumen), que luego pueden incorporar y usarse como soportes de almacenamiento normales. Los ordenadores que acceden al sistema son denominados clientes o clients y es posible que también funcione como servidor al mismo tiempo (IONOS Digital Guide, 2020).

Una característica especial de GlusterFS es su gran escalabilidad, que permite que se puedan añadir a posteriori tantos nodes y bricks como se quiera. De este modo, el espacio de almacenamiento puede adaptarse continuamente a los requisitos de cada momento. El tamaño máximo del espacio de almacenamiento que se puede manejar es de varios petabytes (IONOS Digital Guide, 2020).

También es posible crear clusters tipo RAID, que requieren que se añada una unidad copiada (replicated volume) en lugar de una distribuida (distributed volume), como suele ser la opción por defecto. Dicha unidad copiada guarda cada archivo por duplicado y se corresponde con el llamado mirror RAID (IONOS Digital Guide, 2020).

4.4.2.3.2. Cephfs.

Ceph es una solución completa de almacenamiento que dispone de un sistema de archivos propio, el Ceph File System (CephFS). Con Ceph pueden almacenarse datos distribuidos en diversos componentes de la propia red. Además, los datos pueden guardarse en

diferentes soportes físicos de almacenamiento. Ceph garantiza una gran flexibilidad en la elección del soporte de almacenamiento, así como una gran escalabilidad (IONOS Digital Guide, 2020).

Cómo funciona Ceph

Ceph requiere que varios ordenadores estén conectados entre ellos en lo que se denomina un cluster (literalmente grupo o montón, es decir, un conjunto de varios ordenadores). Cada ordenador conectado se llama node o nodo. Según (IONOS Digital Guide, 2020) en un cluster existen diferentes tipos de nodos, según las tareas que realicen:

- **Monitor nodes:** Gestionan el estado de cada uno de los nodos en el cluster y vigilan en especial los componentes manager service, object storage service y metadata server (MDS). Para poder asegurar cierta seguridad, se recomienda disponer de al menos tres nodos monitores.
- **Manager:** Gestionan el estado de la utilización del espacio, de la carga del sistema y del nivel de utilización de los nodos.
- **Ceph OSDs (Object Storage Devices):** Son los servicios de fondo que realmente se encargan de gestionar los archivos: son responsables del almacenamiento, el duplicado y la restauración de los datos. Se recomienda tener al menos tres OSD en el cluster.
- **Metadata server (MDSs):** Se encargan de almacenar metadatos como, por ejemplo, rutas de almacenamiento, sellos de tiempo y nombres de los archivos guardados en CephFS, por motivos de rendimiento. Están creados siguiendo el estándar POSIX y pueden solicitarse mediante líneas de comando de Unix, como ls, find y like.

El componente clave del almacenamiento de datos es un algoritmo llamado CRUSH (Controlled Replication Under Scalable Hashing, es decir, replicación controlada bajo hashing escalable). Este algoritmo es capaz de encontrar un OSD con el archivo solicitado gracias a una tabla de asignaciones.

La distribución de los archivos en Ceph se realiza de forma pseudoaleatoria, es decir, de forma que pudiera parecer que se ubican de cualquier manera. En realidad, sin embargo, CRUSH calcula el lugar más adecuado para almacenarlos basándose en criterios definidos por el administrador de la red. Al hacerlo, además, se duplican los archivos y se almacenan en soportes físicos separados (IONOS Digital Guide, 2020).

Los archivos se distribuyen en los llamados placement groups (grupos de ubicación), procesando el nombre de archivo como valor hash. Otra característica en la que se basa la ubicación es, por ejemplo, el número de duplicados del archivo (IONOS Digital Guide, 2020).

Acceso a los datos guardados

La base del almacenamiento de datos en Ceph se llama RADOS (a reliable, distributed object store comprised of self-healing, self-mapping, intelligent storage nodes), es decir, una memoria fiable y distribuida compuesta por nodos de almacenamiento inteligentes que se auto regeneran y se autoorganizan. Según (IONOS Digital Guide, 2020) se puede acceder a los archivos guardados usando diferentes métodos:

- **Librados:** Se puede acceder a ellos de forma nativa usando la biblioteca librados a través de las interfaces de programación (API) con lenguajes de programación y de script como C/C++, Python, Java o PHP.
- **Radosgw:** Por esta gateway se pueden leer o escribir datos usando el protocolo de Internet HTTP.
- **CephFS:** Se trata del sistema de archivos propio de Ceph, que se ajusta al estándar POSIX, ofrece un módulo de núcleo para los ordenadores que accedan y es compatible con FUSE (complemento al sistema de archivos, sin derechos de administrador).
- **RADOS Block Device:** Se integra como memoria orientada a bloques a través de módulos de núcleo o sistemas virtuales como QEMU/KVM.

¿Cuándo conviene utilizar cada sistema?

Ceph, gracias a sus variadas interfaces, funciona bien en redes heterogéneas, en las que no solo se utiliza Linux, sino también otros sistemas operativos. El punto fuerte de GlusterFS, en cambio, es el almacenamiento de grandes cantidades de datos en formato tradicional, así como de datos de gran tamaño. Puesto que Ceph se desarrolló desde un primer momento como solución open source, en el pasado resultaba más fácil utilizarlo en muchos casos, hasta que GlusterFS también pasó a ser de código abierto. Un ámbito de aplicación muy relevante para los sistemas de almacenamiento distribuido son los servicios en la nube. En este sentido, OpenStack es uno de los proyectos de software más importantes que ofrecen arquitecturas para computación en la nube. Ambos, GlusterFS y Ceph, funcionan igual de bien con OpenStack (IONOS Digital Guide, 2020).

4.5. Automatización

La automatización consiste en usar la tecnología para realizar tareas con muy poca intervención humana. Permite agilizar los procesos, ampliar los entornos y crear flujos de trabajo de integración, distribución e implementación continuas (CI/CD). (VMware, s.f.)

Hay muchos tipos de automatización:

- Automatización de la TI.
- Automatización empresarial.
- Automatización robótica de los procesos.
- Automatización industrial.
- La inteligencia artificial.
- Aprendizaje automático.
- Aprendizaje profundo.

(NUTANIX. s.f.) Sin automatización en la nube, los equipos de TI deben implementar manualmente las cargas de trabajo. Como es de esperarse, esta actividad es lenta y tediosa; los administradores de TI deben realizar tareas repetitivas y laboriosas como:

- Gestionar los recursos de nube.
- Configurar clusters de máquina virtual (VM).
- Creación de redes virtuales
- Implementar cargas de trabajo en la nube.
- Mantener el ritmo de los estándares de disponibilidad y rendimiento.

Ventajas de la automatización en la nube

Según (VMware, s.f.) las ventajas de la automatización en la nube son las siguientes:

- **Reducción de procesos propensos a errores:** la automatización en la nube ayuda a reducir los procesos manuales propensos a errores y a entregar recursos de infraestructura más rápido. La automatización de la nube debe admitir múltiples hipervisores y estándares de virtualización en recursos de hardware como KVM, XenServer, Hyper-V, Docker y Kubernetes, así como el ciclo de vida de desarrollo de software para equipos de programación.
- **Ahorro de costes:** la automatización en la nube ahorra dinero a las empresas al reducir dinámicamente el tiempo que lleva aprovisionar recursos de infraestructura, eliminando errores y eliminando cuellos de botella. También ahorra dinero al optimizar la ubicación de la carga de trabajo, de modo que use el hardware menos costoso y priorice el hardware para proyectos importantes. También aumenta el control a través de políticas. El uso de estructuras de facturación de nube pública está diseñado para reflejar aproximadamente un 40% de ahorro en comparación con los centros de datos internos o las instalaciones de nube privada.

4.5.1. Terraform.

Terraform Cloud es un servicio alojado desarrollado por HashiCorp que proporciona un espacio de trabajo colaborativo para que los equipos utilicen Terraform, una herramienta de software de infraestructura como código (IaC) de código abierto. Permite a los equipos administrar el aprovisionamiento, el cumplimiento y la administración de la infraestructura en varios proveedores de nube, centros de datos y servicios (env0, s.f.).

Importancia de la infraestructura como código y porque automatizar la infraestructura

La infraestructura como código (IaC) es el proceso de administrar y aprovisionar la infraestructura de TI a través de archivos de definición legibles por máquina, en lugar de procesos manuales o herramientas de configuración interactivas. IaC permite a los equipos crear, cambiar y administrar la infraestructura de una manera segura, consistente y repetible. La automatización del aprovisionamiento de infraestructura utilizando herramientas de IaC como Terraform Cloud mejora la productividad, reduce el riesgo y aumenta la velocidad del negocio (env0, s.f.).

Beneficios de Terraform Cloud

- Colaboración y control de acceso.
- Administración remota del estado.
- Integración de control de versiones.
- Escalabilidad y aprovisionamiento de infraestructura multinube.

Características

- Espacios de trabajo y proyectos.
- Control de acceso basado en roles.
- Inicio de sesión único (SSO).
- Aplicación de políticas y gobernanza (Sentinel y OPA).

- Almacenamiento variable seguro.
- Integración con canalizaciones de CI/CD.
- Agentes auto hospedados.

4.5.2. SaltStack.

SaltStack es un software libre y multiplataforma de la Fundación de Software Apache. Permite realizar el mantenimiento a distancia, crear estados de destino previamente definidos e iniciar controles, tanto en el propio centro de datos de la empresa como en una nube externa (independientemente del proveedor). El fabricante menciona como principales ventajas de SaltStack la sencillez de la instalación, la rapidez de control de la comunicación en milisegundos y la posibilidad de gestionar un gran número de servidores en paralelo (IONOS, 2023).

Ventajas para los administradores.

1. No importa qué sistema deseen modificar los administradores: los comandos en SaltStack son siempre los mismos. Esto lo hace fácil de manejar y fácil de aprender.
2. Esta herramienta no solo asegura la propagación automatizada de las configuraciones, sino que también puede reaccionar ante acontecimientos, ya que registra qué tipo de comunicación se intercambia en la infraestructura.
3. SaltStack está completamente incluido en Salt Open y está disponible para el público en la licencia Apache 2.0. Además, podrás participar activamente en el diseño del software. Si también quieres recibir asistencia técnica del fabricante, se ofrece la versión Enterprise.

(SaltStack, 2023) En resumen salt es un marco de ejecución remota de código abierto basado en Python que se utiliza para:

- Gestión de la configuración

- Automatización
- Aprovisionamiento
- Orquestación

4.5.3. Ansible.

Ansible es una herramienta open source que automatiza los procesos informáticos para preparar la infraestructura, gestionar la configuración, implementar las aplicaciones y organizar los sistemas, entre otros procedimientos manuales de TI. A diferencia de las herramientas de gestión más simples, los usuarios de Ansible (como los administradores de sistemas, los desarrolladores y los arquitectos) pueden utilizar la automatización que ofrece esta herramienta para instalar sistemas de software, automatizar las tareas diarias, preparar la infraestructura, mejorar la seguridad y el cumplimiento, ejecutar parches en los sistemas y compartir la automatización en toda la empresa (RedHat, 2021).

¿Cómo funciona Ansible?

Ansible se conecta con los recursos que desea automatizar e implementa programas para ejecutar instrucciones que se llevarían a cabo de forma manual. Los programas utilizan módulos de Ansible que se escribieron en función de las expectativas específicas de la interfaz, los comandos y la conectividad del extremo. Ansible ejecuta los módulos con el protocolo SSH de forma predeterminada y los elimina al finalizar, en caso necesario.

No se requiere ningún servidor, daemon ni base de datos adicional. Por lo general, trabajará con su terminal favorito, un editor de texto y un sistema de control de versiones para hacer un seguimiento de los cambios en el contenido.(RedHat, 2021)

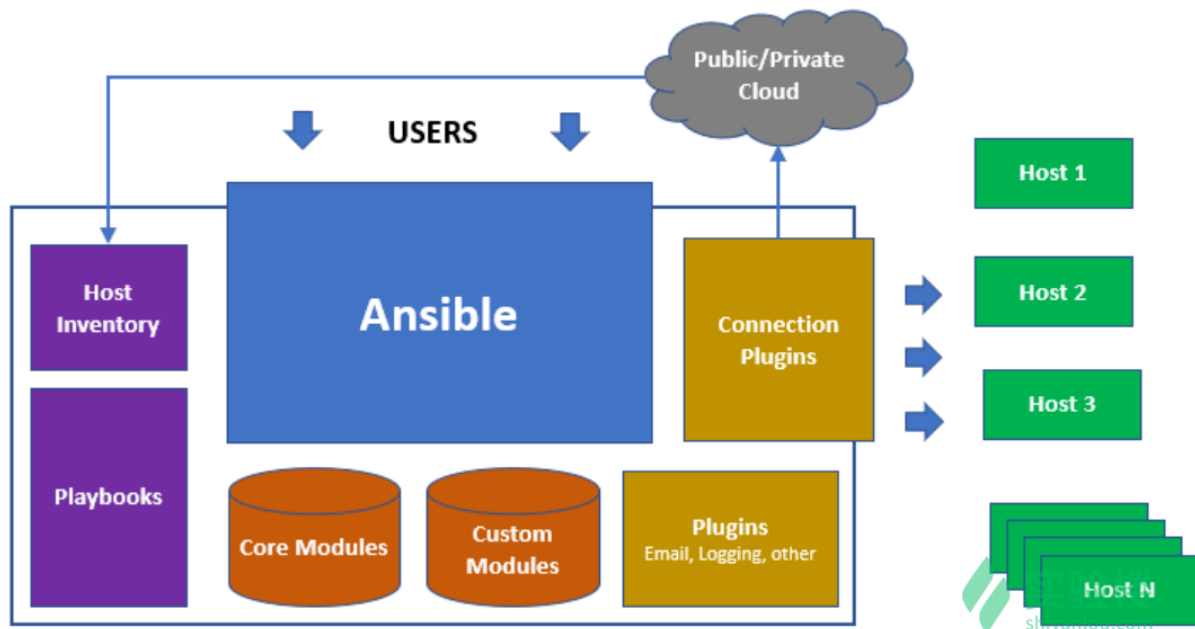


Figura 3: Avi. (s.f.). *Arquitectura Ansible* [Imagen]. Ansible para principiantes: conceptos básicos de Ansible y cómo funciona. <https://geekflare.com/es/ansible-basics/>

En la Figura 3 muestra La arquitectura mencionada se basa en servidores Linux que actúan como nubes pública o privada y sirven como repositorio para configuraciones de TI. Utiliza Ansible como motor de automatización, que se conecta a máquinas host a través de SSH.

Los componentes clave incluyen un inventario de hosts (lista de direcciones IP), módulos (código ejecutable en Ansible), y libros de jugadas (definen flujos de trabajo en YAML). Los módulos realizan acciones en las máquinas host, y se pueden personalizar. También hay plugins que son tipos especiales de módulos, como plugins de devolución de llamada, caché y acción.

Los complementos de conexión permiten alternativas a SSH, como el complemento de conexión para contenedores Docker. En resumen, Ansible facilita la automatización de tareas de configuración y gestión en entornos Linux mediante libros de jugadas, módulos y plugins. Según (invgate, 2023) los beneficios de ansible son los siguientes:

1. **Reduce los recursos necesarios para la gestión IT:** los Sysadmins pueden contener cientos o incluso miles de máquinas desde un único punto y de una sola vez.
2. **Hace accesible la automatización:** uno de los objetivos de Ansible fue que se requiriera un aprendizaje mínimo de uso. La plataforma utiliza YAML (Yet Another Markup Language), un lenguaje legible para personas, con elementos de otros idiomas de programación.
3. **No afecta al rendimiento:** no requiere agentes, ni software, ni recursos informáticos que se ejecute o instale en sistemas gestionados.
4. **Garantiza la coherencia:** la plataforma se diseñó para ser mínima y permitir a los usuarios crear entornos coherentes. Además, como toda la operación se realiza a través de una conexión SSH, no añade más complejidad a los sistemas.

5. Metodología de investigación

5.1. Metodología de la investigación

La metodología de investigación es un marco sistemático que se utiliza para resolver el problema de investigación mediante el uso de métodos para llevar a cabo la investigación, al tiempo que se alinea con la finalidad y los objetivos de su investigación (QuestionPro, s.f.).

En el contexto de esta investigación se decidió utilizar la metodología cualitativa debido a su capacidad para proporcionar una comprensión profunda de la temática que se investiga, ya que su enfoque se orientó en descubrir la percepción de las personas involucradas y evaluar la factibilidad del prototipo.

5.1.1. *Enfoque cualitativo utilizado.*

El investigador cualitativo suele rechazar deliberadamente la formulación de teorías antes de empezar a trabajar sobre el terreno, por considerar que podría inhibir su capacidad de comprender el punto de vista del sujeto estudiado, que podría cerrarle horizontes a priori. (Corbetta, 2007, p. 24).

Dentro del marco de la investigación sobre la creación de un prototipo de despliegue de OpenStack orientado a una región con una infraestructura centralizada, el enfoque cualitativo permitió comprender el contexto en el que se implementó el prototipo, además de conocer y entender de primera mano las necesidades con las que contaba la Escuela de Sistemas Informáticos por medio de diálogo con docentes de la facultad de Sistemas Informáticos.

Por lo cual un aspecto importante de la investigación era conocer desde el punto de vista de los docentes de la Escuela de Sistemas Informáticos la percepción que tenían ellos acerca de la implementación del prototipo de regiones en OpenStack, su opinión acerca de las necesidades que se tienen en la Escuela, como esta solución solventó problemas en el aspecto tecnológico, además de conocer la disposición de los docentes que lo requieran en capacitarse

en Openstack y la forma en que la Escuela administraría la solución una vez implementado el prototipo.

Por medio de la metodología cualitativa era necesario conocer los aspectos anteriores con el objetivo de elaborar un prototipo que no solamente fuera funcional, sino que además resolverá un problema y dotará de nuevas herramientas a la Escuela de Ingeniería de Sistemas Informáticos.

5.2. Población y Muestra

Según Corbetta (2007) afirma “el muestreo es el procedimiento por el cual, de un conjunto de unidades que forman el objeto de estudio (la población), se elige un número reducido de unidades (muestra) aplicando unos criterios tales que permitan generalizar los resultados obtenidos del estudio de la muestra a toda la población” (p. 272).

Por lo tanto para el desarrollo de la investigación se determinó que la población de interés está compuesta por catedráticos de los distintos departamentos y directores de la Escuela de Sistemas de la facultad de Ingeniería y Arquitectura de la Universidad de El Salvador, ya que ellos desempeñan un papel fundamental en la toma de decisiones y la implementación de una infraestructura de región en OpenStack, donde la facultad de ingeniería actúa como la infraestructura principal y las distintas escuelas se configuran como regiones separadas.

La muestra seleccionada para este estudio incluyó al director de la escuela de Ingeniería de Sistemas Informáticos, se eligió al director como representante y participante clave en la implementación de la solución de infraestructura en OpenStack. Su conocimiento y experiencia en el campo fueron esenciales para comprender los aspectos técnicos y las necesidades específicas de esta implementación. También se seleccionó a un docente del departamento de Programación y Manejo de Datos, se eligió al docente como participante clave ya que

proporcionó valiosa información sobre la infraestructura en la nube y las necesidades específicas del departamento.

Aclarar que para esta investigación no se tomó en cuenta la población de estudiantes de la facultad de Ingeniería en Sistemas Informáticos debido a que no todos poseen conocimientos previos en Openstack. Tampoco a los directores y docentes del resto escuelas de la Facultad de Ingeniería y Arquitectura debido a que el alcance del caso de estudio se centró en la Escuela de Ingeniería de Sistemas Informáticos.

5.3. Técnica de recopilación de datos utilizada

A continuación, se detalla la técnica de recolección de datos que se utilizó a través del enfoque cualitativo:

5.3.1. *Entrevista estructurada.*

“En este tipo de entrevistas se hacen las mismas preguntas a todos los entrevistados con la misma formulación y en el mismo orden. El estímulo es, por tanto, igual para todos los entrevistados. Estos, sin embargo, tienen plena libertad para responder como deseen.” (Corbetta, 2007, p. 350)

Para la investigación se ha abordado a partir de la entrevista estructurada debido a al tipo de preguntas que se utilizaron eran las mismas para todos los entrevistados ya que se quería obtener información de primera mano acerca de su punto de vista individual, se identificaron dos actores para este caso, el primero es el director de la Escuela de Ingeniería de Sistemas Informáticos el Ing. Rudy Chicas y el segundo es un docente del departamento de Programación y Manejo de Datos el ing. Cesar Augusto, los cuales por medio de la entrevista estructurada se conoció su punto de vista particular acerca de aspectos relacionados: factibilidad de implementación del prototipo en la Escuela de Sistemas Informáticos, capacidad

y disposición del personal, desafíos y soluciones que aportaría la implementación del prototipo, así como ventajas, beneficios y su visión a largo plazo una vez implementado el proyecto.

5.4. Análisis y Resultados

Durante las entrevistas realizadas, se obtuvieron percepciones valiosas sobre la viabilidad y relevancia de implementar una infraestructura de nube en la Escuela de Ingeniería de Sistemas Informáticos. A continuación, se presenta un resumen de los hallazgos:

Entrevista con el director:

- 1. Conocimiento sobre OpenStack:** Se observa experiencia previa en el uso de OpenStack por parte del entrevistado, con la implementación de un clúster en la escuela en el pasado.
- 2. Importancia de la Infraestructura de Nube:** Se destaca la relevancia de la infraestructura de nube para fines académicos e investigativos en la escuela.
- 3. Capacidad Técnica del Personal:** Se expresó confianza en la capacidad técnica del personal para aprender y utilizar tecnologías de nube, aunque se reconocen variaciones según las áreas de especialización.
- 4. Desafíos Potenciales:** Se identifican desafíos potenciales, como la necesidad de infraestructura física, la formación de un equipo de trabajo y la inversión de tiempo y recursos.
- 5. Visión a Largo Plazo:** La implementación se percibe como una mejora en el acceso tecnológico para estudiantes, aunque no se espera un impacto trascendental en el futuro cercano.

Entrevista con el docente:

- 1. Conocimiento sobre OpenStack:** El entrevistado tiene un conocimiento teórico sobre OpenStack, con experiencias pasadas relacionadas con la tecnología.

2. **Importancia de la Infraestructura de Nube:** Se reconoce la importancia de implementar una infraestructura de nube para proporcionar recursos adecuados a los estudiantes.
3. **Capacidad Técnica del Personal:** Se indica la disposición de los docentes para profundizar en el conocimiento de la tecnología de nube, pero se señalan variaciones en el nivel de conocimiento.
4. **Desafíos Potenciales:** Se mencionan desafíos como la factibilidad económica, la necesidad de automatizar la administración de recursos y garantizar el acceso a servicios críticos.
5. **Visión a Largo Plazo:** Se comparte una visión en la que el proyecto podría mejorar la eficiencia y escalabilidad de los recursos en la escuela, beneficiando la educación de los estudiantes.

6. Factibilidad

6.1. Factibilidad técnica

Según (EDU.LAT, s.f.) la factibilidad técnica determina si se dispone de los conocimientos, habilidades, equipos o herramientas necesarios para llevar a cabo los procedimientos, funciones o métodos involucrados en un proyecto.

La importancia de la factibilidad técnica

Realizar este tipo de estudios es esencial. En primer lugar, porque con ellos se averigua si se puede llevar a cabo un proyecto con los recursos técnicos disponibles. En segundo término, porque en caso de tener que adquirir otros, se pueden saber cuántos, cuándo y cuál podrá ser su coste. De esta forma, se podrán cuantificar a través de la factibilidad financiera (Economipedia, 2020).

La factibilidad técnica es un aspecto crucial al considerar la adopción de servicios en la nube, los servicios en la nube ofrecen una serie de ventajas, como escalabilidad, flexibilidad y acceso a tecnologías avanzadas, pero no todas las organizaciones o aplicaciones son adecuadas para migrar a la nube.

6.1.1. Comparación de soluciones de nube.

Tabla 1

Principales componentes de los diferentes proveedores de nube

Servicio	OpenStack	Microsoft Azure	Google Cloud Platform	Amazon Web Service	Open Nebula
Servicio de computo	Nova	Azure Virtual Machines	Google Compute Engine	Amazon EC2	Servicio de Gestión de Máquinas Virtuales
Servicio de identidad	Keystone	Azure Active Directory	Google Identity and Access Management	AWS Identity and Access Management	Integración de Identidad
Servicio de imágenes	Glance	Azure Marketplace	Google Cloud Marketplace	AWS Marketplace	Gestión de Imágenes de Máquinas Virtuales
Panel de control (Dashboard)	Horizon	Azure Portal	Google Cloud Console	AWS Management Console	Interfaz de Administración
Observaciones	OpenStack es una plataforma de código abierto que proporciona una gama completa de servicios para la nube privada e híbrida.	Microsoft Azure es una plataforma en la nube ofrecida por Microsoft, con un énfasis en la integración con el ecosistema de Microsoft.	Google Cloud Platform ofrece una amplia variedad de servicios de nube, enfocándose en la escalabilidad y la innovación.	Amazon Web Services es uno de los proveedores de nube más grandes y ofrece una amplia gama de servicios en la nube.	OpenNebula es una plataforma de nube de código abierto que se centra en la gestión y orquestación de máquinas virtuales en entornos de nube privada e híbrida.

La tabla 1 muestra una comparativa entre soluciones de nube en base a los servicios que brinda cada uno.

OpenStack es una opción sólida a considerar, especialmente si se valora la flexibilidad y el control que ofrece una plataforma de código abierto. Ofrece una amplia gama de servicios esenciales para la gestión de recursos en la nube, desde cómputo y almacenamiento hasta orquestación y telemetría. Además, OpenStack es conocido por su capacidad para implementar nubes privadas e híbridas, lo que brinda un alto grado de personalización y adaptabilidad a tus necesidades específicas. Estos son algunos puntos claves para respaldar la decisión:

- 1. Control y Personalización:** OpenStack es conocido por proporcionar un alto grado de control y personalización. Se pueden ajustar y adaptar la plataforma según necesidades específicas, lo que es especialmente beneficioso si se tienen requisitos técnicos específicos o se necesita una nube altamente personalizada.
- 2. Código Abierto:** OpenStack es una plataforma de código abierto respaldada por una comunidad activa de desarrolladores y usuarios. Esto significa que se tiene acceso al código fuente y se puede contribuir o personalizar según las necesidades, lo que aporta una mayor transparencia y flexibilidad.
- 3. Nubes Privadas e Híbridas:** OpenStack es conocido por su capacidad para implementar nubes privadas e híbridas. Si una organización requiere mantener el control sobre la infraestructura y los datos, OpenStack ofrece una opción viable.
- 4. Costos:** En términos de costos, OpenStack puede ser una opción más accesible en comparación con algunos de los proveedores de nube más grandes. Al ser de código abierto, no hay costos de licencia asociados, aunque se deben considerar los costos operativos y de soporte.
- 5. Compatibilidad Multiplataforma:** OpenStack se ha diseñado para ser compatible con una variedad de hipervisores, sistemas operativos y tecnologías de almacenamiento, lo que facilita la integración con la infraestructura existente.

- 6. Comunidad Activa:** La comunidad de OpenStack es activa y colaborativa. Puede ser beneficioso en cuanto a recursos, documentación y la experiencia de otros usuarios en la comunidad.

6.1.2. Soluciones de almacenamiento.

6.1.2.1. MinIO.

MinIO es un servidor de almacenamiento de objetos de código abierto que se integra eficazmente con OpenStack. Ofrece almacenamiento de objetos altamente escalable y es compatible con la API S3 de Amazon, lo que lo convierte en una solución atractiva para el almacenamiento de objetos en la nube de OpenStack.

- 1. Escalabilidad:** MinIO es altamente escalable y se especializa en almacenamiento de objetos, lo que lo hace adecuado para entornos con necesidades de almacenamiento masivo.
- 2. Rendimiento:** Ofrece un rendimiento excelente para almacenamiento de objetos y es compatible con la API S3 de Amazon.
- 3. Costos:** Es asequible debido a su enfoque en código abierto y opciones de implementación sencilla.
- 4. Facilidad de Gestión:** Su implementación y gestión son sencillas y bien documentadas.

6.1.2.2. HekaFS (LizardFS).

HekaFS, también conocido como LizardFS, es un sistema de archivos distribuido que brinda opciones de almacenamiento distribuido en OpenStack. Es una alternativa para gestionar datos en clústeres y puede ser útil en entornos que requieren una alta disponibilidad y redundancia.

1. **Escalabilidad:** HekaFS es escalable y adecuado para entornos que requieren alta confiabilidad y disponibilidad de datos.
2. **Rendimiento:** Ofrece un rendimiento sólido y es capaz de manejar grandes cantidades de datos.
3. **Costos:** Los costos pueden variar, pero puede ser una opción rentable para entornos con necesidades específicas.
4. **Facilidad de Gestión:** Requiere configuración y gestión adecuadas, pero proporciona almacenamiento distribuido.

6.1.2.3. MooseFs.

MooseFS es otro sistema de archivos distribuido que puede ser considerado en entornos OpenStack. Proporciona capacidades de almacenamiento escalable y puede gestionar grandes cantidades de datos en la nube de OpenStack.

1. **Escalabilidad:** MooseFS es escalable y puede gestionar grandes volúmenes de datos en entornos OpenStack.
2. **Rendimiento:** Ofrece un rendimiento sólido y es adecuado para aplicaciones que requieren almacenamiento distribuido.
3. **Costos:** Los costos pueden variar, pero es una opción a considerar para entornos que necesitan escalabilidad.
4. **Facilidad de Gestión:** Requiere configuración y gestión adecuadas, pero proporciona almacenamiento distribuido.

6.1.3. *Comparativa de soluciones de almacenamiento.*

Tabla 2

Principales soluciones de almacenamiento, incluye aquellas descritas en el marco teórico.

Característica	Ceph	NFS (Network File System)	GlusterFS	MinIO	BeeGFS	HekaFS (LizardFS)	MooseFS
Escalabilidad	Altamente escalable y confiable.	Escalabilidad limitada.	Escalable y adecuado para almacenamiento distribuido.	Altamente escalable para almacenamiento de objetos.	Altamente escalable y adecuado para aplicaciones intensivas en datos.	Escalable y adecuado para entornos con alta disponibilidad.	Escalable y capaz de gestionar grandes volúmenes de datos.
Rendimiento	Buen rendimiento, especialmente en configuraciones en clúster.	Rendimiento aceptable para aplicaciones de menor exigencia.	Ofrece un rendimiento decente y escalabilidad.	Excelente rendimiento para almacenamiento de objetos.	Ofrece un rendimiento excepcional para aplicaciones intensivas en datos.	Ofrece un rendimiento sólido y alta disponibilidad.	Ofrece un rendimiento sólido y capacidad de gestión de grandes cantidades de datos.
Costos	Puede ser rentable debido a su uso de hardware estándar y software de código abierto.	Costos generales bajos en términos de hardware y configuración.	Los costos pueden ser moderados y dependen de la escala y configuración.	Asequible debido a su enfoque en código abierto y opciones de implementación sencilla.	Costos moderados, justificados para aplicaciones de alto rendimiento.	Puede ser una opción rentable para entornos con necesidades específicas.	Costos variados, pero es una opción a considerar para entornos que necesitan escalabilidad.
Facilidad de Gestión	Requiere una configuración inicial	Implementación y gestión sencillas y bien	Requiere atención en la implementación	Implementación y gestión sencillas y bien	Requiere una configuración inicial	Requiere configuración y gestión adecuadas, pero	Requiere configuración y gestión adecuadas,

más compleja, pero la gestión diaria es relativam ente sencilla.	conocida s.	inicial, pero la gestión es eficiente.	documen tadas.	más detallada , pero proporci ona alto rendimie nto.	proporciona almacenamie nto distribuido.	pero proporcion a almacena miento distribuido .
---	----------------	--	-------------------	--	---	---

La tabla 2 muestra una comparativa de soluciones de almacenamiento a base a sus características

Ceph emerge como la elección preferida para satisfacer las necesidades de almacenamiento en el entorno OpenStack. Su alta escalabilidad y confiabilidad lo convierten en una solución robusta, capaz de manejar grandes volúmenes de datos de manera eficiente. Además, Ceph ofrece un buen rendimiento, especialmente cuando se implementa en configuraciones en clúster, lo que garantiza un acceso rápido y confiable a los datos.

En términos de costos, Ceph se destaca debido a su enfoque en el uso de hardware estándar y software de código abierto, lo que resulta en una inversión asequible a largo plazo. La facilidad de gestión también es un factor a su favor, ya que, aunque requiere una configuración inicial más compleja, la gestión diaria es relativamente sencilla.

Por último, la amplia compatibilidad de Ceph con OpenStack lo convierte en una opción sólida para entornos cloud, lo que facilita su integración y uso eficiente. En resumen, Ceph ofrece una combinación de escalabilidad, rendimiento, costo efectividad y compatibilidad que lo posiciona como la solución ideal para satisfacer las necesidades de almacenamiento en el entorno OpenStack.

6.2. Factibilidad económica

El análisis de factibilidad no se limita a una única empresa, sino que se enfoca en brindar una solución adaptable a cualquier organización o entidad que desee implementarla. La propuesta se centró en evaluar la viabilidad de la implementación de servicios de nube

personalizados, permitiendo a las organizaciones aprovechar al máximo las ventajas de la nube, ajustando los costos y beneficios a sus necesidades específicas.

En el análisis de factibilidad de la implementación de servicios de nube personalizados, se consideraron diversos factores que pueden variar significativamente de una organización a otra. Estos factores incluyen, pero no se limitan a:

- 1. Especificaciones y número de servidores:** La elección de la cantidad y potencia de los servidores se adapta a las necesidades específicas de la organización.
- 2. Volumen de datos:** Se evalúa el almacenamiento y transferencia de datos, ajustándose a las demandas particulares de cada entidad.
- 3. Sistema operativo y software:** Se permite la selección de sistemas operativos y software acorde a los requerimientos de la organización.
- 4. Modelo de pago:** Se ofrece flexibilidad en cuanto al modelo de pago, ya sea por minuto, hora, mes, o mediante opciones como el pago por uso, instancias reservadas o contratos a largo plazo.
- 5. Ubicación del centro de datos:** Se considera la ubicación del centro de datos de nube, permitiendo a las organizaciones elegir la ubicación que mejor se ajuste a sus necesidades.

Para brindar una perspectiva completa y personalizable, se compararon los principales proveedores de servicios de nube, como AWS, Azure y Google Cloud Platform, en función de las necesidades y preferencias de la organización. El análisis se enfocó en un despliegue combinado de alojamiento en nube público de servidores y almacenamiento.

Las calculadoras de precios proporcionadas por los principales proveedores de servicios en la nube, Amazon Web Services (AWS), Microsoft Azure y Google Cloud Platform (GCP), fueron herramientas esenciales para realizar cálculos precisos en el análisis de factibilidad. Estas herramientas permitieron estimar los costos asociados con la implementación de servicios

en la nube, tomando en cuenta las necesidades y preferencias específicas de la organización.

Estas herramientas tienen el nombre de:

1. AWS Pricing Calculator
2. Azure Pricing Calculator
3. Google Cloud Pricing Calculator

Utilizando estas calculadoras, se realizaron estimaciones detalladas de costos, lo que permitió tomar decisiones informadas y adaptar la implementación de servicios en la nube de acuerdo a los requerimientos y presupuesto de la organización.

Para el estudio de factibilidad se elaboró un resumen del precio para las siguientes máquinas:

Tabla 3
Infraestructura del prototipo

CPU	DISCO	RAM	SO
4	1X20GB	2048 MB	Ubuntu Server 20.04
2	1X10GB	1024 MB	FreeBSD 12.3
8	1X70GB	16384 MB	Ubuntu Server 20.04
8	1X60GB	5120 MB	Ubuntu Server 20.04
2	1X10GB	2048 MB	Ubuntu Server 20.04
4	1X15GB	1536 MB	Ubuntu Server 20.04
4	1X15GB	1024 MB	Ubuntu Server 20.04
4	1X15GB, 1X100GB	700 MB	Ubuntu Server 20.04
4	1X15GB, 1X100GB	700 MB	Ubuntu Server 20.04
4	1X15GB, 1X100GB	700 MB	Ubuntu Server 20.04

La tabla 3 muestra las especificaciones de cpu, almacenamiento, ram y sistema operativo de las máquinas virtuales para el funcionamiento del prototipo

A continuación, se presentan los gastos proyectados en las principales plataformas de nube durante un período de cinco años.

Tabla 4

Tabla de gastos proyectados en plataformas de nube públicas.

Proveedor	Año 1	Año 2	Año 3	Año 4	Año 5
Amazon Web Services	\$6,300.00	\$12,600.00	\$37,800.00	\$151,200.00	\$756,000.00
Microsoft Azure	\$5,800.00	\$11,600.00	\$34,000.00	\$139,200.00	\$696,000.00
Google Cloud Platform	\$4,900.00	\$9,800.00	\$29,400.00	\$117,600.00	\$588,000.00

La tabla 4 muestra una comparativa de los gastos por año según su plataforma

A continuación, se presenta una detallada proyección de los gastos asociados con la adquisición y mantenimiento de servidores on-premises necesarios para la puesta de la solución multiregión.

Tabla 5

Tabla de gastos proyectados de servidores on-promises

Nodo	Características	Año 1	Año 2	Año 3	Año 4	Año 5
SERVIDOR OPEN STACK2						
ControlPlane	Intel Xenon 32Gb Ram 512GB Disco	\$2,100.00	\$150.00	\$150.00	\$150.00	\$150.00
Compute1	Intel Xenon 16Gb Ram 512GB Disco	\$1,050.00	\$150.00	\$150.00	\$150.00	\$150.00
Compute2	Intel Xenon 16Gb Ram 512GB Disco	\$1,050.00	\$150.00	\$150.00	\$150.00	\$150.00
Deployment	Intel Xenon 16Gb Ram 512GB Disco	\$1,050.00	\$150.00	\$150.00	\$150.00	\$150.00
SERVIDOR OPEN STACK 2						
ControlPlane	Intel Xenon 32Gb Ram 512GB Disco	\$2,100.00	\$150.00	\$150.00	\$150.00	\$150.00
Compute1	Intel Xenon 16Gb Ram 512GB Disco	\$1,050.00	\$150.00	\$150.00	\$150.00	\$150.00

Compute2	Intel Xenon 16Gb Ram 512GB Disco	\$1,050.00	\$150.00	\$150.00	\$150.00	\$150.00
Deployment	Intel Xenon 16Gb Ram 512GB Disco	\$1,050.00	\$150.00	\$150.00	\$150.00	\$150.00
SERVIDOR CEPH 1						
Almacenamiento	Intel Xenon 500GB Ram 512GB Disco	\$1,500	\$150.00	\$150.00	\$150.00	\$150.00
SERVIDOR CEPH 2						
Almacenamiento	Intel Xenon 500GB Ram 512GB Disco	\$1,500	\$150.00	\$150.00	\$150.00	\$150.00
SERVIDOR CEPH 3						
Almacenamiento	Intel Xenon 500GB Ram 512GB Disco	\$1,500	\$150.00	\$150.00	\$150.00	\$150.00
Total		\$15,500.00	\$1,050.00	\$1,050.00	\$1,050.00	\$1,050.00

La tabla 5 muestra una comparativa del gasto por año del servidores on-promises

Esta tabla presenta una visión resumida de la inversión inicial y los costos de mantenimiento asociados a la adquisición y operación de los servidores on-promises necesarios durante un período de cinco años. A continuación, se detallan los elementos clave de la tabla:

- 1. Año 1 (Inversión Inicial):** En el primer año, se realiza una inversión inicial de \$15,500.00 para adquirir los servidores que anteriormente se detallan. Esta inversión representa el desembolso inicial requerido para obtener el hardware.
- 2. Años 2 a 5 (Costo de Mantenimiento):** A partir del segundo año (Año 2) y a lo largo de los siguientes cuatro años (Años 2 a 5), se incurre en un costo de mantenimiento anual constante de \$1,050.00. Estos costos están destinados a cubrir gastos relacionados con el mantenimiento, el soporte técnico y la operación continua de los servidores

Después de calcular los costos acumulativos, se compararon ambas opciones:

En el primer año, la inversión inicial requerida para la implementación del Prototipo de despliegue automatizado de regiones destinado a su integración en una infraestructura centralizada en OpenStack es de \$15,500.00. Esta inversión inicial resulta ser más económica en comparación con las plataformas de nube privada, lo que es un factor atractivo para proyectos específicos como el prototipo.

Sin embargo, a medida que transcurren los años, los costos de las plataformas de nube privada aumentan significativamente debido a los gastos asociados al uso y la escalabilidad en infraestructuras de este tipo.

En contraste, el prototipo mantiene costos de mantenimiento constantes de \$1,050.00 por año, lo que lo convierte en una opción más rentable a largo plazo en comparación con las plataformas de nube pública. Esta estabilidad en los costos es especialmente relevante para este tipo de proyectos.

7. Prototipo

Para la elaboración del prototipo se dispuso emplear diferentes tipos de herramientas, cada una con una función específica. Dado que la idea central del proyecto se centra en la integración de una región con una infraestructura centralizada, es necesario de que se realice un despliegue de la infraestructura con los componentes mínimos necesarios para la implementación del prototipo. Asimismo, fue considerado por parte del grupo que debía emplearse una herramienta que nos permitiera realizar un despliegue automatizado, de tal forma que fuese posible trabajar de manera más eficiente y poder centrarse en las configuraciones referentes a cómo conectar la región con la infraestructura.

Considerando que se asignó un caso de estudio para este trabajo, se tomó la decisión, para dar cumplimiento a lo solicitado, de emplear OpenStack como solución de nube para el desarrollo del prototipo. También, la elección de Ceph como solución de almacenamiento fue

la elegida para la implementación, puesto que los beneficios que estas tecnologías brindan trabajando de manera conjunta fueron las más cercanas a lo ideal, que se justifica principalmente en la facilidad de integración de los servicios de Cinder, Nova y Glance con Ceph.

7.1. Construcción del prototipo

7.1.1. Tecnologías utilizadas.

- Kernel-based Virtual Machine (KVM)
- Ubuntu Server 20.04
- OpenStack
- Ceph
- Ansible
- LXC
- PFSense
- Docker
- MariaDB
- RabbitMQ
- Memcached

7.1.2. Diseño del prototipo.

La implementación del prototipo brinda infraestructura como servicio y está enfocada en el manejo de regiones de una nube. La implementación ha sido desplegada utilizando máquinas virtuales bajo KVM con el sistema operativo Ubuntu Server 20.04, donde se utilizó la herramienta Ansible para realizar la instalación de la mayoría componentes de Openstack incluidos en el prototipo para la parte de la región, donde en el nodo Control Plane utiliza en

su mayoría contenedores LXC. Para la creación de la infraestructura centralizada donde será integrada la región, se realizó una instalación manual de los componentes mínimos necesarios.

Los componentes de OpenStack utilizados en el prototipo son los siguientes:

- Nova, que aporta en la administración y ejecución de las máquinas virtuales de la solución.
- Glance, cuya función nos facilita la administración de imágenes para las instancias que deseemos crear.
- Keystone, el cual se encarga de realizar la función de identificación y autorización dentro del prototipo.
- Placement, siendo el servicio que nos auxilia en la gestión de recursos de los nodos de cómputo disponibles en nuestra nube.
- Horizon, siendo el componente que nos proporciona el panel web de administración de la solución.
- Neutron, que nos aprovisiona las herramientas para gestionar el apartado de redes dentro del prototipo.
- Cinder, que no dota de facilidades al prototipo en la gestión de almacenamiento, puntualmente la del tipo en bloques.
- Swift, que brinda a solución el acceso a un almacenamiento de objetos.

7.1.3. Desarrollo del prototipo.

7.1.3.1. Infraestructura.

El despliegue de la infraestructura requiere de dos nodos. Sus características son las siguientes:

Tabla 6

Características de los nodos de infraestructura.

#	Nodo	CPUs	Discos	RAM	NIC	Sistema operativo
1	Controller	4	1 x 20 GiB	2048 MiB	1	Ubuntu Server 20.04
2	pfSense	2	1 x 10 GiB	1024 MiB	2	FreeBSD 12.3

La tabla 6 muestra las especificaciones de las máquinas virtuales para los nodos de infraestructura

Los componentes que se han instalado en el nodo Controller son los siguientes:

- Keystone
- MariaDB
- RabbitMQ
- Memcached
- Glance (para que exista un servicio en la región y se active el selector que nos permita movernos entre regiones)

Para la instalación de pfSense se utilizó únicamente una imagen provista por la compañía desarrolladora de la versión 2.6.0-RELEASE bajo FreeBSD 12.3-STABLE.

La configuración detallada de ambos nodos es tratada en la sección Diseño de bajo nivel (ver Anexo III).

7.1.3.2. Región.

Para la región, se han utilizado tres máquinas virtuales para el despliegue de OpenStack y cinco para Ceph. Sus características se detallan a continuación:

Tabla 7

Características de los nodos de la región.

#	Nodo	CPUs	Discos	RAM	NIC	Sistema operativo
1	Controller	4	1 x 20 GiB	2048 MiB	1	Ubuntu Server 20.04
2	pfSense	2	1 x 10 GiB	1024 MiB	2	FreeBSD 12.3
3	Control Plane	8	1 x 70 GiB	16384 MiB	3	Ubuntu Server 20.04

4	Compute	8	1 x 60 Gib	5120 MiB	2	Ubuntu Server 20.04
5	Deployment	2	1 x 10 GiB	2048 MiB	1	Ubuntu Server 20.04
6	Ceph-admin	4	1 x 15 GiB	1536 MiB	1	Ubuntu Server 20.04
7	Ceph-mon	4	1 x 15 GiB	1024 MiB	1	Ubuntu Server 20.04
8	Ceph-osd1	4	1 x 15 Gib 1 x 100 GiB	700 MiB	1	Ubuntu Server 20.04
9	Ceph-osd2	4	1 x 15 Gib 1 x 100 GiB	700 MiB	1	Ubuntu Server 20.04
10	Ceph-osd3	4	1 x 15 Gib 1 x 100 GiB	700 MiB	1	Ubuntu Server 20.04

La tabla 7 muestra las especificaciones de las máquinas virtuales para los nodos de la región

Inicialmente, el clúster de Ceph fue el primero en ser desplegado para, posteriormente, conectarlo con OpenStack. Para dicho despliegue se utilizaron cinco máquinas virtuales, donde las asignadas para el servicio OSD se agregaron discos adicionales. El método de despliegue de OpenStack-Ansible fue realizado mediante las máquinas virtuales Deployment, Control Plane y Compute.

Cabe aclarar que el nodo adicional Deployment para trabajar el método de OpenStack-Ansible, solamente sirve como un orquestador al momento del despliegue y, una vez se finaliza la instalación, ya no es necesario de ninguna manera para la operación de la nube, independientemente la solución sea integrada a una infraestructura centralizada o no.

Toda la configuración necesaria para el despliegue de la región se encuentra detallada en la sección Diseño de bajo nivel (ver Anexo III).

7.1.3.3. Redes empleadas.

Las redes utilizadas fueron las siguientes:

Tabla 8
Redes empleadas en el despliegue del prototipo.

Nombre	Red	Tipo	ID
Management	172.29.236.0/22	VLAN	10
Tunnel	172.29.240.0/22	VLAN	30
Storage	172.29.244.0/22	VLAN	20
VLAN (vlan provider)	172.29.248.0/22	VLAN	40
NAT (flat provider)	192.168.122.0/24	FLAT	-
LAN	192.168.1.0/24	FLAT	-

La tabla 8 muestra las redes con su nombre, tipo e id necesarias para el despliegue del prototipo

La red LAN es utilizada simplemente como un método de acceso a las máquinas virtuales, por tanto, no es realmente obligatoria su configuración, puesto que el host físico puede tener una dirección de la vlan de management y conectarse a través de ella. De ser utilizada, no es de relevancia la dirección que se le asigne a las máquinas virtuales, a excepción de la empleada para el nodo Control Plane. Además, cabe la aclaración de que los contenedores dentro de dicho nodo toman direcciones de las redes anteriormente mencionadas.

Es importante aclarar que la red Storage es el medio por el cual OpenStack tiene interacción con Ceph, dado que los servicios monitores de la solución de almacenamiento han sido desplegado en dicha red.

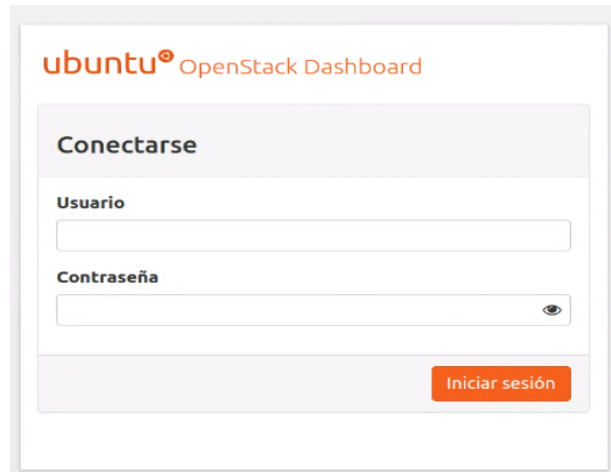
La configuración necesaria referente a las interfaces, direcciones y redes empleadas de todas las máquinas virtuales se encuentra detallada en la sección Diseño de bajo nivel (ver Anexo III).

7.2. Prueba del prototipo

7.2.1. Funcionamiento de OpenStack.

Login

En la Figura 4, se observa el formulario de Login, donde se nos consulta por el usuario y la contraseña.



The image shows a login form titled "Conectarse" (Connect) within the "ubuntu® OpenStack Dashboard". It features two input fields: "Usuario" (Username) and "Contraseña" (Password), with a toggle icon for password visibility. An orange "Iniciar sesión" (Log in) button is positioned at the bottom right of the form.

Figura 4: Formulario de login del panel web de OpenStack Horizon.

Panel gráfico

En la siguiente figura, se aprecia la pantalla posterior a iniciar sesión donde se muestran diferentes gráficos impulsados por la API del servicio Placement.

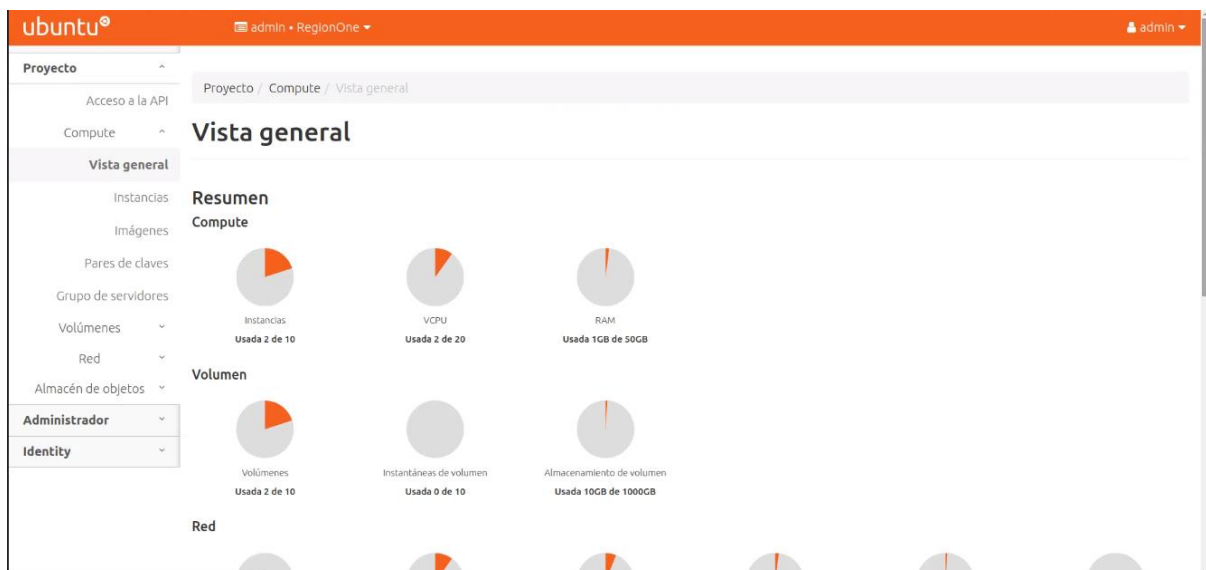


Figura 5: Vista inicial del panel web de OpenStack, donde se muestra información referente a los recursos utilizados por la nube.

Redes

En la Figura 6, se observa la pantalla referente a las redes de la nube. Es en esta sección donde se permite administrar todo lo relacionado a ellas, como la creación, modificación y eliminación.

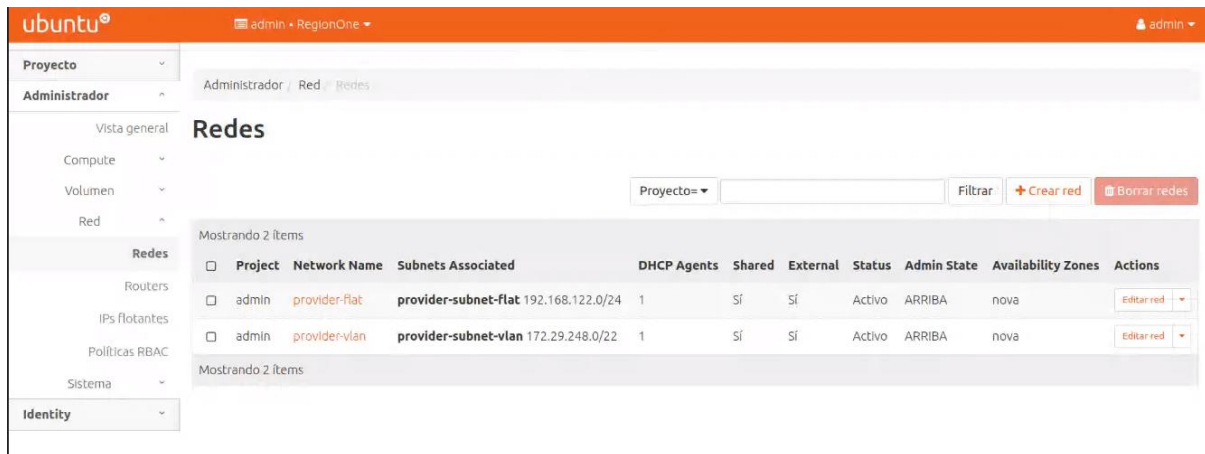


Figura 6: Vista de la pantalla de redes dentro de OpenStack.

Sabores

En la Figura 7 se observa la pantalla de administración de Sabores de la nube. Es donde se permite crear, editar y eliminar las diferentes opciones que se tienen a disposición para la creación de instancias.

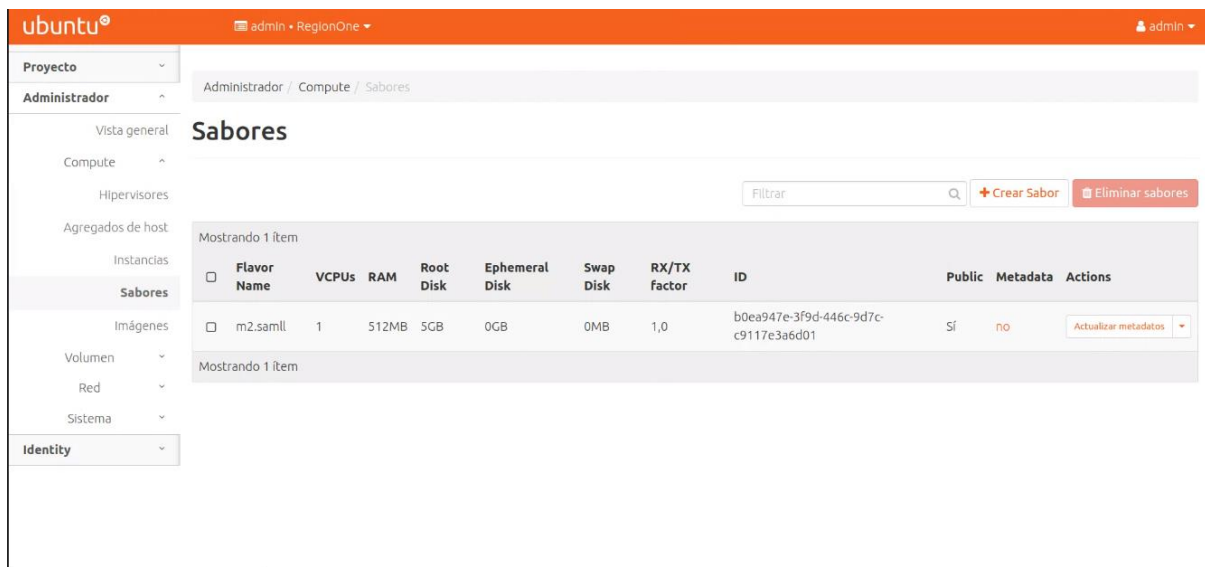
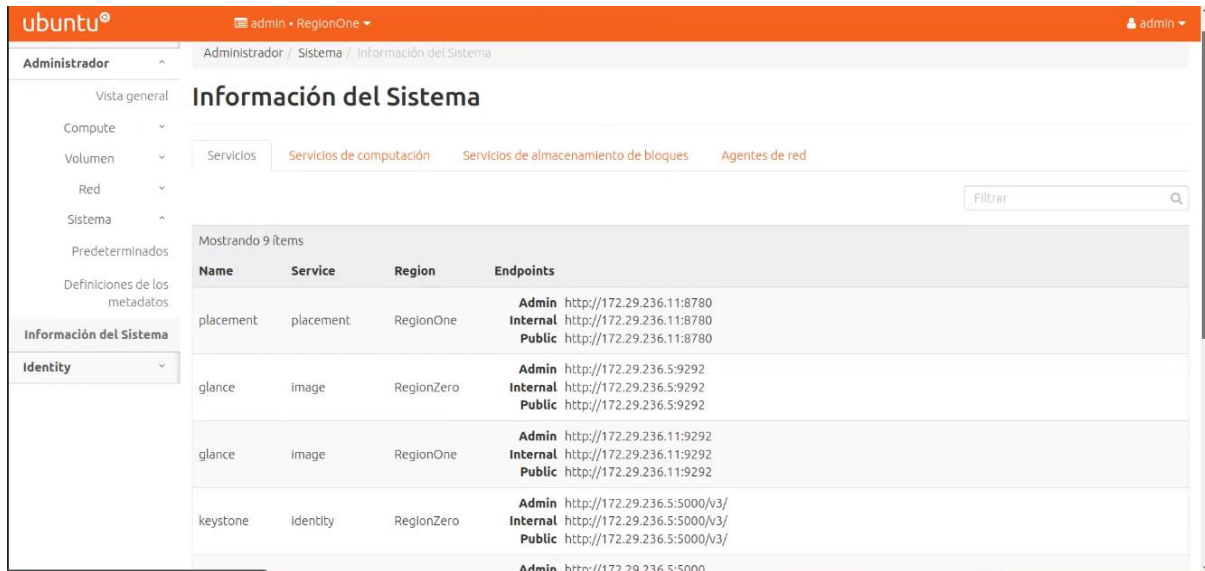


Figura 7: Pantalla de Sabores dentro de OpenStack.

Listado de endpoints

En la siguiente figura se muestra el listado de endpoints que consume la infraestructura centralizada. Por cada servicio correspondiente a una región, es agregada una nueva entrada al listado donde se detalla el tipo de servicio y la región a la que pertenece.

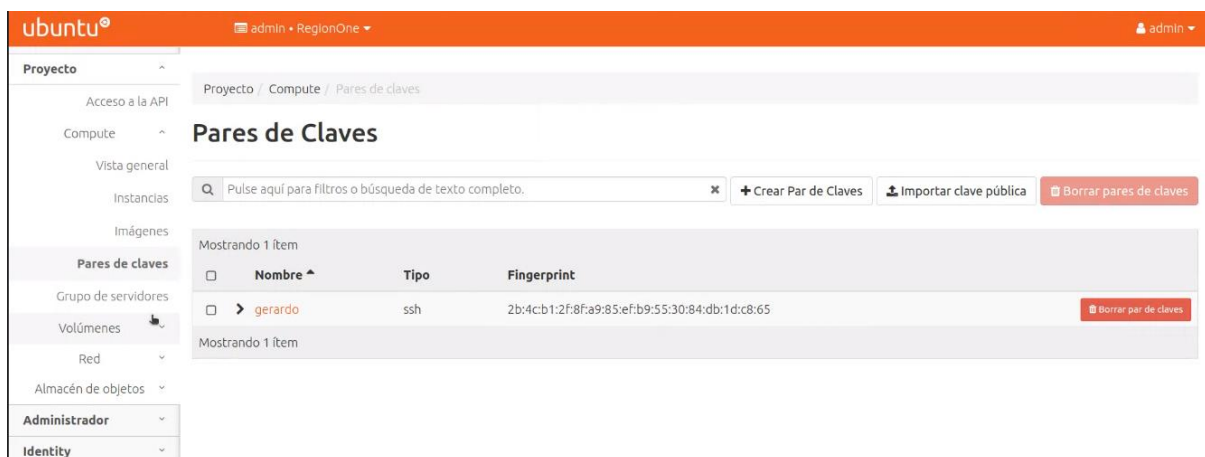


Name	Service	Region	Endpoints
placement	placement	RegionOne	Admin http://172.29.236.11:8780
			Internal http://172.29.236.11:8780
			Public http://172.29.236.11:8780
glance	image	RegionZero	Admin http://172.29.236.5:9292
			Internal http://172.29.236.5:9292
			Public http://172.29.236.5:9292
glance	image	RegionOne	Admin http://172.29.236.11:9292
			Internal http://172.29.236.11:9292
			Public http://172.29.236.11:9292
keystone	identity	RegionZero	Admin http://172.29.236.5:5000/v3/
			Internal http://172.29.236.5:5000/v3/
			Public http://172.29.236.5:5000/v3/
			Admin http://172.29.236.5:5000

Figura 8: Listado de los endpoints de cada componente consumidos por OpenStack.

Llaves SSH

En la Figura 9 se detalla la pantalla de administración de llaves SSH.



Nombre	Tipo	Fingerprint
gerardo	ssh	2b:4cb1:2f:8fa9:85:ef:b9:55:30:84:db:1d:c8:65

Figura 9: Vista de la pantalla de administración de llaves SSH en OpenStack.

Instancias

En la siguiente figura se muestra la pantalla de gestión de todas las instancias de OpenStack.

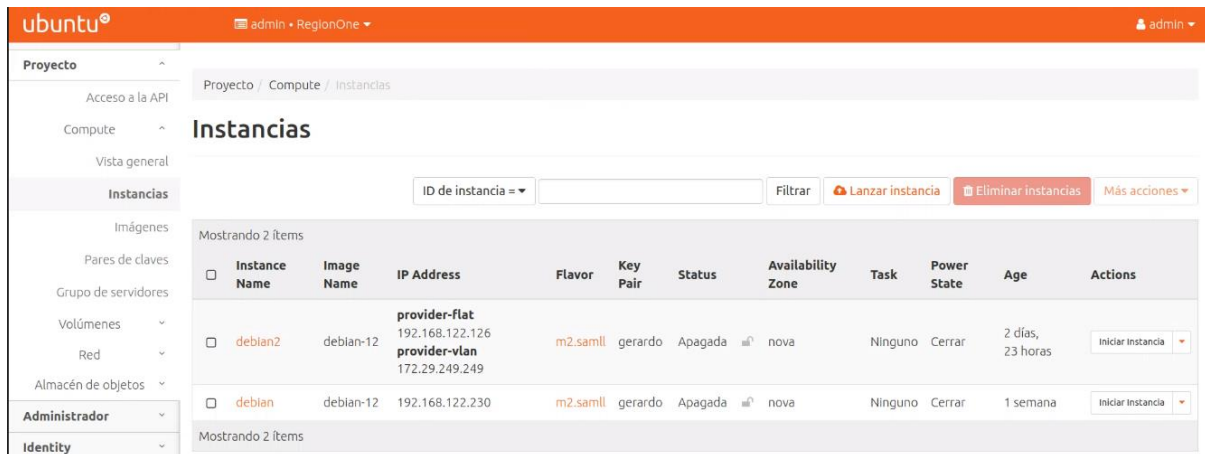


Figura 10: Vista de la pantalla de instancias dentro de OpenStack.

Volúmenes

En la Figura 11, se aprecia la pantalla de gestión de volúmenes empleados por las instancias de la nube.

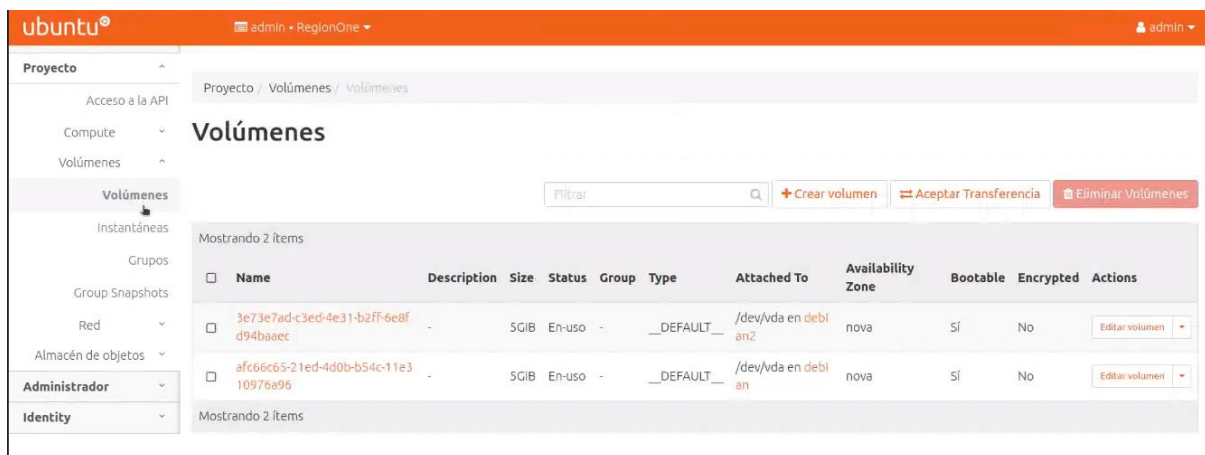


Figura 11: Vista de la pantalla de volúmenes dentro de OpenStack.

Contenedor de objetos

En la siguiente figura se muestra la pantalla de administración de un contenedor de almacenamiento de objetos dentro de OpenStack. Además, se listan los objetos almacenados dentro de dicho contenedor.

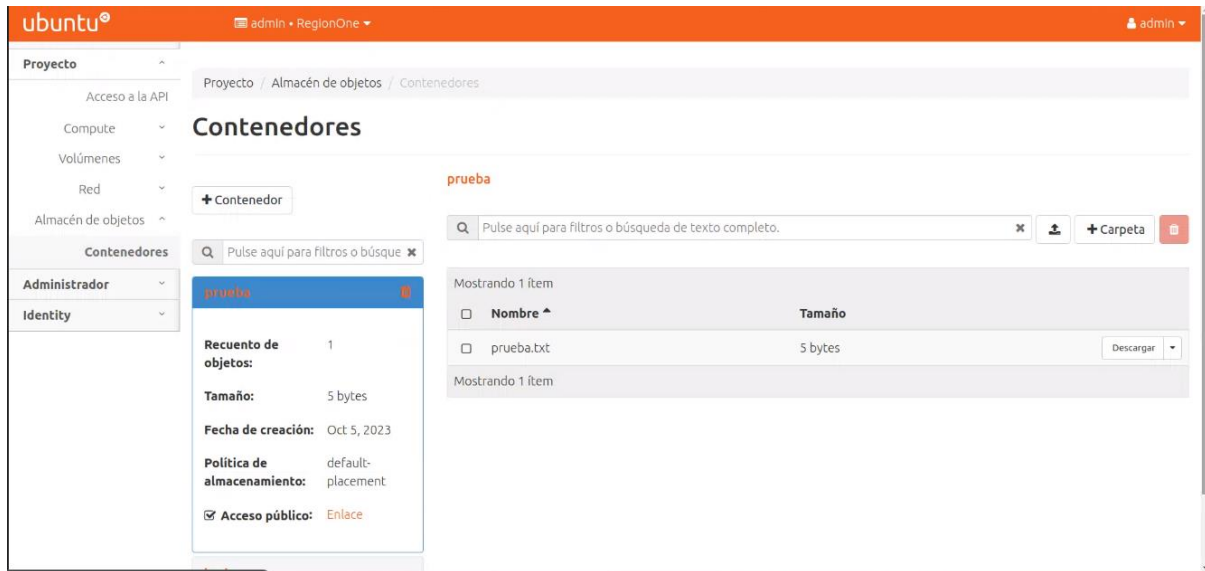


Figura 12: Pantalla de administración de un contenedor dentro OpenStack.

7.2.2. *Funcionamiento de Ceph.*

Login

En la Figura 13, se puede observar el formulario de inicio de sesión brindado por la herramienta Ceph.



Figura 13: Formulario de Login de Ceph.

Panel gráfico

En la siguiente figura, se muestra el panel en su estado inicial luego de ingresar exitosamente a la plataforma. Se despliega información relevante al usuario como la cantidad de hosts que conforman el clúster, si se encuentran ejecutando los servicios monitores, la capacidad del almacenamiento, entre otros.

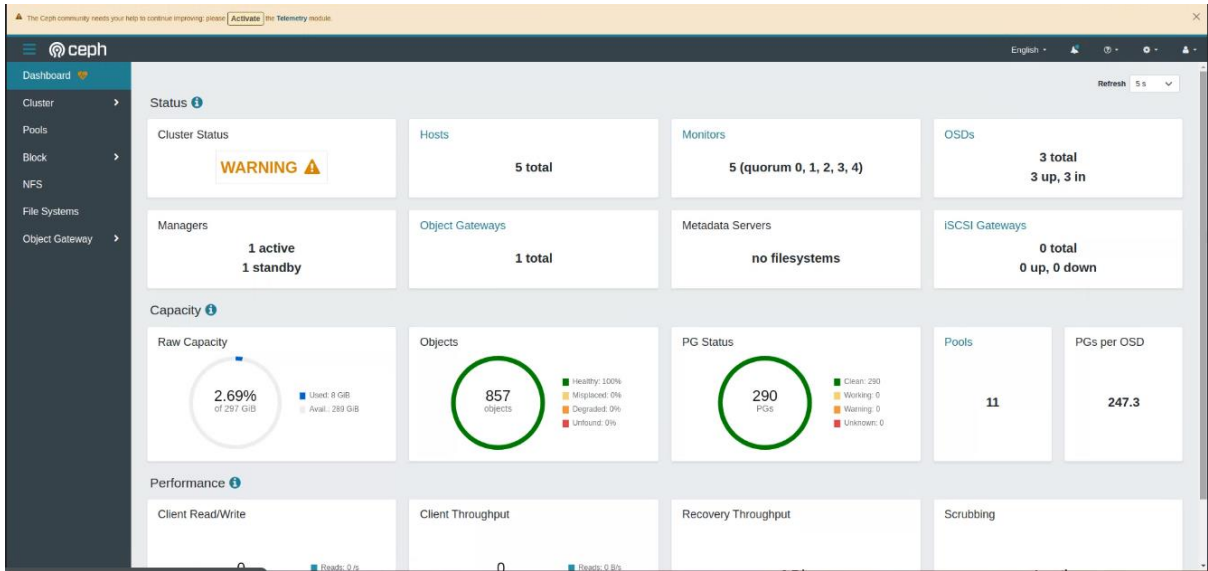


Figura 14: Vista general del panel web de Ceph.

OSD

En la Figura 15 se detalla la pantalla de los servicios OSD que posee el clúster de Ceph. Se pueden observar métricas referentes a dicho servicio.

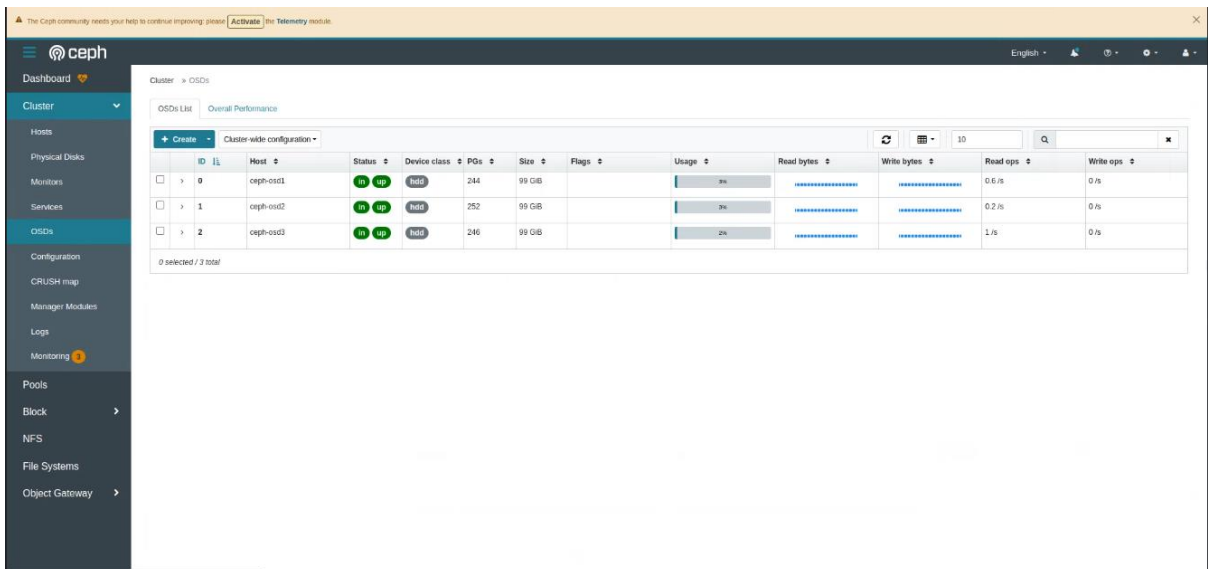
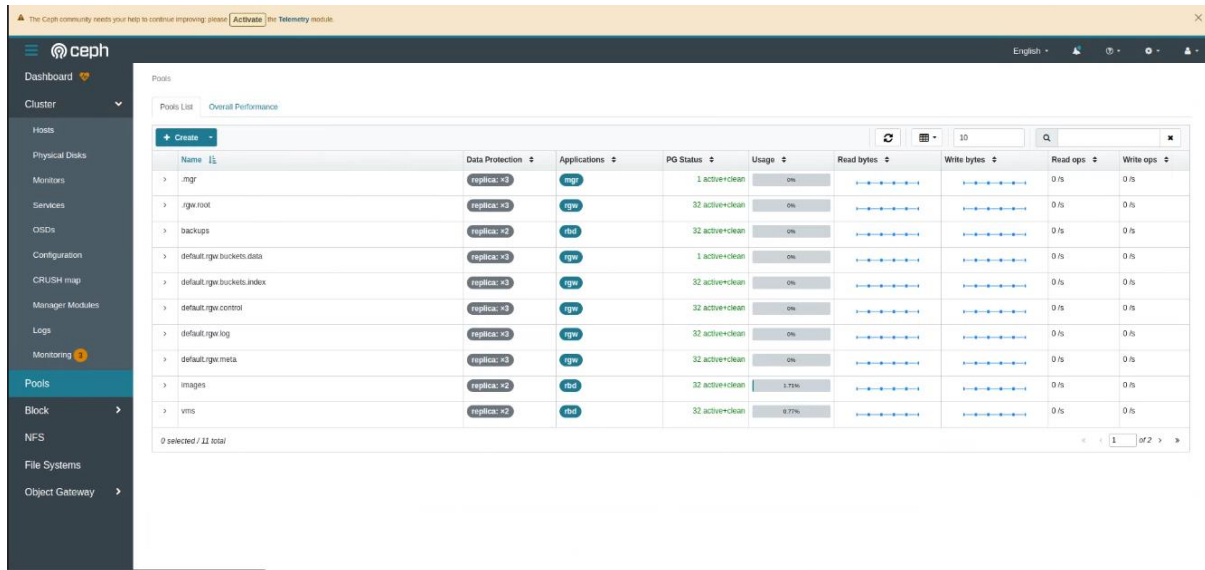


Figura 15: Listado de OSDs dentro de Ceph.

Pools

En la Figura 16 se puede observar la pantalla dedicada a presentar el listado de pools del clúster.

Además, se indican datos relevantes como la réplica que se lleva a cabo para cada pool.



Name	Data Protection	Applications	PG Status	Usage	Read bytes	Write bytes	Read ops	Write ops
mgr	replica:2	mgr	1 active+clean	0%			0/s	0/s
rgw.root	replica:3	rgw	32 active+clean	0%			0/s	0/s
backups	replica:3	rbd	32 active+clean	0%			0/s	0/s
default.rgw.buckets.data	replica:3	rgw	1 active+clean	0%			0/s	0/s
default.rgw.buckets.index	replica:3	rgw	32 active+clean	0%			0/s	0/s
default.rgw.control	replica:3	rgw	32 active+clean	0%			0/s	0/s
default.rgw.log	replica:3	rgw	32 active+clean	0%			0/s	0/s
default.rgw.meta	replica:3	rgw	32 active+clean	0%			0/s	0/s
images	replica:2	rbd	32 active+clean	1.7%			0/s	0/s
vmfs	replica:2	rbd	32 active+clean	0.7%			0/s	0/s

Figura 16: Listado de Pools dentro de Ceph.

Images

En la siguiente figura se muestra el listado de imágenes dentro del clúster. El panel detalla el tipo de servicio que está haciendo uso de la imagen. En el caso de la interacción entre las imágenes y volúmenes de OpenStack, con sus respectivos pools en Ceph se aprecia que la imagen es padre del volumen, puesto que dicho volumen es generado a partir de la imagen subida por el servicio Glance.

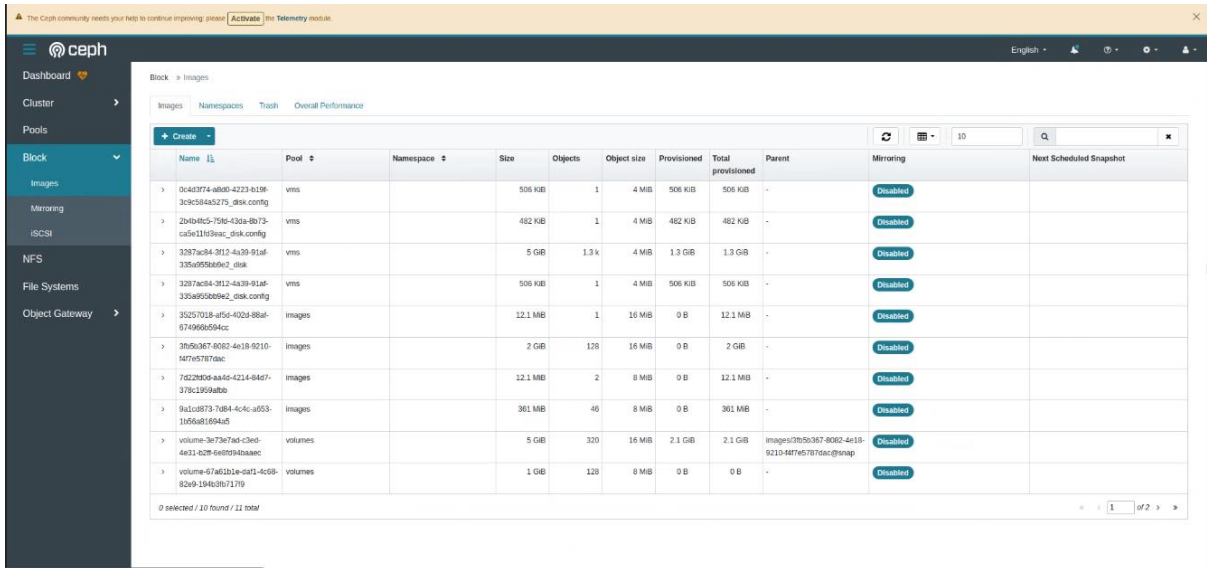


Figura 17: Listado de Imágenes dentro de Ceph.

Buckets (RGW)

En la Figura 18 se detallan los contenedores que han sido creados para el servicio de almacenamiento de objetos del cluster.

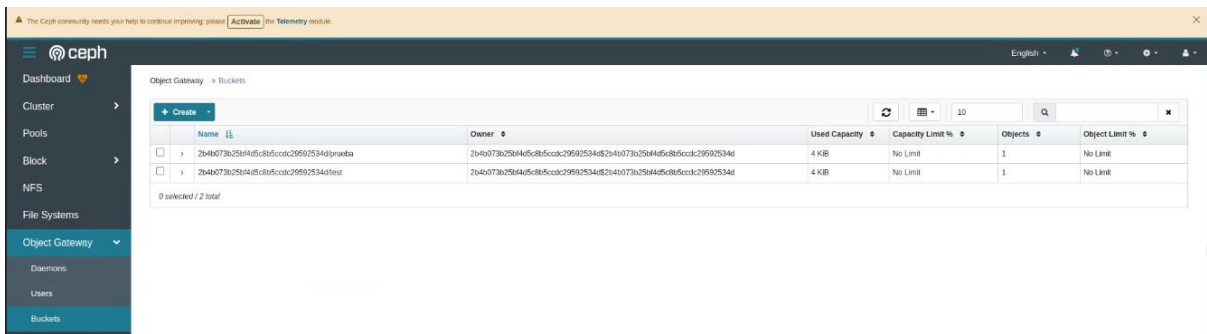


Figura 18: Listado de contenedores de objetos dentro de Ceph.

Users (RGW)

En la Figura 19 se detallan el usuario generado por Keystone para el servicio de almacenamiento de objetos a través de contenedores.

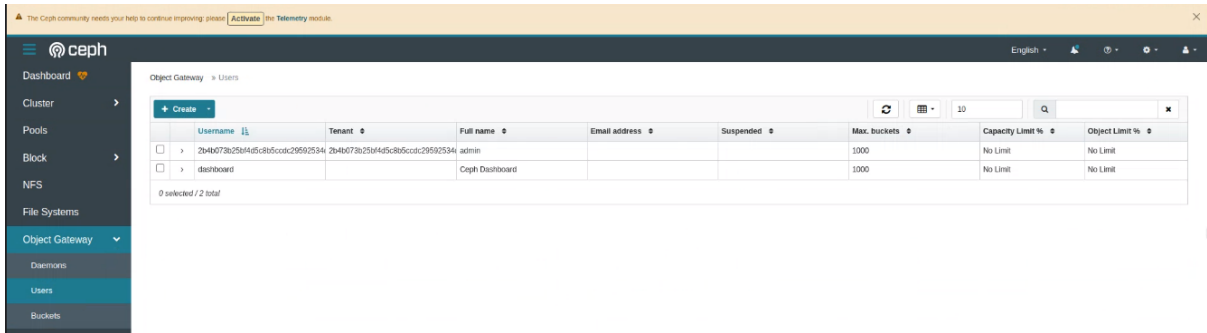


Figura 19: Listado de usuarios de contenedores dentro de Ceph.

Daemons (RGW)

En la siguiente figura se listan los servicios de gateway desplegados para el almacenamiento de objetos de Ceph.

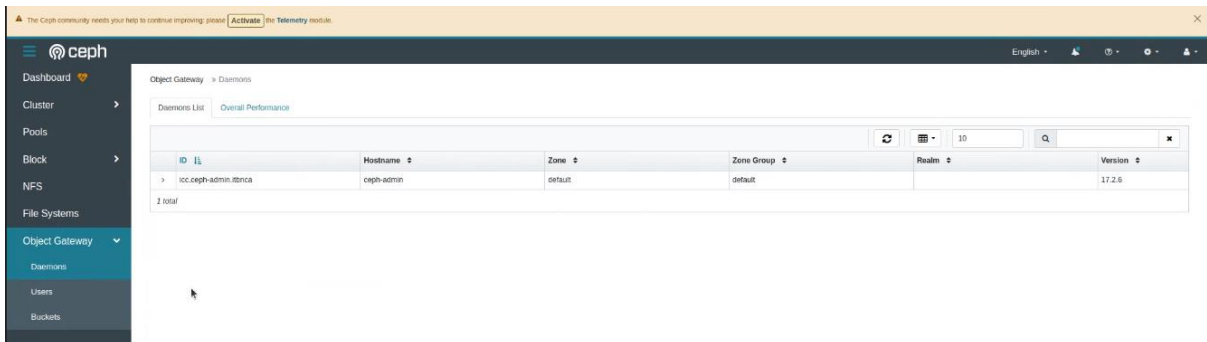


Figura 20: Listado de daemons de RGW dentro de Ceph.

8. Caso de uso

8.1. Implementación de Regiones para una Infraestructura Centralizada de nube comunitaria para la Escuela de Ingeniería de Sistemas Informáticos

En este caso de uso, se planteó la necesidad de implementar una infraestructura en la nube basada en OpenStack para cumplir los requerimientos de computación, almacenamiento y red de la Escuela de Ingeniería de Sistemas Informáticos de la Facultad de Ingeniería y Arquitectura de la Universidad de El Salvador. La infraestructura se organizó en forma de regiones, siendo cada región asignada a cada uno de los cuatro departamentos de la facultad:

Comunicaciones y CC. de la Computación, Programación y Manejo de Datos, Desarrollo de Sistemas y Administración.

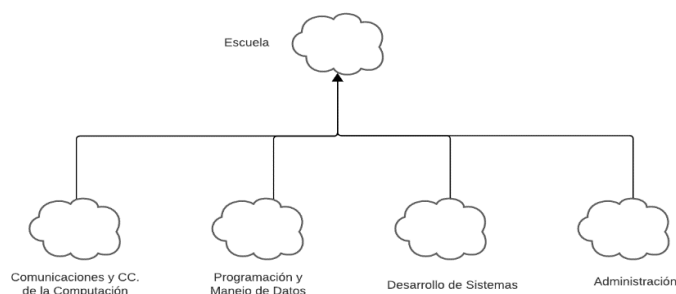


Figura 21: Diagrama de caso de uso de la escuela de sistemas.

El objetivo principal era un entorno de nube altamente escalable, seguro y disponible para los departamentos de la escuela. Esta infraestructura en la nube ha permitido el despliegue de servicios, garantizando la alta disponibilidad, la tolerancia a fallos y una gestión eficiente de los recursos. La infraestructura se divide en cuatro regiones, cada una correspondiente a un departamento específico de la escuela. Cada región se implementó utilizando OpenStack y se equipó con capacidades de computación, almacenamiento y red.

Los servicios de almacenamiento de cada región fueron integrados con Ceph, un sistema de almacenamiento distribuido, para asegurar una gestión segura y eficiente de los

datos. Esta integración proporcionó una capa adicional de seguridad para los datos almacenados.

Dentro de cada región, se establecieron zonas de disponibilidad para garantizar la alta disponibilidad y la tolerancia a fallos lo que permite la replicación de aplicaciones y datos, asegurando la continuidad del servicio incluso en caso de fallos de hardware o red.

Además cada región consume la API de la infraestructura central para integrarse de forma transparente por la cual facilita el aprovisionamiento de recursos, como máquinas virtuales, redes y almacenamiento, permitiendo una gestión centralizada y eficiente de los recursos de la nube.

Al asignar recursos de forma independiente a cada departamento, se evitó la sobrecarga, optimizando así el rendimiento de la infraestructura en su conjunto.

Se presenta una solución integral para la Escuela de Ingeniería de Sistemas Informáticos, proporcionando una infraestructura en la nube que se adapta a las necesidades específicas de cada departamento, garantizando la eficiencia operativa, la seguridad de datos y la continuidad del servicio.

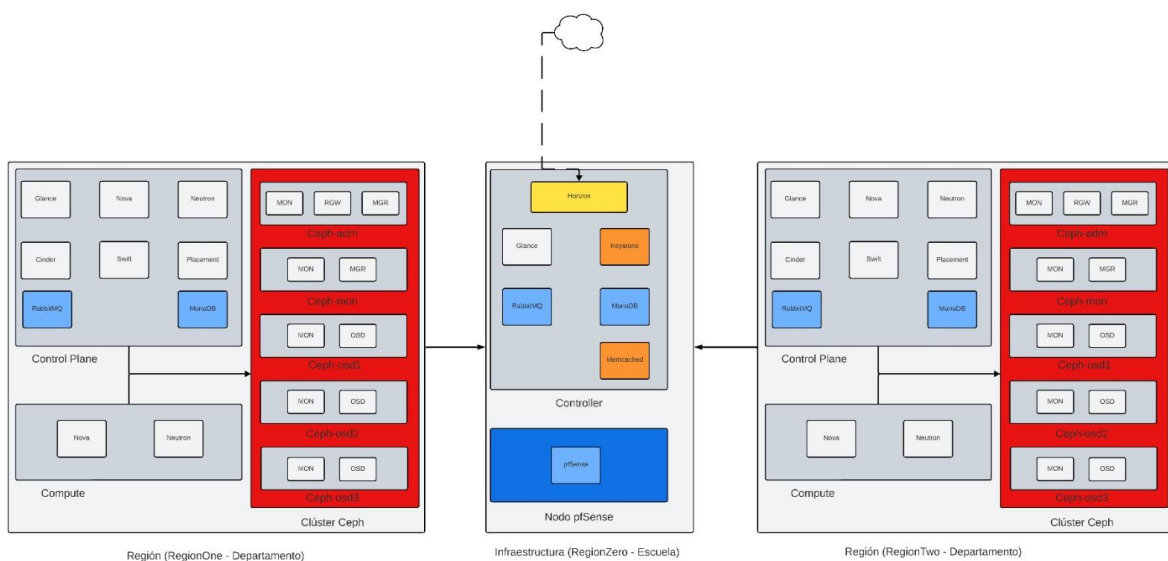


Figura 22: Funcionamiento de prototipo de la infraestructura (Escuela) conectado con las regiones (Departamentos).

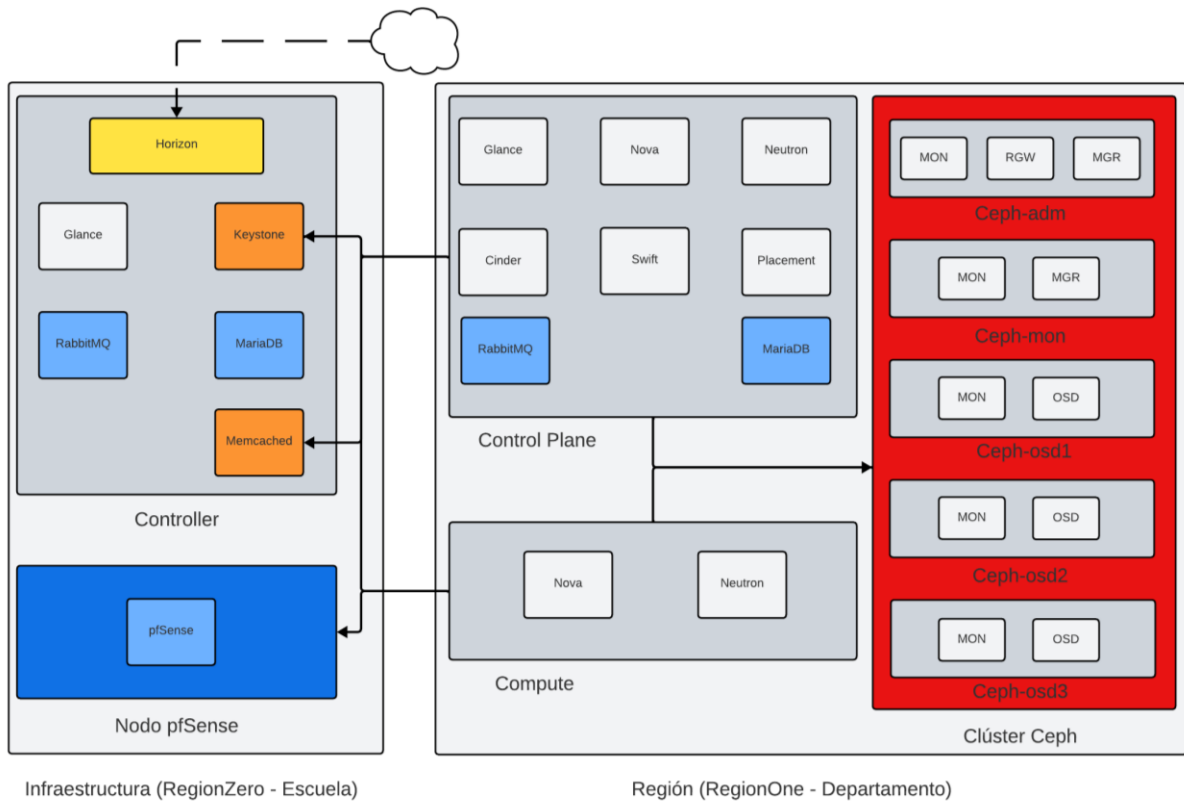


Figura 23: Funcionamiento detallado del prototipo para una región (departamento) aplicado al caso de uso en producción.

En la figura anterior se muestra la infraestructura (Escuela) conectada a una región (departamento) , se tiene lo que es la infraestructura que en este caso de uso RegionZero representa la Escuela de Ingeniería de Sistemas Informáticos, esta se compone de dos hosts uno que contiene el pfsense y otro host con los servicios esenciales para el funcionamiento de la infraestructura, en el caso de pfsense es un firewall pero para este escenario se utiliza más como un router que como firewall y en el caso del controller contiene los servicios que entre los más relevantes se tiene el servicio de Keystone y al memcached. El primero mencionado es el servicio de autenticación, y el segundo es porque ahí se genera el token que se utiliza en las regiones. También se cuenta con el servicio de Horizon que es el servicio que proporciona la interfaz web donde se pueden visualizar el resto de servicios. En el caso de las regiones representan los departamentos de la Escuela de Ingeniería de Sistemas Informáticos para este caso RegionOne como se nombra en la figura (puede tomar el nombre de cualquier departamento de la escuela de sistemas) consta de diferentes servicios, como Nova (servicio

de cómputo), Neutron (servicio de redes), Glance (servicio de imágenes) y Keystone (servicio de identidad). Todos ellos son servicios sumamente críticos para el funcionamiento del proyecto . En cuanto al almacenamiento, se está utilizando una solución distribuida para este caso ceph donde se guardan tanto máquinas virtuales, imágenes de sistema operativo, volúmenes de disco y respaldo de estos discos.

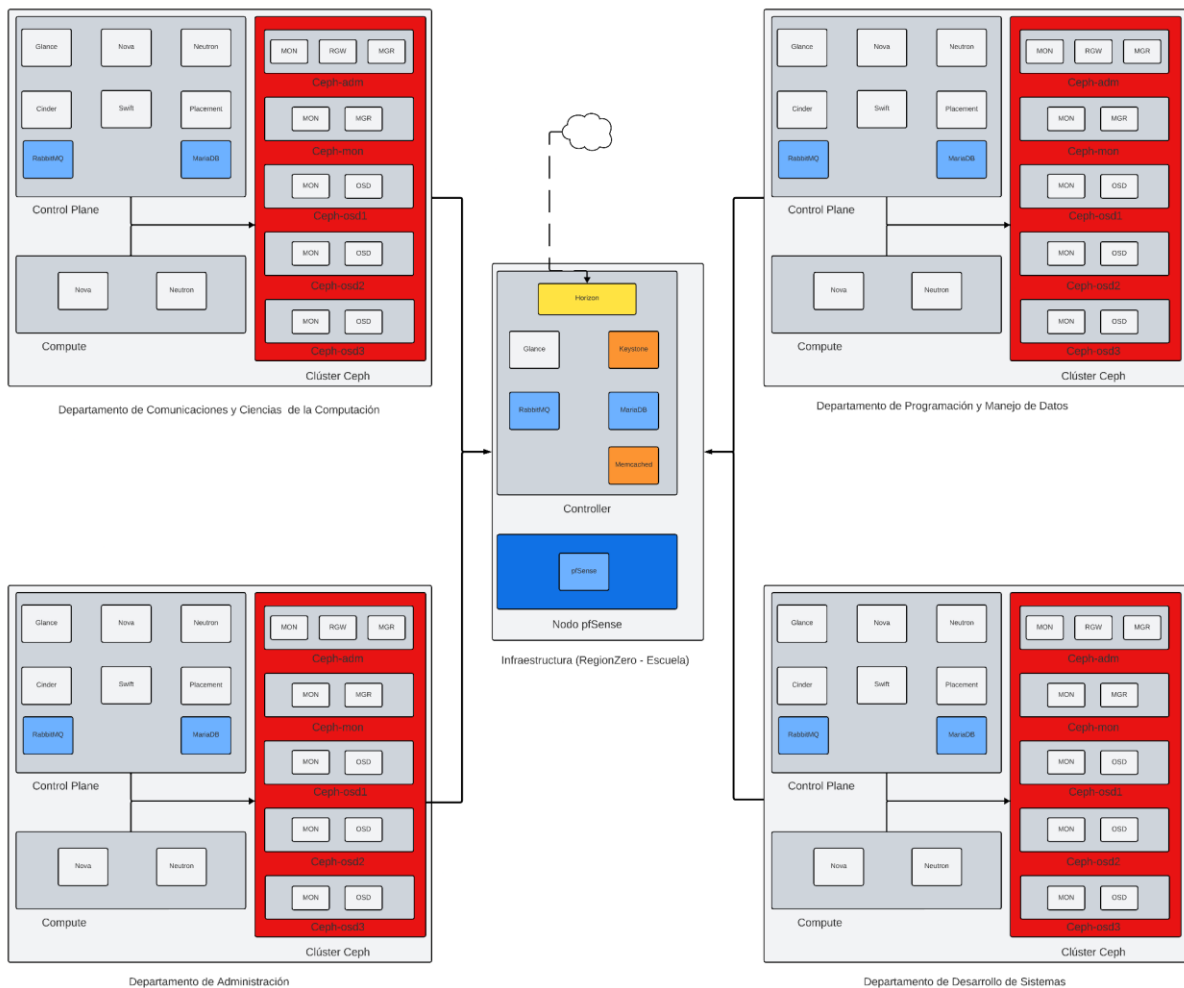


Figura 24: Funcionamiento de prototipo integrando las cuatro regiones de los departamentos aplicado al caso de uso en producción.

La solución completa puesta en producción sería una plataforma donde cada región, en este caso, los departamentos, se conecta a la infraestructura de la escuela. Cada región debería tener una solución de almacenamiento, que para este caso se decidió utilizar ceph . Dado que la escuela solo está manejando uno, si existieran las regiones separadas geográficamente, cada

una de las regiones debería tener un pfSense y se debería establecer una VPN entre ellas. Las tecnologías utilizadas incluyen máquinas virtuales bajo KVM, utilizando el sistema operativo Ubuntu Server. La resolución de seguridad también está bajo Ubuntu Server. Solamente para pfSense se utilizaría FreeBSD como sistema operativo.

9. Conclusiones

A través de este proyecto se logró desarrollar un prototipo de una nube configurada como región, con la capacidad de poder ser integrada a una infraestructura centralizada, siendo esta región desplegada de forma automatizada a través de Ansible utilizando múltiples nodos. El despliegue fue realizado mediante la alternativa de OpenStack, que ha demostrado ser una opción totalmente viable de utilizar, puesto que los servicios que brinda a usuarios finales son lo suficientemente completos para realizar las tareas que inherentemente una nube se compromete a soportar.

También, se logró realizar una exitosa integración de la región a la infraestructura, que fue probada primeramente construyendo una infraestructura centralizada con los componentes mínimos necesarios, y luego conectando los servicios de la región desplegada a ella, a través de la modificación de los archivos de configuración de dichos servicios. Para los potenciales usuarios finales, el resultado de esta integración puede ser apreciado mediante el panel de administración web que la infraestructura centralizada provee.

Además, mediante el caso de aplicación, se observó que es posible implementar el concepto de regiones a un contexto como el de la Escuela de Ingeniería de Sistemas Informáticos de la Facultad de Ingeniería y Arquitectura de la Universidad. Se consideró que la implementación de una nube sería de beneficio para los estudiantes, puesto que tendrían interacción con una tecnología que permitiría superar limitantes como la falta de recursos propios para poder realizar prácticas o desplegar los proyectos solicitados en las asignaturas.

Finalmente, se redactó un documento de diseño de bajo nivel, donde se detallan exhaustivamente las configuraciones requeridas para poder realizar el prototipo construido en este proyecto.

10. Recomendaciones

El éxito de la implementación de regiones dentro de una infraestructura en la nube y sus componentes depende en gran medida de seguir estas recomendaciones clave:

- 1. Capacitación del Personal:** Se recomienda brindar capacitación al personal en las nuevas tecnologías relacionadas con la infraestructura en la nube, con un enfoque en OpenStack y su implementación mediante Ansible, con el objetivo de mejorar la eficacia y productividad de la organización.
- 2. Considerar Futuras Aplicaciones:** No limitarse a la solución propuesta, sino considerar en futuras aplicaciones cómo aprovechar una arquitectura basada en microservicios en la infraestructura de OpenStack. Proporcionar opciones adicionales a empleados y usuarios es fundamental, y la adopción de microservicios puede mejorar la escalabilidad, flexibilidad y eficiencia de las aplicaciones en el entorno de OpenStack.
- 3. Personal Especializado:** Asegurarse de contar con personal especializado en el mantenimiento y uso de la tecnología de la nube, con un enfoque en OpenStack y la implementación de microservicios. Esto es esencial para garantizar la sostenibilidad a largo plazo de la infraestructura y la adopción exitosa de futuras aplicaciones basadas en microservicios.
- 4. Equipos de Alto Rendimiento:** Asegurarse de que el equipo informático utilizado para la implementación de la infraestructura de nube sea altamente capacitado y esté equipado con recursos de hardware y software adecuados para garantizar un rendimiento óptimo. Esto incluye la selección de servidores, sistemas de almacenamiento, y la configuración de la red para respaldar la infraestructura de OpenStack y los microservicios, asegurando así una ejecución eficiente y sin problemas de las aplicaciones y servicios.

Estas recomendaciones tienen como objetivo guiar a la entidad u organización responsable de llevar a cabo la implementación, asegurando una adopción exitosa y sostenible de la infraestructura de nube y sus componentes. Cada recomendación se centra en aspectos clave, desde la capacitación del personal hasta la consideración de futuras aplicaciones basadas en microservicios, con el fin de optimizar el rendimiento y la eficacia de la infraestructura en la nube. La implementación exitosa de estas recomendaciones puede contribuir a la mejora de la productividad y eficiencia de la organización en el uso de la tecnología de la nube.

11. Referencias Bibliográficas

Ambit BST. (9 de enero de 2020). Definición de IaaS, PaaS y SaaS: ¿En qué se diferencian?

Recuperado de <https://www.ambit-bst.com/blog/definición-de-iaas-paas-y-saas-en-qu%C3%A9-se-diferencian>

Amazon Web Services. (s.f.). ¿Qué es la nube híbrida? - Explicación de la computación en la nube híbrida - AWS. Recuperado de <https://aws.amazon.com/es/what-is/hybrid-cloud/>

AWS. (s.f.). ¿Qué es el almacenamiento de objetos? - Explicación sobre el almacenamiento de objetos. Recuperado de <https://aws.amazon.com/es/what-is/object-storage/>

AWS. (s.f.). ¿Qué es la virtualización? Recuperado de <https://aws.amazon.com/es/what-is/virtualization/>

AWS. (s.f.). ¿Qué es la virtualización? Recuperado de <https://aws.amazon.com/es/what-is/virtualization/#:~:text=La%20virtualizaci%C3%B3n%20es%20una%20tecnolog%C3%ADa,en%20una%20%C3%BAnica%20m%C3%A1quina%20f%C3%ADsica.>

AWS Pricing Calculator. (s.f.). Calculadora de precios de Amazon. Recuperado de <https://calculator.aws/#/>

Azure. (s.f.). Calculadora de precio de Azure. Recuperado de <https://azure.microsoft.com/es-es/pricing/calculator/>

Beservices. (octubre 2018). Servicios Clave de Google Cloud Platform para tu Empresa. Recuperado de <https://blog.beservices.es/blog/servicios-clave-de-google-cloud-platform-para-tu-empresa>

BITS. (s.f.). GlusterFS. Recuperado de <https://www.grupobits.co/integraciones/almacenamiento/glusterfs/>

Cloudflare. (s.f.). ¿Qué es la nube? | Conceptos esenciales | Cloudflare. Recuperado de <https://www.cloudflare.com/es-es/learning/cloud/what-is-the-cloud/>

Cognizant. (s.f.). Cloud Deployment. Cognizant. Recuperado de <https://www.cognizant.com/es/es/glossary/cloud-deployment>

Conzultek. (s.f.). ¿Qué es computación en la nube? Sus alcances. Blog Conzultek. Recuperado de <https://blog.conzultek.com/teletrabajo/que-es-computacion-nube-sus-alcances>

Cyrilla. (s.f.). Communications and Information Technology Comission. Cloud Computing Regulatory Framwork. Recuperado de <https://cyrilla.org/api/files/15894453395475rkkiz05dft.pdf>

Dgtlinfra. (10 de abril de 2023). Regiones de nube y zonas de disponibilidad: explicación. Recuperado de <https://dgtlinfra.com/cloud-regions-availability-zones/>

Educative. (s.f.). ¿Qué son las regiones, zonas y ubicaciones perimetrales en la computación en la nube? Recuperado de <https://www.educative.io/answers/what-are-regions-zones-and-edge-locations-in-cloud-computing>

EDU.LAT. (s.f.). Factibilidad técnica. Recuperado de <https://definicion.edu.lat/economia/factibilidad-tecnica.html>

Economipedia. (1 de septiembre de 2020). Factibilidad técnica. Recuperado de <https://economipedia.com/definiciones/factibilidad-tecnica.html>

Env0. (s.f.). Terraform Cloud: Benefits, Key Features, and Examples. Recuperado de <https://www.env0.com/blog/terraform-cloud>

Equipo editorial de IONOS. (28 de julio de 2020). GlusterFS vs. CEPH: los dos sistemas de almacenamiento cara a cara. IONOS Digital Guide. Recuperado de <https://www.ionos.es/digitalguide/servidores/know-how/glusterfs-vs-ceph/>

Equipo editorial de IONOS. (29 de julio de 2020). GlusterFS – ¿De qué se trata? IONOS Digital Guide. Recuperado de <https://www.ionos.es/digitalguide/servidores/know-how/que-es-glusterfs/>

Equipo editorial de IONOS. (30 de julio de 2020). CEPH – una práctica solución de almacenamiento para empresas de cualquier tamaño. IONOS Digital Guide. Recuperado de <https://www.ionos.es/digitalguide/servidores/know-how/que-es-ceph/>

Equipo editorial de IONOS. (7 de octubre de 2020). Galera Cluster: Vista general de un clúster MariaDB. IONOS Digital Guide. Recuperado de <https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/galera-cluster-para-mariadb/>

Gerald J. Popek y Robert P. Goldberg. (1974). Virtual Machine Concepts. Formal Requirements for Virtualizable Third Generation Architectures, 7(3), 2-3. <https://dl.acm.org/doi/pdf/10.1145/361011.361073>

Google. (s.f.). ¿Qué es cloud computing? | Google Cloud. Recuperado de <https://cloud.google.com/learn/what-is-cloud-computing?hl=es#:~:text=Cloud%20computing%20es%20la%20disponibilidad,%C3%BAnicamente%20por%20los%20que%20usen.>

Google Cloud. (s.f.). ¿Qué es una nube pública? Recuperado de <https://cloud.google.com/learn/what-is-public-cloud?hl=es-419>

Google Cloud. (s.f.). Calculadora de precios de google cloud. Recuperado de <https://cloud.google.com/products/calculator?hl=es-419>

IBM. (s.f.). ¿Qué es el almacenamiento en bloque? Recuperado de <https://www.ibm.com/mx-es/topics/block-storage>

IBM documentation. (s.f.). Recuperado de <https://www.ibm.com/docs/es/was-zos/9.0.5?topic=servers-introduction-clusters>

Invgate. (24 de marzo de 2023). Qué es Ansible: la herramienta DevOps para automatizar tareas de IT. Recuperado de <https://blog.invgate.com/es/ansible>

Instituto Tecnológico de Costa Rica. (25 de noviembre de 2014). Guía de Referencias para la Implementación de Operaciones Técnicas en OpenStack (Versión 1.4). Recuperado de <https://repositoriotec.tec.ac.cr/bitstream/handle/2238/6698/2-GuiaReferencias-Implementaci%C3%B3n-OperacionesTécnicas-OpenStack-1.4.pdf?sequence=3&isAllowed=y>

IONOS. (s.f.). ¿Qué es OpenStack? Recuperado de <https://www.ionos.es/digitalguide/servidores/herramientas/que-es-openstack/>

IONOS. (24 de agosto de 2023). ¿Qué es SaltStack? IONOS. Recuperado de <https://www.ionos.es/digitalguide/servidores/configuracion/que-es-saltstack/>

ITI. (s.f.). Herramientas de virtualización libres para sistemas GNU/LINUX. Recuperado de https://www.uv.es/sto/charlas/2010_CIM/hvl-cim-2010.html/index.html

Itilcom. (27 de noviembre de 2020). Los diferentes roles en los servicios cloud. Recuperado de <https://www.italcom.com/blog/los-diferentes-roles-en-los-servicios-cloud/>

Kinsta. (22 de junio de 2021). Beneficios de la Cloud Computing. Recuperado de <https://kinsta.com/es/blog/beneficios-de-la-cloud-computing/>

McKinsey & Company. (s.f.). IT as a Service: From Build to Consume. McKinsey & Company. Recuperado de <https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/it-as-a-service-from-build-to-consume>

Microsoft Azure. (s.f.). ¿Qué es una nube privada? Recuperado de <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-a-private-cloud>

Nube comunitaria. (s.f.). Recuperado de <https://www.capterra.es/glossary/482/community-cloud>

NUTANIX. (s.f.). ¿Qué es la automatización en la nube? Recuperado de <https://www.nutanix.com/mx/info/cloud-automation>

OpenNebula Project. (s.f.). OpenNebula Overview. Documentación de OpenNebula.

Recuperado de

https://docs.opennebula.io/6.6/overview/opennebula_concepts/opennebula_overview.html

OpenStack Documentation. (s.f.). Install Guide Overview. Documentación de OpenStack.

Recuperado de <https://docs.openstack.org/install-guide/overview.html>

Questionpro. (s.f.). ¿Qué es la metodología de la investigación? Recuperado de

<https://www.questionpro.com/blog/es/metodologia-de-la-investigacion/>

RedHat. (25 de enero de 2023). ¿Qué es la virtualización? Recuperado de

<https://www.redhat.com/es/topics/virtualization/what-is-virtualization>

Red Hat. (s.f.). ¿Qué es una nube pública y cómo funciona? Recuperado de

<https://www.redhat.com/es/topics/cloud-computing/what-is-public-cloud>

Revista Byte. (14 de febrero de 2023). Los desafíos de la nube. Byte TI. Recuperado de

<https://revistabyte.es/tema-de-portada-byte-ti/los-desafios-de-la-nube/>

Rootstack. (s.f.). ¿Qué es AWS y cuáles son sus servicios? Recuperado de

<https://rootstack.com/es/blog/que-es-aws-y-cuales-son-sus-servicios>

RedHat. (9 de noviembre de 2023). ¿Qué es Ansible y para qué sirve? Recuperado de

<https://www.redhat.com/es/technologies/management/ansible/what-is-ansible>

Stackscale. (s.f.). Hipervisores: definición, tipos y soluciones. Recuperado de

<https://www.stackscale.com/es/blog/hipervisores/>

SALTSTACK. (18 de abril de 2023). SALT SYSTEM ARCHITECTURE. Recuperado de

https://docs.saltproject.io/en/3004/topics/salt_system_architecture.html

Sumologic. (s.f.). What is role-based access control? Recuperado de

<https://www.sumologic.com/glossary/role-based-access-control/>

TIC Portal. (s.f.). Microsoft Azure. Recuperado de [https://www.ticportal.es/temas/cloud-](https://www.ticportal.es/temas/cloud-computing/microsoft-cloud/microsoft-azure)

[computing/microsoft-cloud/microsoft-azure](https://www.ticportal.es/temas/cloud-computing/microsoft-cloud/microsoft-azure)

Tecnozero. (2023). Servidor de dominio en la nube. Tecnozero Soluciones Informáticas.

Recuperado de [https://www.tecnozero.com/blog/servidor-de-dominio-en-la-](https://www.tecnozero.com/blog/servidor-de-dominio-en-la-nube/#:~:text=y%20medianas%20empresas.-)

nube/#:~:text=y%20medianas%20empresas.-

,%C2%BFQu%C3%A9%20es%20un%20servidor%20de%20dominio%20en%20la%20n

ube%3F,red%20local%20mediante%20una%20VPN.

Universidad Europea. (2021). Cloud Computing: ¿qué es y para qué sirve? Universidad

Europea. Recuperado de [https://universidadeuropea.com/blog/que-es-cloud-](https://universidadeuropea.com/blog/que-es-cloud-computing/#:~:text=El%20cloud%20computing%20o%20computaci%C3%B3n,escala%20gracias%20a%20la%20conectividad.)

computing/#:~:text=El%20cloud%20computing%20o%20computaci%C3%B3n,escala%

20gracias%20a%20la%20conectividad.

VMware. (16 de marzo de 2022). ¿Qué es una nube pública? | VMware Glossary.

Recuperado de <https://www.vmware.com/es/topics/glossary/content/public-cloud.html>

VMware. (s.f.). What is cloud automation? Recuperado de

<https://www.vmware.com/topics/glossary/content/cloud-automation.html>

Wikipedia. (s.f.). OpenNebula. Recuperado de <https://es.wikipedia.org/wiki/Opennebula>

12. Anexo

I. Guía de entrevista.....	1
A. Guía de entrevista No 1 y 2. Director y docente de Escuela de Sistemas Informáticos.....	1
II. Transcripciones de entrevista.....	4
A. Director de Escuela de Sistemas Informáticos	4
B. Docente de Escuela de Sistemas Informáticos.....	4
III. Diseño de bajo nivel	15

I. Guía de entrevista

A. Guía de entrevista No 1 y 2. Director y docente de Escuela de Sistemas Informáticos

Universidad de El Salvador
Facultad de Ingeniería y Arquitectura
Escuela de sistemas Informáticos
Especialización en Infraestructura Cloud

Trabajo de especialización

Este trabajo de especialización tiene como objetivo la implementación de una Nube Comunitaria en forma de una región basada en la plataforma OpenStack. La región creada se integrará de manera transparente en una infraestructura centralizada preexistente, ofreciendo servicios críticos de almacenamiento, red y cómputo.

Guía de entrevista No 1 y 2 dirigida a:

Director de Escuela de Sistema Informáticos Ing. Rudy Wilfredo Chicas Villegas

Docente Ing. Cesar Augusto González Rodríguez

I. Objetivo

Obtener información relevante sobre la percepción y viabilidad de la implementación de una infraestructura de nube en la Escuela de Ingeniería de Sistemas Informáticos, incluyendo conocimientos previos, importancia, factibilidad, capacidad técnica de los docentes, impacto, desafíos, recursos necesarios y beneficios potenciales.

II. Preguntas

1. ¿Usted tiene algún conocimiento previo sobre Openstack? Si la respuesta es SÍ, ¿Qué conocimientos tiene?
2. ¿Considera de importancia la implementación de una infraestructura de nube para la Escuela de Ingeniería de Sistemas Informáticos?
3. ¿Cree que es factible la implementación de una infraestructura de nube con los componentes mínimos necesarios como paso previo?
4. ¿Considera que los docentes de la Escuela tienen la capacidad técnica para aprender y utilizar esta tecnología de nube?
5. ¿Qué opinión tiene sobre la disposición de los docentes para profundizar en el conocimiento de esta tecnología y participar en un proyecto de este tipo?
6. ¿Qué impacto consideraría que tendría un proyecto de este tipo respecto a la gestión y distribución de recursos dentro de la Escuela de Ingeniería de Sistemas Informáticos? Por ejemplo, ¿implicaría la contratación de nuevo personal solo para administrar?
7. ¿Cuál es su opinión sobre la aplicación práctica del concepto de regiones en una nube en el contexto de la Escuela? Es decir, ¿vería provechosa o innecesaria la separación de los recursos?
8. ¿Cuáles son los desafíos potenciales y la disponibilidad de recursos que la Escuela de Sistemas enfrentará al implementar esta tecnología en términos de tiempo, capacitación de personal y otros aspectos necesarios para el proyecto en cada departamento?

- 9.** ¿Cuáles son las principales limitaciones que enfrentan actualmente los departamentos en términos de recursos tecnológicos?
- 10.** ¿Qué recursos adicionales necesitaría la escuela para implementar el proyecto?
- 11.** ¿Cuáles son los principales desafíos o limitaciones que enfrentan actualmente los departamentos en términos de recursos tecnológicos y cómo este proyecto podría ayudar a superarlos?
- 12.** ¿Cuál cree que sería el principal beneficio para la Escuela y sus departamentos al implementar esta solución de nube?
- 13.** ¿Qué ventajas específicas podría identificar en la integración de los servicios de almacenamiento de la nube con una solución de almacenamiento distribuida?
- 14.** ¿Cuál es su visión a largo plazo sobre cómo este proyecto podría mejorar la eficiencia y escalabilidad de los recursos en la Escuela de Ingeniería de Sistemas Informáticos?

II. Transcripciones de entrevista

A. Director de Escuela de Sistemas Informáticos

B. Docente de Escuela de Sistemas Informáticos

Transcripción entrevista

**Director de Escuela de Sistema Informáticos. Facultad de Ingeniería y Arquitectura.
Universidad de El Salvador**

Entrevista a : Ing. Rudy Wilfredo Chicas Villegas
Realizada : 6 de octubre de 2023
Hora de inicio : 19:00
Duración : 30:52 minutos
Lugar : Plataforma google meet
Guía utilizada : GE-TE-CG-1

Jorge: Buenas noches, muchas gracias por darnos la oportunidad. Como le comentaba, le solicitábamos el espacio debido al proyecto que estamos desarrollando dentro de la especialización de infraestructura en la nube. El cual se le explicara en breves momentos, además solicitamos su autorización para poder realizar la grabación de dicha llamada.

Rudy: Buenas noches jóvenes, claro es placer poderles apoyar y claro no hay problema con que graben.

José: Buenas noches, entonces para dar inicio, queremos brindar un pequeño contexto para que se entienda un poco más lo que hemos venido trabajando en el proyecto final en la especialización de infraestructura cloud.

Bueno principalmente así de forma muy breve mencionar lo que es una región. Que básicamente es una forma de organizar de gestionar los recursos de una nube. Ahora pueden existir despliegues de regiones que se encuentran separadas geográficamente o pueden estar divididas dentro del mismo centro de datos. Ahora son principalmente utilizadas para que exista redundancia y aislamiento ya sea de instancias o de volúmenes y pueden actuar como entidades independientes o pueden estar administradas de manera centralizada es de hecho el enfoque que le hemos venido dando. De forma general lo que hemos venido trabajando en el proyecto ha sido realizar el despliegue de una nube a manera de región para que esto se pueda integrar con una infraestructura y para probar que la integración sea exitosa también debemos de realizar el despliegue de una infraestructura con los componentes mínimos. La región que nosotros desplegamos tendrá los servicios críticos necesarios para que esta pueda funcionar y es importante también el elemento de almacenamiento, por lo cual se estará utilizando una solución distribuida.

Ahora como aplicación práctica hemos planteado tener a la escuela de sistemas informáticos que juegue el rol de infraestructura y luego los cuatro departamentos que la conforman tengan el papel de una región está forma global ahora en la siguiente diapositiva Sería más o menos como lo tendríamos, ya con los componentes de la nube que estamos utilizando que es openstack donde los diferentes servicios que estén instalados en una región se conecten con lo que esté brindando la infraestructura y puedan ser administrados bajo el panel web que es Horizon. En la siguiente diapositiva ya está cómo se vería aproximadamente cuando los cuatro departamentos están haciendo las veces de región y conectándose a la infraestructura Ahora con esta introducción vamos a dar inicio a lo que es las preguntas.

Jorge: Primera pregunta ¿Usted tiene algún conocimiento previo sobre OpenStack?

Rudy: Sí.

Jorge: ¿Qué conocimiento tiene al respecto?

Rudy: En el año 2018, si no recuerdo mal, creamos un clúster, en la escuela no sé si ustedes tuvieron la oportunidad de verlo funcionando. En resumen lo que hicimos fue usar OpenStack para convertir ocho computadoras de escritorio en un solo servidor y luego en ese servidor creamos las máquinas virtuales que iban a utilizar los estudiantes para los proyectos de entonces. Ese fue el uso que le hicimos y sí conozco la tecnología y la hemos implementado.

Jorge: Yo recuerdo que cuando llevé la materia, medio se habló que se quería implementar eso pero por lo visto ya en el siguiente año ya se implementó.

Rudy: Sí es probable que haya sido después de que ustedes la cursaron.

Jorge: Siguiendo ¿considera de importancia la implementación de una infraestructura de nube para la escuela de ingeniería de sistemas informáticos.

Rudy: Sí también, es que hay que decir que depende también del uso que se le quiera dar a este, en una buena parte nosotros la concebimos como una infraestructura que nos puede servir para efectos académicos la parte digamos administrativa está cubierta digamos que la facultad tiene esta bastante capacidad de cómputo para aplicaciones internas de la facultad. Pero obviamente también se requiere infraestructura para el ámbito académico que actualmente no tenemos.

Entonces sí yo lo vería bastante útil y este y creo que puede trabajar en eso actualmente.

Jorge: ¿Los servidores con los que cuenta la facultad están destinados más que todo para aplicativos, más para ámbito académico no les brindan ningún espacio?

Jorge: Por ejemplo cuando yo llevé lo que es programación móvil esté siempre se necesitaba lo que es darle al estudiante lo que es unas máquinas virtuales para poder levantar todos sus requerimientos verdad Pero recuerdo que el ingeniero César nos decía de que no tenía donde darnos esa posibilidad entonces por eso le pregunto.

Rudy: Depende como te decía para qué lo querrás nosotros en la escuela lo vemos más como espacios en los que ustedes pueden hacer eh prácticas o proyectos por ejemplo los proyectos de diseño actualmente yo conozco varios estudiantes y no es algo nuevo es algo que viene ocurriendo desde hace varios años que ellos pagan digamos servicio un servidor en la nube, o sea tienen que gastar en varias cosas para poder esté digamos ejecutar los proyectos en una infraestructura que sea medianamente decente yo creo que nosotros deberíamos de proveerles esa infraestructura y en eso es lo que se podría trabajar. Intentar crear digamos esas porque para la parte como decía de aplicaciones internas de la facultad si tenemos infraestructura.

Jorge: Vamos a hablar con respecto a la factibilidad, ¿cree que es factible la implementación de una infraestructura de nube con los componentes mínimos necesarios?

Rudy: Una infraestructura mínima de nube. Sí lo que pasa es que yo no sé si ustedes u otro grupo creo que fue otro grupo que me preguntaban sobre este tema. Y me decía si lo veía factible y yo les decía sí, pero hay que tomar en cuenta que hay restricciones más de índole, de la estructura interna de la universidad en la que muy probablemente lo que ocurriría es que hayan facultades que no tengan esas condiciones o simplemente no tengan ese interés, es también eso es posible entonces yo no como te digo sí lo veo factible pero también depende

como en parte para qué exactamente es lo que quieres hacer o sea qué exactamente es lo que quieres hacer y qué y digamos cuáles serían los niveles de inversión por ejemplo que vas a requerir para hacer esa ese trabajo ósea montar esa infraestructura no sé si respondí a tu pregunta.

Jorge: Sí me respondió la pregunta y que en ese caso yo tengo una experiencia en lo que es Cuando llevé las materias no sé si todavía el ingeniero Vides sigue dando lo que es comercio electrónico cuando yo la llevé con él , lo que hace pagar el propio cloud en digitalocean de su dinero y se lo brinda a los estudiantes para que pueda realizar las prácticas y por eso le preguntaba porque más que todo siempre nos entrega lo que es máquina bien estándar Podemos continuar entonces vamos a hablar con respecto a la capacidad ¿considera que los docentes de la escuela tienen la capacidad técnica para aprender y utilizar esta tecnología de la nube?

Rudy: Sí, sí con algunas restricciones pero sí.

Jorge: Okay ¿qué opinión tiene sobre la disposición de los docentes para profundizar en el conocimiento de esta tecnología y participar en un proyecto de esta índole de este

Rudy: Mira hay que decir varias cosas al respecto en primer lugar hay que entender que hay varias carreras que se diversifican en varias áreas y que por lo tanto no podemos esperar que todos los compañeros tengan la disposición ni el tiempo para invertir en un aprendizaje de esa naturaleza o sea es así, tal cual no hay matices ahí. No es nada extraño pensar en que un compañero por ejemplo que esté más enfocado en el área de bases de datos , se vaya a meter a querer implementar una nube por ejemplo ósea, no sería nada eh extraño pensar en que no va a ser así, sin embargo este hay, yo considero que si hay voluntad de varios compañeros por lo menos convertirse en usuarios yo creo que la mayoría se convertiría en usuarios de esas herramientas en el nivel que les corresponda. No si hay, eso sí te lo puedo decir si hay compañeros que están enfocados en hacer implementaciones es decir ya gente que está pensando en meterse así como lo hicimos nosotros en su momento con el ingeniero Montano en meterse en ver qué se puede hacer y utilizar una vez implementadas, las soluciones utilizarlas así que yo creo que cada quien en el ámbito que le corresponde es bastante viable que se haya una disposición bastante grande de parte de los compañeros con respecto a la aplicación práctica.

Jorge: Vamos a preguntar ¿qué importancia tendría un proyecto de este tipo respecto a la gestión y distribución de recursos dentro de la escuela de ingeniería de sistema informático? Por ejemplo, implicaría la contratación de nuevo personal solo para administrarlo.

Rudy: La infraestructura sería parte del docente que ya se encuentra en escuela o se contrata personal extra. Lo que nosotros hemos hecho en otras oportunidades es contratar personal. Otra opción mejor es asumirlo nosotros como profesores eso es una la digamos la principal estrategia que utilizó, evidentemente este es algo que tiene sus pros y sus contras porque no forma parte del trabajo ordinario de nosotros los profesores y que por lo tanto como puede que lo haga o puede que no lo haga, eso trae consecuencias evidentemente es decir que de repente no te resuelven, hay un error algún problema, la gente no tiene tiempo para resolverlo etcétera y lo que hemos optado a partir del año 2021 es contratar gente no es gente experimentada no es gente digamos con experiencia en esa área pero que nos apoya en esa tarea de buscar mantener los servicios y acciones que tenemos en la escuela funcionando. Yo tendería más a eso o sea pero alguien que se dedique estrictamente a eso a administrar esa infraestructura porque

obviamente es algo que es demandante cuando pasan incidencias. Esa tarea te demanda mucho trabajo y nos ha pasado ya que hay escenarios en los que no podemos dar abasto, no damos abasto simplemente.

Jorge: Continuando ¿cuáles son los desafíos potenciales y la disponibilidad de recursos que la escuela de sistemas enfrentará al implementar esta tecnología en términos de tiempo capacitación de personal y otros aspectos necesarios para el para que el proyecto en cada departamento.

Rudy: Mira primero infraestructura porque como mencioné antes la facultad tiene infraestructura pero no está destinada para los efectos. Que para los que yo lo concibo te lo voy a resumir así para mí un proyecto de esta naturaleza tiene sentido para efectos académicos hablo por la facultad o por la escuela no hablo por toda la universidad para para eso tendría sentido para nosotros un proyecto como este por qué la infraestructura para aplicaciones administrativas o académico administrativas en la facultad ya está resuelto ósea ya tenemos infraestructura.

Por lo tanto habría que crear una infraestructura física que es lo que se expondría a la nube para esos efectos eso sería el primer reto el segundo reto sería conformar un equipo de trabajo entre los que tendría que estar un profesor del área para que él sea el que el que trabaje en eso y un personal administrativo una persona que es el que va a estar el que haría digamos ya la administración de la de la infraestructura, por decir algo crear las máquinas virtuales este etcétera este eso es lo que lo que creo yo que sería lo primero como les respondí a la pregunta anterior si había personal capacitado sí lo hay y hacía la acotación de que no todos no todos están capacitados porque hay unos que se dedican a otras áreas pero si hay personal capacitado que pueda, bueno yo mismo como les comentaba hicimos una implementación de OpenStack en la escuela junto con el ingeniero Boris Montano hay otros compañeros que conocen la tecnología también y ese no sería problema pero digamos contratar el personal que se requiere y montarlo en una infraestructura física creo que serían los principales retos que hay.

Jorge: Podríamos cerrar con ¿cuál es su visión a largo plazo sobre este proyecto podría mejorar la eficiencia y la escalabilidad de los recursos con los que cuenta la escuela actualmente ya viéndolo a un futuro?

Rudy: Esa pregunta es complicada porque estamos asumiendo de que por implementar una solución como esa, todo va a ser color de rosa y no es así eh va a mejorar evidentemente el acceso que tienen los estudiantes a esta tecnología para efectos de poder desarrollar los trabajos o los proyectos que se les dejan en diferentes materias con lo cual obviamente va a mejorar , si va a mejorar las asimilación de las competencias que estamos buscando, este pero decir de qué va digamos va a tener un impacto trascendental que no fue ese el término que usaron pero es más o menos como suena, no lo podría decir así no sé si contesté la pregunta sí se entendió.

Jorge: Ese era el apoyo que necesitábamos con esas interrogantes y pues agradecerle por el espacio que nos brindó y no sé si mis compañeros quisieran abonar algo más.

José: No, estaríamos bien, muchas gracias ingeniero por su tiempo.

Rudy: De nada con gusto.

Jorge: Entonces prácticamente eso sería todo y si necesitamos otra ayuda le

estaremos escribiendo.

Rudy: No hay problema, con gusto.

Jorge: Muchas gracias, buenas noches.

Rudy: Noches, jóvenes.

Transcripción entrevista

**Docente de Escuela de Sistema Informáticos. Facultad de Ingeniería y Arquitectura.
Universidad de El Salvador**

Entrevista a : Ing. Cesar Augusto González Rodríguez
Realizada : 20 de octubre de 2023
Hora de inicio : 18:00
Duración : 47:44 minutos
Lugar : Plataforma google meet
Guía utilizada : GE-TE-CG-1

Jorge: Buenas noches, muchas gracias por darnos la oportunidad. Como le comentaba, le solicitábamos el espacio debido al proyecto que estamos desarrollando dentro de la especialización de infraestructura en la nube. El cual se le explicara en breves momentos, además solicitamos su autorización para poder realizar la grabación de dicha llamada.

César: Buenas noches jóvenes, adelante, graben la llamada.

Jorge: Brindó el espacio a José Andrés, para que realice lo que es la introducción de la entrevista, que consiste en lo que es explicarle de manera resumen de lo que trata el prototipo y caso estudio que tenemos asignado.

José: Buenas noches. Como mencionaba mi compañero, hemos estado trabajando bajo OpenStack, que es la tecnología principal que hemos utilizado para nuestro prototipo. OpenStack es una plataforma de código abierto que nos permite la creación y gestión de infraestructura pública y privada. Esta consta de diferentes servicios, como Nova, Neutron, Glance y Keystone. Todos ellos son sumamente críticos para poder desplegarla, y por supuesto, existen proyectos adicionales que nos brindan características más especializadas. También es respaldada por una gran comunidad de desarrolladores y empresas.

Ahora, como estamos trabajando en la idea del despliegue de una región, básicamente una región es una forma de organizar y gestionar los recursos de una nube. Pueden existir estas regiones de forma separada, ya sea geográficamente o incluso desde dentro del mismo Data Center, solo de forma lógica. Principalmente se utilizan para tener redundancia y aislamiento. Estas pueden ser entidades independientes o pueden ser administradas desde un panel central, que es lo que hemos estado trabajando.

Nuestro prototipo ha sido tratar de desplegar una nube configurada como región e integrar en una infraestructura centralizada. Para que nosotros podamos probar que se integre de forma correcta, también hemos desplegado una infraestructura con los componentes mínimos para comprobar que todo está funcionando bien. Esta infraestructura y región se pueden administrar desde un panel web, que es un proyecto llamado Horizon.

En cuanto al almacenamiento, se está utilizando una solución distribuida. En referencia a la aplicación práctica, hemos manejado que la escuela tome el rol de infraestructura y los departamentos que la componen tomen el rol de las regiones. Idealmente, la solución completa sería una plataforma donde cada región, en este caso, los departamentos, se conecta a la infraestructura de la escuela. Cada región debería tener una solución de almacenamiento, que hemos decidido utilizar.

Desde un punto de vista más detallado, realmente lo crucial que se necesita para que funcione correctamente es que los servicios de las regiones puedan conectarse al servicio de autenticación en Keystone y al memcached. El primero mencionado es el servicio de autenticación, y el segundo es porque ahí se genera el token que se utiliza en las regiones.

También tienen que conectarse a un pfSense que estamos utilizando más como firewall que como router. Dado que la escuela solo está manejando uno, si existieran las regiones separadas geográficamente, cada una de las regiones debería tener un pfSense y se debería establecer una VPN entre ellas.

Las tecnologías que hemos estado utilizando incluyen máquinas virtuales bajo KVM, utilizando el sistema operativo Ubuntu Server. La resolución de seguridad también está bajo Ubuntu Server. Solamente el pfSense estamos utilizando FreeBSD como sistema operativo. Finalmente, para la automatización se usa ansible, que despliega contenedores LXC para los nodos controladores de las regiones.

Jorge: Solo para abordar un resumen de la temática que queremos abordar en la escuela es que actualmente, a nivel académico, los estudiantes se enfrentan a la dificultad de no poder levantar, ya sea un emulador o algo, en los trabajos finales para desarrollarlos debido a la falta de recursos en sus computadoras locales. Lo que queremos atacar es darles a los estudiantes una herramienta por la cual la escuela les va a estar otorgando esos recursos para que puedan desarrollar ya sea un trabajo final o una aplicación.

Entonces, con eso, ya podemos iniciar con las preguntas. Nosotros tenemos elaborada una guía estructurada de preguntas, de las cuales, a medida que vayamos hablando, posiblemente vayamos contestando una o más adelante. Si es así, estas preguntas se irán omitiendo y solo realizaremos las que no se han cubierto todavía.

Iniciando, ¿usted tiene algún conocimiento previo acerca de OpenStack?

César: Bueno, yo tengo conocimientos más teóricos que prácticos. Por referencia, hace unos 6 años, el ingeniero Montano con el ingeniero Chicas hicieron un arreglo de un servidor utilizando OpenStack para asignarle instancia a cada estudiante y que desarrollarán su proyecto final. También recuerdo que el servidor de donde estamos alojados como página institucional en la UES utiliza OpenStack, aunque no estoy seguro. Tal vez ustedes se lo han dicho en el curso.

Jorge: ¿Tiene conocimiento de cómo funciona OpenStack o si ha configurado alguno de sus servicios?

César: Sé cómo funciona, pero no lo he configurado. Sin embargo, puedo orientar en el sentido de las necesidades que tenemos en las materias. En la virtual que tenemos asignada por parte de la unidad de donde está el ingeniero Damián, tengo un pequeño FTP, que es uno de los servicios proporcionados. Pero entramos en una cuestión de ampliación de espacios, por lo tanto, no lo puedo utilizar. Igual hay cuestiones que necesitamos madurar en cuanto al desarrollo.

Me gustaría saber si este servicio o estos servicios pueden permitirme ampliar el espacio asignado, acceder a la consola de mi máquina virtual o asignar virtuales a estudiantes, ya que eso sería de gran interés.

Jorge: ¿Considera de importancia la implementación de una infraestructura de nube para la Escuela de Ingeniería de Sistemas Informáticos?

César: Lo que necesitamos es una infraestructura, ya que las materias que usamos requieren este tipo de infraestructura. No solo las dos que mencionó, sino también otras como Técnicas de Programación e Internet que, debido a la falta de recursos, no se usan. La idea es que evolucione y que, al final, entreguemos una tarea que se haga de forma colaborativa, lo que no se hace actualmente debido a la falta de recursos. La expectativa es que, al finalizar estas

materias, los estudiantes se gradúen como desarrolladores capaces de manejar todas estas tecnologías.

La importancia de esto es significativa, y si su proyecto incluye alguna implementación, sería beneficioso. En caso de que lo hagan, podríamos solicitar una IP pública y alojar el servidor en donde está José Andrés. Esto se podría hacer en conjunto con Rudy o Damián.

Jorge: Por ahora, el proyecto solo se enfoca en presentar un prototipo. La implementación completa podría quedar para futuros casos de estudio.

César: Si desean tener algo semi-utilizable que no implique costos significativos, podríamos partir de la idea de una distribución donde la escuela y los departamentos tomen un rol. Por ejemplo, el departamento de programación para dispositivos móviles podría ser un nodo en la infraestructura que están manejando.

Jorge: ¿Cree que es factible la implementación de una infraestructura de nube con los componentes mínimos necesarios como paso previo?

César: Entendemos que esta pregunta no es de sí o no, sino que requiere una respuesta documentada. La factibilidad de la implementación de una infraestructura de nube con los componentes mínimos necesarios tiene tres aspectos. En el aspecto técnico, el ingeniero Damián podría proporcionar más detalles. En cuanto a la factibilidad económica, si el presupuesto es demasiado elevado, se debe evaluar si la inversión es justificada. Esto podría incluir una valoración subjetiva de cuánto se podría invertir o ahorrar si se realiza la implementación.

A nivel operativo, también debemos considerar la aceptación de la solución. En el pasado, ha habido sistemas que no tuvieron éxito porque no se garantiza acceso a recursos críticos, como el LDAP. Es importante asegurar que los usuarios puedan acceder a las funciones necesarias, como la validación de cuentas de docentes y estudiantes.

La estimación de crecimiento y disponibilidad es fundamental para determinar la viabilidad técnica. La factibilidad no debe tomarse a la ligera; se deben considerar las necesidades y la disponibilidad de recursos.

Jorge: ¿Qué opinión tiene sobre la disposición de los docentes para profundizar en el conocimiento de esta tecnología y participar en un proyecto de este tipo?

César: En cuanto a la disposición de los docentes, la mayoría que ha salido del sistema se ha mostrado dispuesta a profundizar en el conocimiento de esta tecnología y a participar en un proyecto de este tipo. Algunos ya tienen experiencia en el área, mientras que otros pueden necesitar actualizaciones. Los docentes están dispuestos a participar, pero su nivel de conocimiento varía.

Jorge: ¿Cuáles son los desafíos potenciales y la disponibilidad de recursos que la Escuela de Sistemas enfrentará al implementar esta tecnología en términos de tiempo, capacitación de personal y otros aspectos necesarios para el proyecto en cada departamento?

César: En cuanto a la administración de recursos, la idea es que esta implementación no requiera personal adicional. Debería quedar automatizada de tal manera que con la capacitación adecuada, los docentes que utilicen estos recursos puedan realizar las instalaciones con facilidad utilizando guías y comandos. No sería necesario un proceso manual de administración

de usuarios. Los docentes de la escuela deberían poder manejarlo con capacitación adecuada, y no requeriría una contratación adicional de personal.

En cuanto a la aplicación práctica, sería beneficioso implementarlo en la escuela. No es necesario dividirlo en unidades administrativas separadas, como jefaturas, ya que una sola región que represente la Escuela de Sistemas es suficiente. La separación no debería ser por estudiantes individuales, sino, como mencioné anteriormente, por materia o asignatura. Esto sería más conveniente y evitaría la burocracia innecesaria. La distribución de acceso por asignatura sería más apropiada en lugar de un único acceso global para todos.

Jorge: ¿Cuáles son las principales limitaciones que enfrentan actualmente los departamentos en términos de recursos tecnológicos?

César: En cuanto a las limitaciones, tenemos diversas restricciones. Por ejemplo, en términos de recursos tecnológicos, nuestras limitaciones incluyen las máquinas y cómputos disponibles. A menudo, las necesidades dentro de las materias requieren un enfoque más profesional. Por ejemplo, en el caso de una materia de bases de datos hace unos 6 años, optamos por la virtualización ya que carecemos de la infraestructura necesaria. Creamos un servidor FTP llamado "aula.Fia", y aunque esta fue una solución, no fue la ideal. La falta de una infraestructura más robusta ha limitado nuestra capacidad de proporcionar un entorno más profesional para la práctica. En realidad, lo que deseamos es brindar a los estudiantes un entorno semi profesional para sus prácticas, pero estas limitaciones a veces nos obligan a utilizar soluciones más caseras.

En términos de disponibilidad, también enfrentamos restricciones, como los cortes de energía. Los cortes de energía inesperados ocurren con cierta frecuencia en nuestra área. Aunque ha habido mejoras en la infraestructura eléctrica, estas interrupciones pueden afectar nuestra infraestructura, lo que obliga a buscar alternativas que pueden no ser las óptimas para las prácticas de los estudiantes.

Respecto al espacio físico y recursos asignados, nuestras limitaciones están relacionadas con el espacio en disco duro y la capacidad eléctrica. Como mencioné, el espacio de disco duro es una restricción, y a veces tenemos que trabajar con recursos asignados, lo que puede limitar nuestra capacidad de proporcionar servicios más amplios y completos.

Jorge: ¿Cuál es su visión a largo plazo sobre cómo este proyecto podría mejorar la eficiencia y escalabilidad de los recursos en la Escuela de Ingeniería de Sistemas Informáticos?

César: En cuanto a la visión a largo plazo de este proyecto, es una idea que debe ser evaluada, especialmente en el contexto de una nueva currícula. En la actualidad, la escuela se enfrenta a desafíos para proporcionar recursos académicos óptimos a los estudiantes, y la implementación de este proyecto podría abordar esos problemas.

A corto plazo, se requiere que tanto la universidad presencial como la de distancia dispongan de recursos de alta calidad para todos los estudiantes, de forma gratuita. Esto implica que la universidad debe financiar este tipo de servicio y garantizar que todas las materias cuenten con los recursos tecnológicos necesarios para impartir sus tecnologías. El objetivo es que los estudiantes tengan acceso a estos recursos para desarrollar sus competencias, como la creación de micro servicios o aplicaciones móviles, lo que les daría ventajas en el mercado laboral.

Además, se busca proporcionar suficientes recursos para cubrir las necesidades de las asignaturas y los trabajos de grado. Los estudiantes a menudo incurren en gastos al realizar prácticas o proyectos, y la idea es ofrecer un espacio en la infraestructura de la escuela para

que las materias puedan realizar prácticas de alta calidad sin problemas de infraestructura o recursos.

En resumen, la visión a largo plazo implica proporcionar recursos de alta calidad y suficientes para todas las materias y los proyectos de los estudiantes, lo que mejoraría la eficiencia y calidad de la educación en la escuela de sistemas. Esto permitiría a los estudiantes adquirir las habilidades y competencias necesarias para destacar en el mercado laboral.

Jorge: Ese era el apoyo que necesitábamos con esas interrogantes y pues agradecerle por el espacio que nos brindó y no sé si mis compañeros quisieran abonar algo más.

José: No, estaríamos bien, muchas gracias ingeniero por su tiempo.

César: Estoy para ayudarles en lo que pueda y esté al alcance.

Jorge: Entonces prácticamente eso sería todo y si necesitamos otra ayuda le estaremos escribiendo, buenas noches.

César: Nos vemos jóvenes , buenas noches.

III. Diseño de bajo nivel

**UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE SISTEMAS INFORMÁTICOS**



**DISEÑO DE BAJO NIVEL PARA EL PROTOTIPO DE DESPLIEGUE
AUTOMATIZADO DE REGIONES PARA SU INTEGRACIÓN A UNA
INFRAESTRUCTURA CENTRALIZADA CON UN CASO DE USO EN UNA
ESCUELA DE UNA UNIVERSIDAD**

PRESENTADO POR:

CASTRO SÁNCHEZ JOSÉ ANDRÉS

MÁRQUEZ VENTURA JONATHAN ENRIQUE

MEDINA HUEZO GERARDO ANTONIO

MEJÍA ALVARADO JORGE OVIDIO

Índice de contenido de diseño de bajo nivel

I.	Arquitectura.....	1
A.	Redes	1
B.	Asignación de IPs	2
II.	Requisitos iniciales.....	3
III.	Máquinas virtuales.....	3
A.	Inventario de máquinas virtuales	3
B.	Imágenes	5
1.	Ubuntu Server 20.04.	5
2.	pfSense.....	5
C.	Creación de máquinas virtuales en VMM.	6
1.	Instalando paquetes.....	6
2.	Máquinas virtuales para pfSense.....	7
3.	Máquinas virtuales con Ubuntu.....	24
IV.	Configuración e Instalación	42
A.	pfSense	42
1.	Agregando VLANS.	42
2.	Agregando Interfaces.....	44
3.	Agregando Reglas.....	46
B.	Ceph	48
1.	Preparando nodos.	48
2.	Iniciando el clúster.	49
3.	Registrando nodos al clúster.	51
4.	Inflando el clúster.....	52
5.	Creando pools.	52
6.	Creando usuarios y keyrings.....	53
C.	Infraestructura	53
1.	Preparando nodo.	53
2.	Chrony.....	54
3.	Repositorio de OpenStack.	54
4.	MariaDB.....	55
5.	RabbitMQ.	55

6.	Memcached.....	55
7.	Keystone.....	56
8.	Glance.....	59
9.	Horizon.....	61
D.	Región.....	62
1.	Preparando nodos.	62
a)	Nodo Deployment.	62
(1)	Configurando llaves SSH.....	63
(2)	Instalar dependencias y repositorio.....	63
b)	Nodos Control-Plane y Compute.....	64
(1)	Instalando paquetes.....	66
2.	Configurando despliegue de Ansible.	66
3.	Ejecutando playbooks.	71
4.	Verificación del despliegue.....	71
V.	Integración	72
A.	Modificando configuración de servicios de la región.....	72
1.	HAProxy.....	72
2.	Glance.....	73
3.	Placement.	73
4.	Nova.....	74
a)	Nodo control-plane.	74
b)	Nodo compute.	75
5.	Neutron.....	75
a)	Nodo control-plane.	75
b)	Nodo compute.	76
6.	Cinder.....	76
B.	Agregando el servicio swift a la región	77

I. Arquitectura

OpenStack-Ansible soporta diferentes arquitecturas de red, que pueden ser desplegadas usando una sola interfaz de red para ambientes de pruebas o usando múltiples interfaces para entornos de producción.

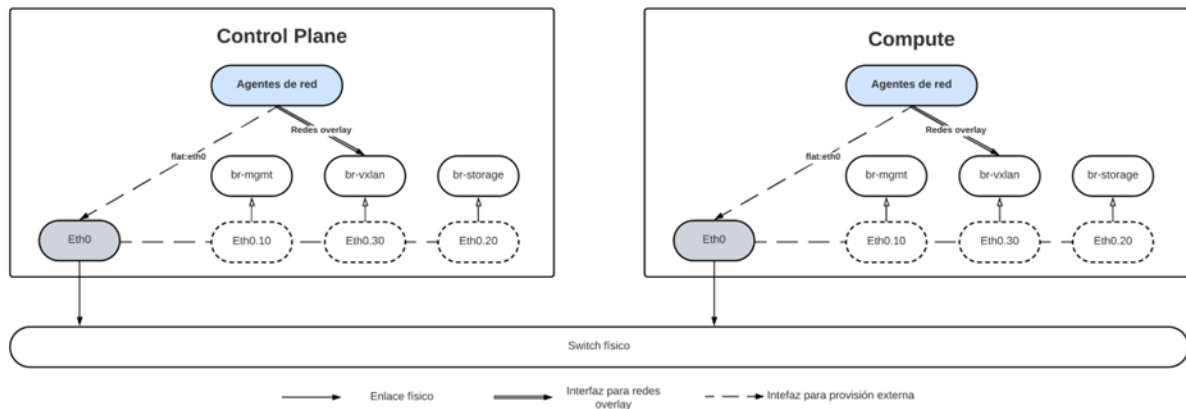


Figura 1: Disposición de interfaz de red única para OpenStack-Ansible.

A. Redes

Tabla 1
Interfases de red

Red	CIDR	VLAN	Observaciones
Management	172.29.236.0/22	10	
Tunnel (Vxlan)	172.29.240.0/22	30	
Storage	172.29.244.0/22	20	
VLAN	172.29.248.0/22	40	Será utilizada como red provider
LAN	192.168.16.0/24		Utilizada para comunicar a los hosts físicos
NAT	192.168.122.0/24		Será utilizada como red provider

La tabla 1 muestra la disposición de interfaz de red única para OpenStack-Ansible.

B. Asignación de IPs


Se indican las IPs para dos regiones, si se desean agregar más puede seguirse un patrón similar.

Tabla 2
Nodos del prototipo

Host	Management	Tunnel	Storage	VLAN	LAN
Control-Plane	172.29.236.11	172.29.240.11	172.29.244.11		
Compute-1	172.29.236.21	172.29.240.21	172.29.244.21		
Compute-2	172.29.236.22	172.29.240.22	172.29.244.22		
Deployment	172.29.236.2				
Ceph-admin			172.29.244.41		
Ceph-mon			172.29.244.42		
Ceph-osd1			172.29.244.43		
Ceph-osd2			172.29.244.44		
Ceph-osd3			172.29.244.45		
pfSense	172.29.236.1	172.29.240.1	172.29.244.1	172.29.248.1	2 ips
Controller	172.29.236.5		172.29.236.5		
Control-plane	172.29.236.12	172.29.240.12	172.29.244.12		
Compute-1	172.29.236.23	172.29.240.23	172.29.244.23		
Compute-2	172.29.236.24	172.29.240.24	172.29.244.24		
Deployment	172.29.236.3				
Ceph-admin			172.29.244.31		

Ceph-mon			172.29.244.32		
Ceph-osd1			172.29.244.33		
Ceph-osd2			172.29.244.34		
Ceph-osd3			172.29.244.35		

La tabla 2 muestra la asignación de IPs para cada nodo del prototipo.

 Región Uno
  Región Dos
  Infraestructura

II. Requisitos iniciales

- Configurar una interfaz puente en las máquinas físicas a utilizar para el prototipo.
- Tener instalado Virtual Machine Manager.
- Contar con un navegador web instalado para poder visualizar los paneles administrativos.

III. Máquinas virtuales

Se recomienda tener máquinas físicas con al menos 32GB de RAM.

A. Inventario de máquinas virtuales

Se utilizaron 10 máquinas virtuales para el despliegue del prototipo. Se recomiendan los siguientes recursos, pero dependerá de las características del host con el que se cuente:

Tabla 3
Características de infraestructura y region.

#	Nodo	Despliegue	CPUs	Discos	RAM	NIC	Sistema operativo
1	Controller	Infraestructura	4	1 x 20 GiB	2048 MiB	1	Ubuntu Server 20.04
2	pfSense	Infraestructura	2	1 x 10 GiB	1024 MiB	2	FreeBSD 12.3
3	Control Plane	Región	8	1 x 70 GiB	16384 MiB	3	Ubuntu Server 20.04
4	Compute1	Región	8	1 x 60 Gib	5120 MiB	3	Ubuntu Server 20.04
5	Compute2	Región	8	1 x 60 Gib	5120 MiB	3	Ubuntu Server 20.04
6	Deployment	Región	2	1 x 10 GiB	2048 MiB	1	Ubuntu Server 20.04
7	Ceph-admin	Región	4	1 x 15 GiB	1536 MiB	1	Ubuntu Server 20.04
8	Ceph-mon	Región	4	1 x 15 GiB	1024 MiB	1	Ubuntu Server 20.04
9	Ceph-osd1	Región	4	1 x 15 Gib 1 x 100 GiB	700 MiB	1	Ubuntu Server 20.04
10	Ceph-osd2	Región	4	1 x 15 Gib 1 x 100 GiB	700 MiB	1	Ubuntu Server 20.04
11	Ceph-osd3	Región	4	1 x 15 Gib 1 x 100 GiB	700 MiB	1	Ubuntu Server 20.04

La tabla 3 muestra el inventario de máquinas virtuales del prototipo.

B. Imágenes

1. Ubuntu Server 20.04.

Para todas las máquinas virtuales, a excepción de pfSense, se utilizó la distro Ubuntu Server 20.04.6 LTS, también conocida como Focal Fossa. La descarga de la imagen se realiza a través del sitio <https://releases.ubuntu.com/focal/>.

La imagen puede ser descargada directamente o utilizar Torrent:

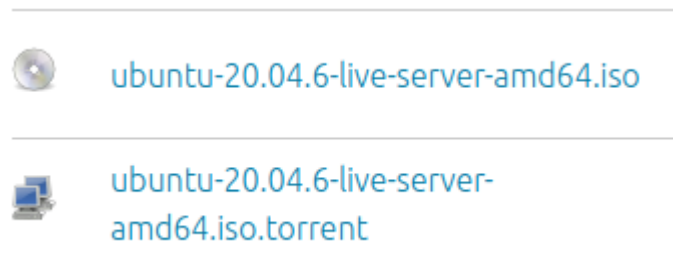


Figura 2: Opciones de descarga de Ubuntu Server 20.04.

2. pfSense.

Su imagen puede ser descargada desde el sitio oficial <https://www.pfsense.org/download/>.

Se recomiendan las siguientes opciones de descarga:

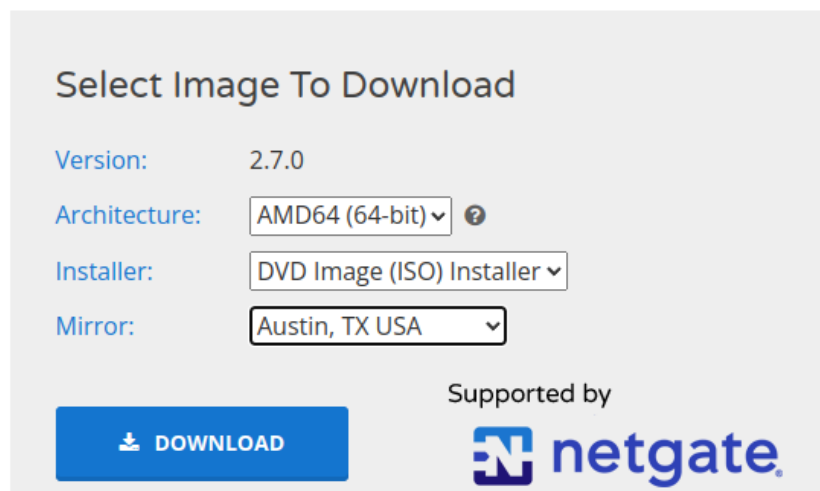


Figura 3: Opciones de descarga de pfSense.

También puede ser descargada la imagen con el siguiente enlace:

```
https://atxfiles.netgate.com/mirror/downloads/pfSense-CE-2.7.0-RELEASE-amd64.iso.gz
```

Descomprimir utilizando el siguiente comando:

```
gzip -d pfSense-CE-2.7.0-RELEASE-amd64.iso.gz
```

C. Creación de máquinas virtuales en VMM.

Los pasos para la creación de las máquinas virtuales se detallan en las siguientes secciones. Cabe aclarar que los recursos mostrados en las imágenes pueden ser diferentes a los mostrados en el cuadro de inventario.

1. Instalando paquetes.

1. Ejecutar el siguiente comando:

```
apt-get install bridge-utils cpu-checker libvirt-clients libvirt-daemon qemu qemu-kvm
```

2. Para comprobar si el procesador soporta virtualización puede ejecutarse el siguiente comando:

```
kvm-ok
```

3. Para instalar la herramienta gráfica Virtual Machine Manager puede ejecutarse el siguiente comando:

```
sudo apt install virt-manager
```

2. Máquinas virtuales para pfSense.

Considerar que la red LAN puede ser diferente, dependiendo de donde se encuentre el host.

1. Abrir VMM y dar clic en el siguiente botón “Create a new virtual machine” (Crear nueva máquina virtual):

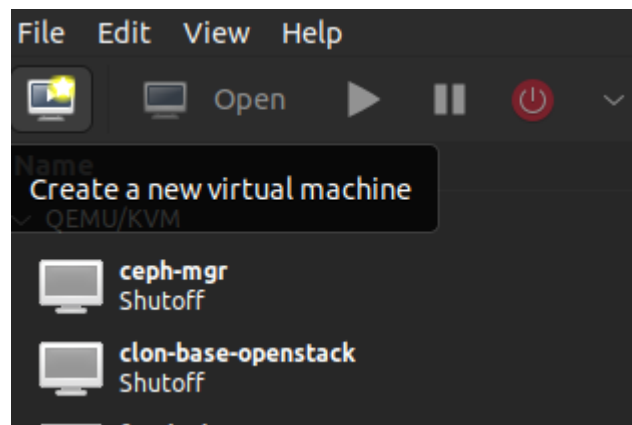


Figura 4: Botón para crear máquina virtual en VMM.

2. En VMM creó una nueva máquina virtual, seleccionando imagen ISO como opción:

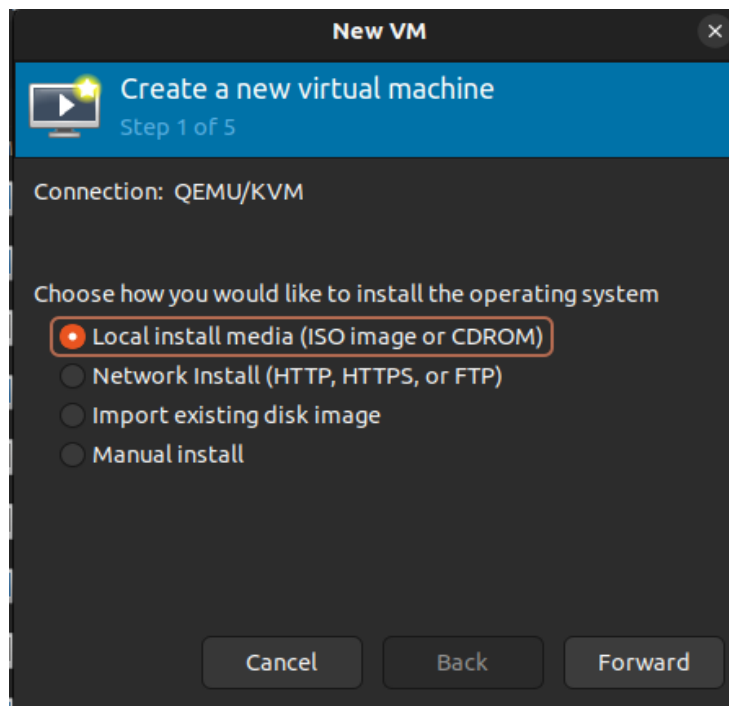


Figura 5: Listado de opciones para crear máquina virtual en VMM.

3. Escogemos el archivo descomprimido:

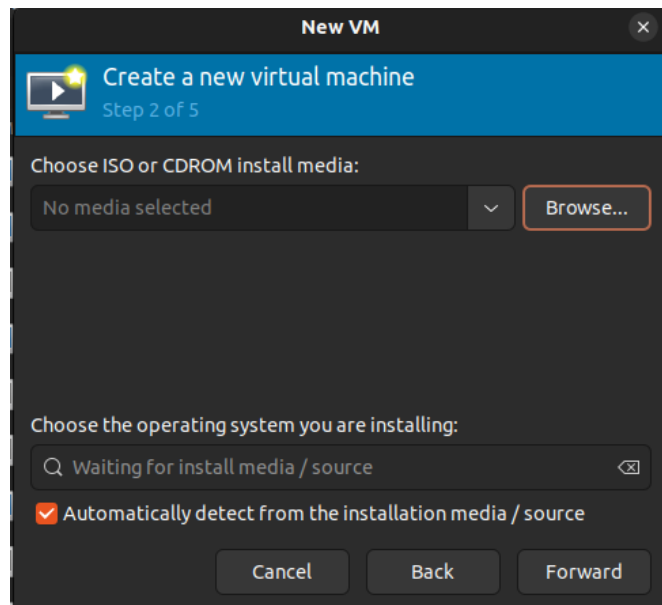


Figura 6: Formulario para ingresar imagen en VMM.

4. Desactivamos el checkbox de detección automática y especificamos FreeBSD 12.4 para el sistema operativo:

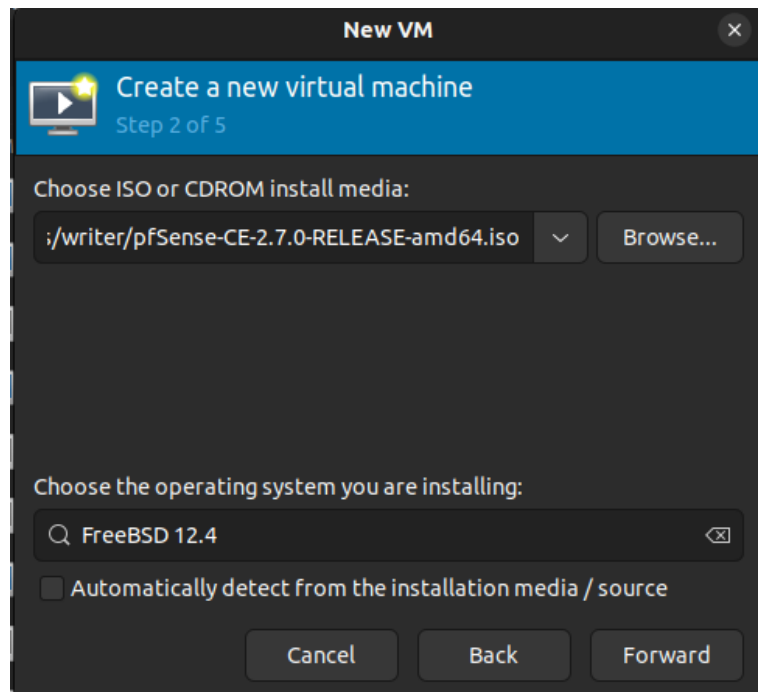


Figura 7: Vista del formulario para ingresar imagen una vez seleccionada en VMM.

5. Especificamos la RAM y los núcleos de la máquina virtual:

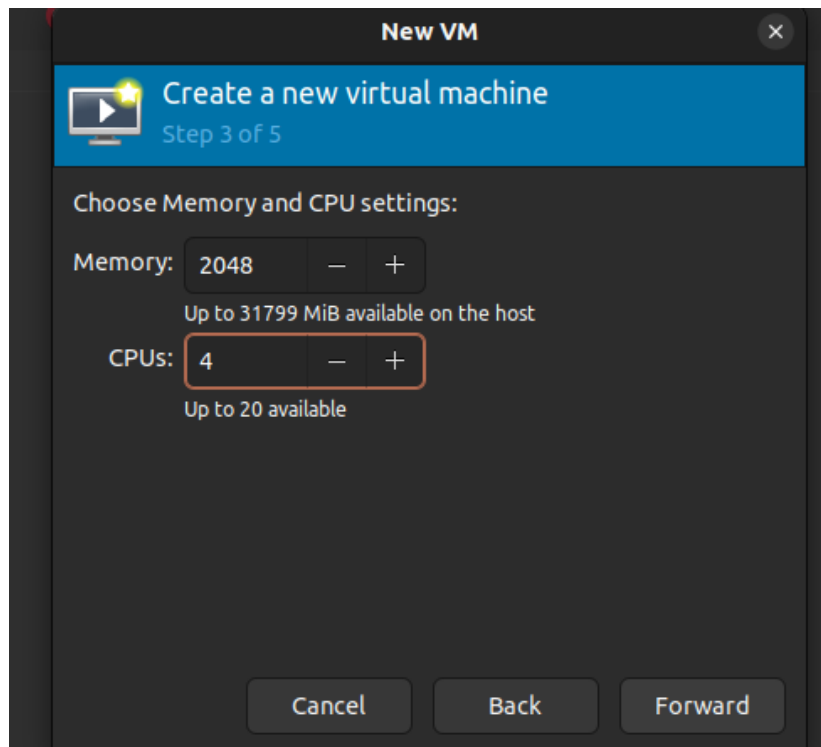


Figura 8: Formulario para ingresar datos de CPU y memoria en VMM.

6. Asignamos el espacio del disco:

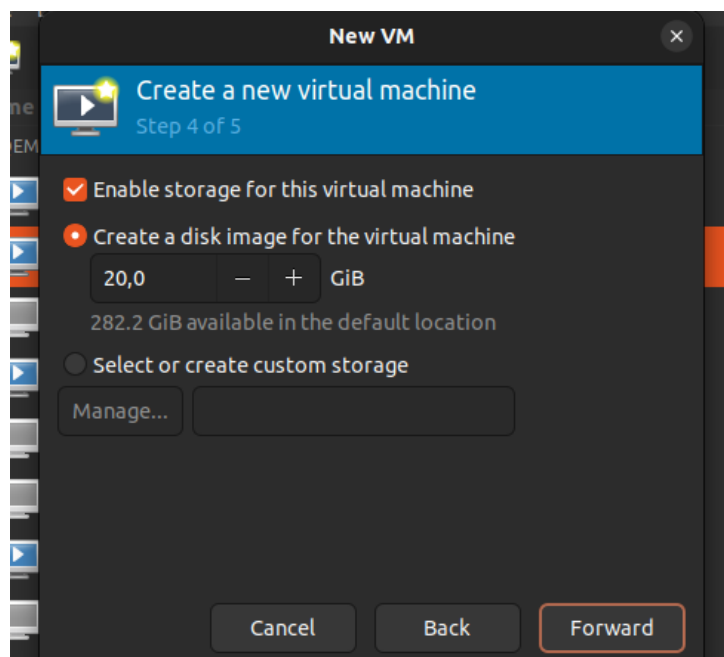


Figura 9: Formulario para ingresar el tamaño del disco en VMM.

7. Seleccionamos la opción referente a personalizar la configuración antes de instalar:

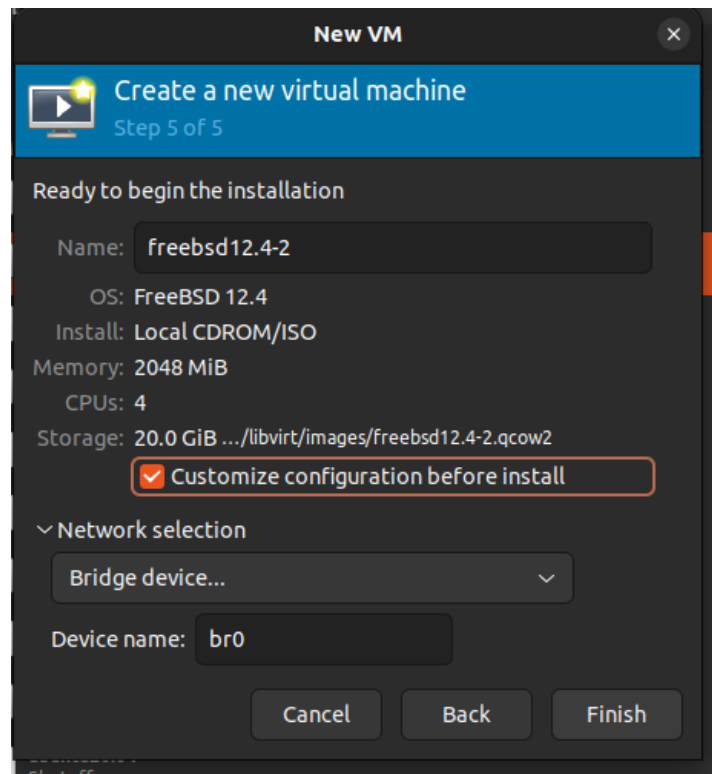


Figura 10: Formulario para ingresar el nombre e interfaz de red en VMM.

8. Damos clic en el botón agregar un nuevo hardware:

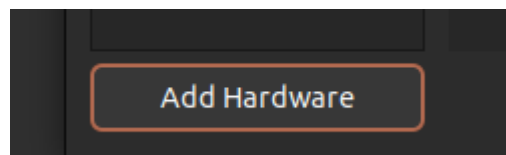


Figura 11: Botón para agregar nuevo hardware en VMM.

9. Agregamos una nueva interfaz de red que tenga como red un dispositivo puente configurado con la red LAN:

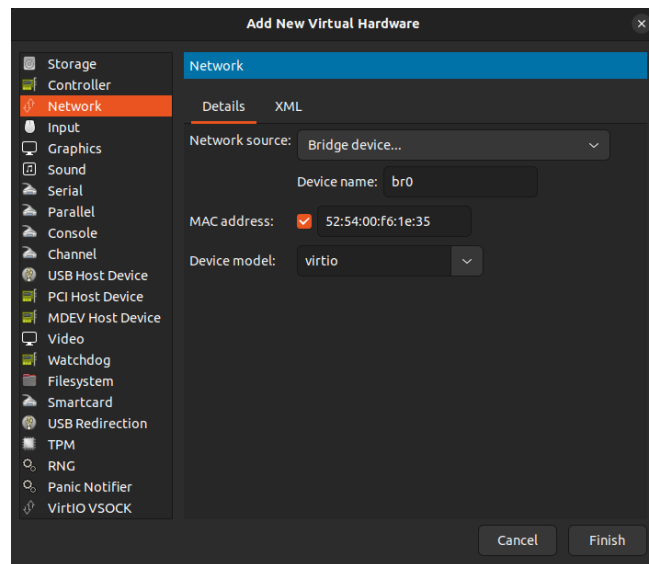


Figura 12: Formulario para ingresar nueva interfaz de red en VMM.

10. Hacemos clic en el botón de iniciar instalación y esperamos unos segundos para el autoboot:

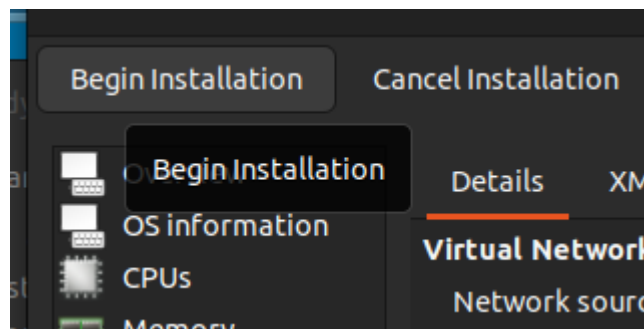


Figura 13: Botón para iniciar el proceso de instalación en VMM.

11. Damos Enter para aceptar:

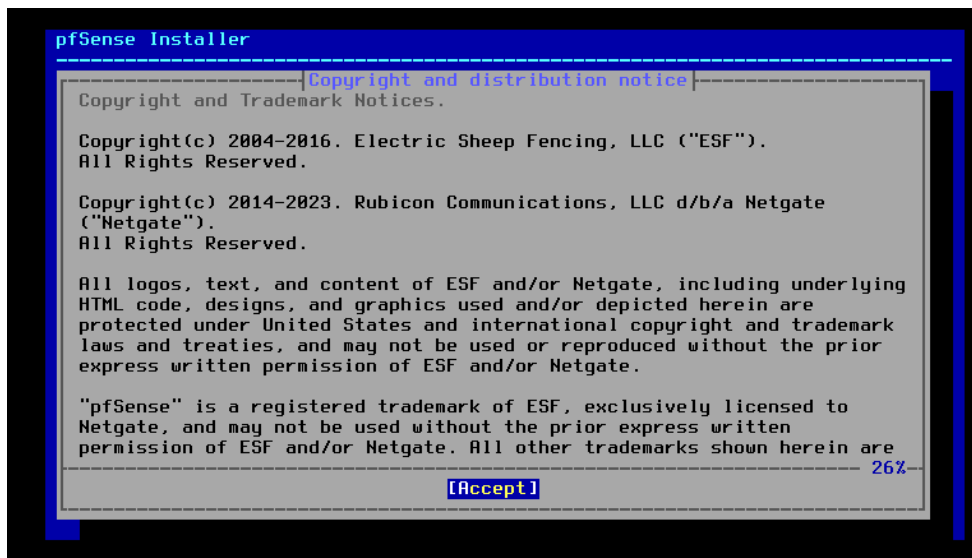


Figura 14: Pantalla de derechos reservados de pfSense.

12. Seleccionamos la opción Instalar y damos Enter:

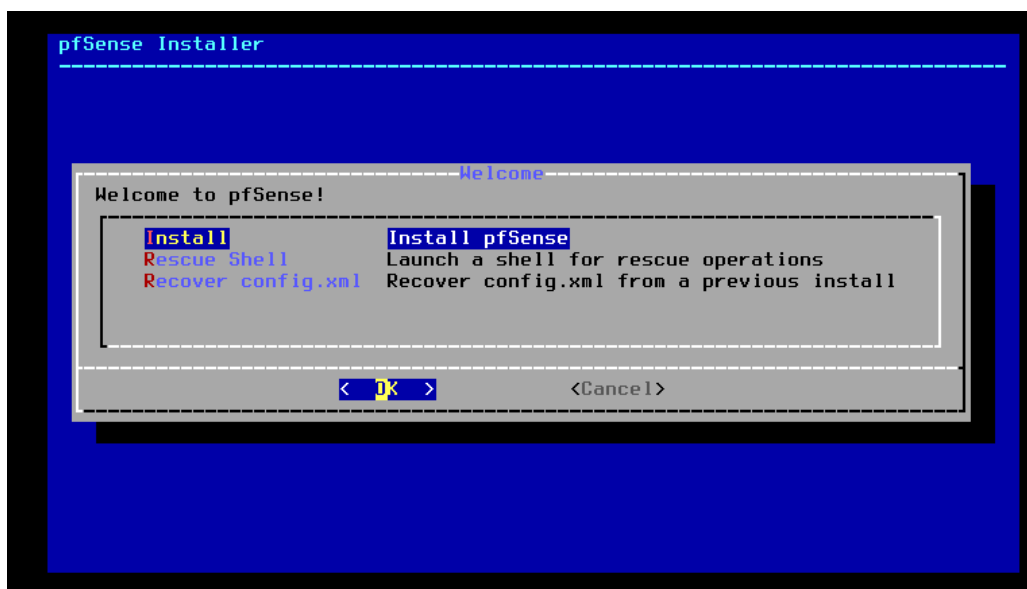


Figura 15: Pantalla de opciones del instalador de pfSense.

13. Seleccionamos la opción Auto y damos Enter:

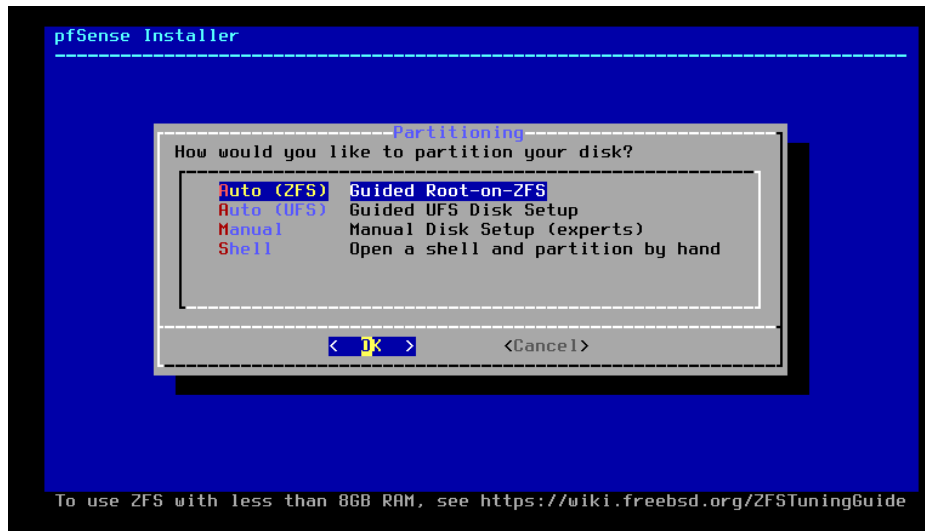


Figura 16: Pantalla de opciones de partición para la instalación de pfSense.

14. Seleccionamos Instalar y damos Enter:

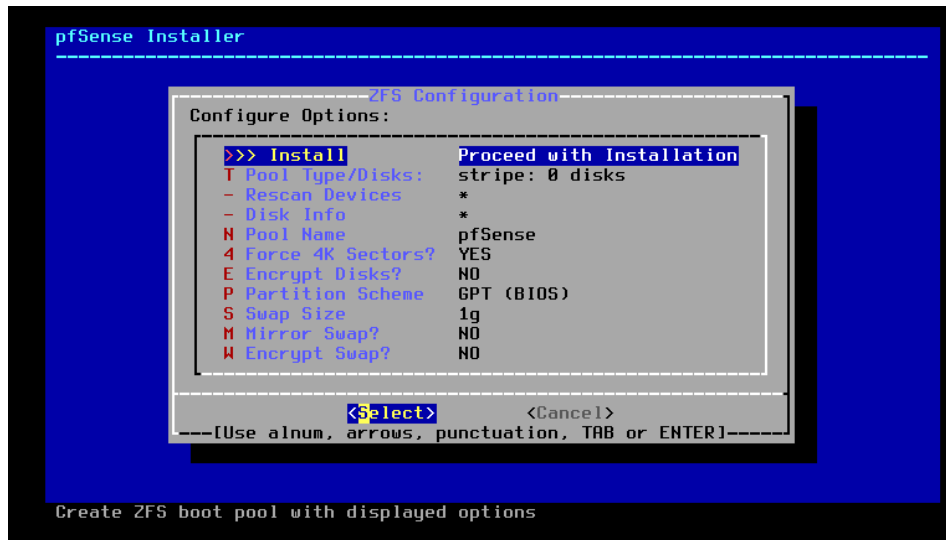


Figura 17: Pantalla de opciones de configuración de ZFS para la instalación de pfSense.

15. Seleccionamos la opción Stripe y damos Enter:

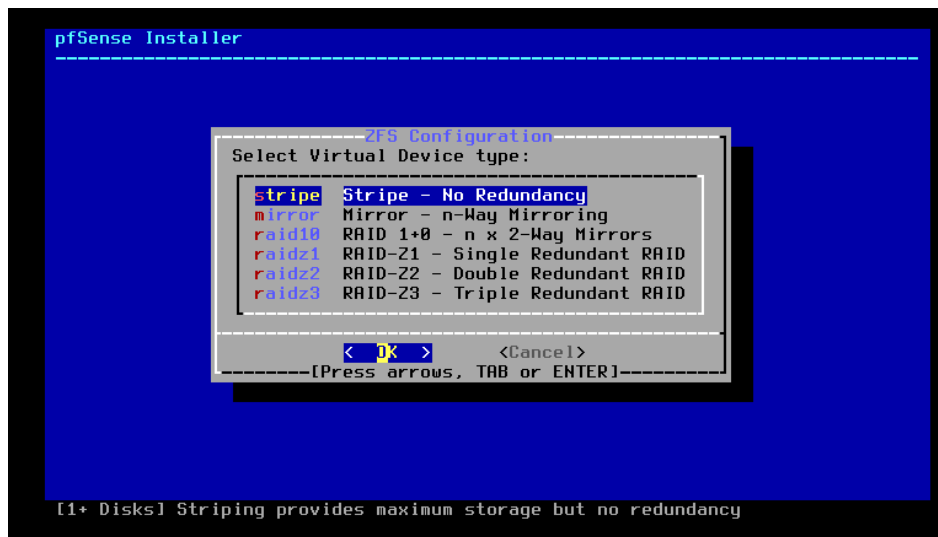


Figura 18: Pantalla de opciones de RAID para la instalación de pfSense.

16. Seleccionamos el único dispositivo con barra espaciadora y luego damos Enter:

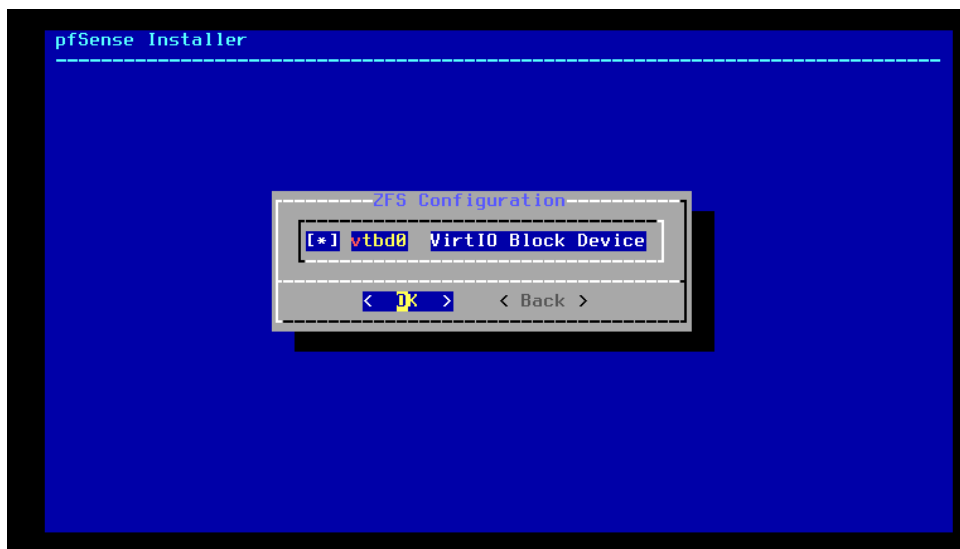


Figura 19: Pantalla de opciones de disco para la instalación de pfSense.

17. Confirmamos la configuración del disco:

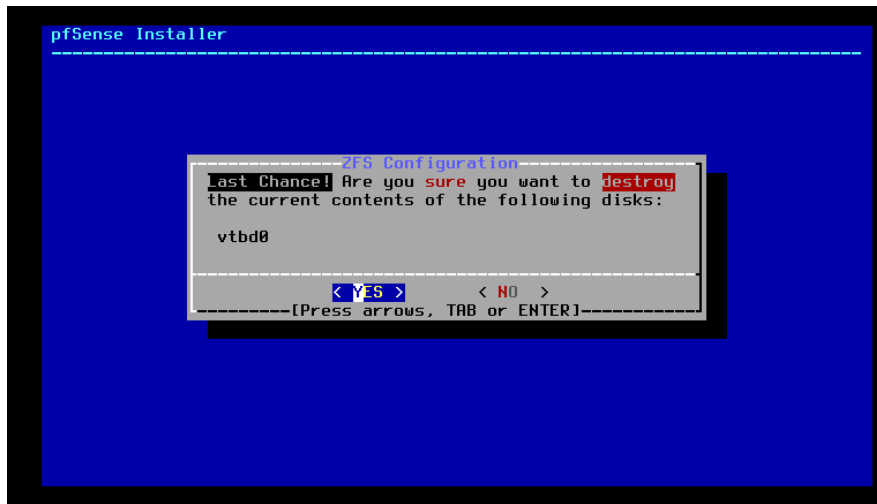


Figura 20: Pantalla de confirmación para la instalación de pfSense.

18. Observamos que se inicia un proceso. Esperamos a que finalice.

19. Se nos indica que la instalación ha sido completada. Procedemos a reiniciar la máquina:

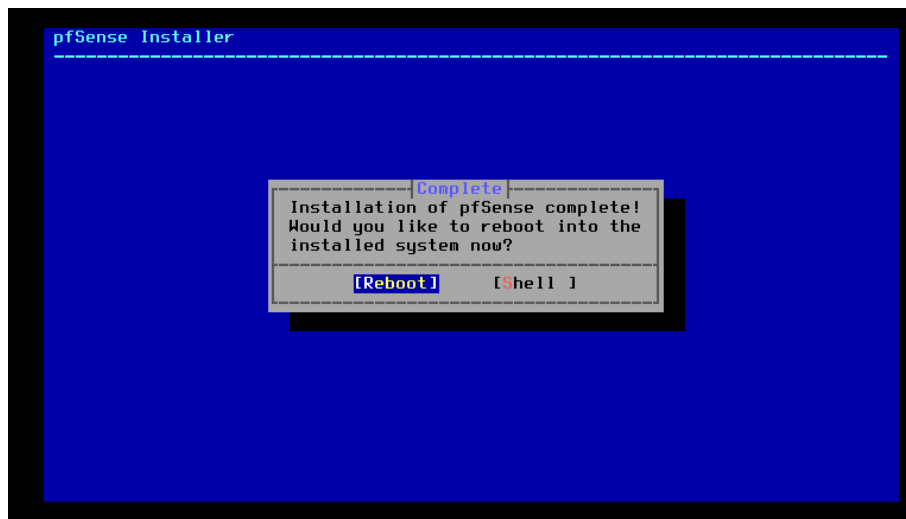


Figura 21: Pantalla de notificación de instalación satisfactoria de pfSense.

20. En este momento, donde se nos pide configurar las VLANS, ingresamos 'N' y damos

Enter:

```
Initializing..... done.
Starting device manager (devd)...done.
Loading configuration...done.
Updating configuration.....Migrating System Memory RRD file to new format
.done.
Warning: Configuration references interfaces that do not exist: em0 em1

Network interface mismatch -- Running interface assignment option.
vtnet0: link state changed to UP
vtnet1: link state changed to UP

Valid interfaces are:

vtnet0 52:54:00:3e:d0:d5 (down) VirtIO Networking Adapter
vtnet1 52:54:00:f6:1e:35 (down) VirtIO Networking Adapter

Do VLANs need to be set up first?
If VLANs will not be used, or only for optional interfaces, it is typical to
say no here and use the webConfigurator to configure VLANs later, if required.

Should VLANs be set up now [y|n]? 2023-09-12T21:11:21.697252+00:00 - php-fpm 376
- - /rc.linkup: Ignoring link event during boot sequence.
2023-09-12T21:11:21.698105+00:00 - php-fpm 375 - - /rc.linkup: Ignoring link eve
nt during boot sequence.
█
```

Figura 22: Pantalla de consola para creación vlans en pfSense.

21. Seleccionamos como interfaz WAN la vtnet0:

```
Network interface mismatch -- Running interface assignment option.
vtnet0: link state changed to UP
vtnet1: link state changed to UP

Valid interfaces are:

vtnet0 52:54:00:3e:d0:d5 (down) VirtIO Networking Adapter
vtnet1 52:54:00:f6:1e:35 (down) VirtIO Networking Adapter

Do VLANs need to be set up first?
If VLANs will not be used, or only for optional interfaces, it is typical to
say no here and use the webConfigurator to configure VLANs later, if required.

Should VLANs be set up now [y|n]? 2023-09-12T21:11:21.697252+00:00 - php-fpm 376
- - /rc.linkup: Ignoring link event during boot sequence.
2023-09-12T21:11:21.698105+00:00 - php-fpm 375 - - /rc.linkup: Ignoring link eve
nt during boot sequence.
n

If the names of the interfaces are not known, auto-detection can
be used instead. To use auto-detection, please disconnect all
interfaces before pressing 'a' to begin the process.

Enter the WAN interface name or 'a' for auto-detection
(vtnet0 vtnet1 or a): vtnet0█
```

Figura 23: Pantalla de consola para elección de interfaces en pfSense.

22. Seleccionamos como interfaz LAN la vtnet1:

```
Valid interfaces are:

vtnet0 52:54:00:3e:d0:d5 (down) VirtIO Networking Adapter
vtnet1 52:54:00:f6:1e:35 (down) VirtIO Networking Adapter

Do VLANs need to be set up first?
If VLANs will not be used, or only for optional interfaces, it is typical to
say no here and use the webConfigurator to configure VLANs later, if required.

Should VLANs be set up now [y/n]? 2023-09-12T21:11:21.697252+00:00 - php-fpm 376
- - /rc.linkup: Ignoring link event during boot sequence.
2023-09-12T21:11:21.698105+00:00 - php-fpm 375 - - /rc.linkup: Ignoring link eve
nt during boot sequence.
n

If the names of the interfaces are not known, auto-detection can
be used instead. To use auto-detection, please disconnect all
interfaces before pressing 'a' to begin the process.

Enter the WAN interface name or 'a' for auto-detection
(vtnet0 vtnet1 or a): vtnet0

Enter the LAN interface name or 'a' for auto-detection
NOTE: this enables full Firewalling/NAT mode.
(vtnet1 a or nothing if finished): vtnet1
```

Figura 24: Pantalla de consola para elección de interfaces en pfSense.

23. Confirmamos la configuración ingresada:

```
say no here and use the webConfigurator to configure VLANs later, if required.

Should VLANs be set up now [y/n]? 2023-09-12T21:11:21.697252+00:00 - php-fpm 376
- - /rc.linkup: Ignoring link event during boot sequence.
2023-09-12T21:11:21.698105+00:00 - php-fpm 375 - - /rc.linkup: Ignoring link eve
nt during boot sequence.
n

If the names of the interfaces are not known, auto-detection can
be used instead. To use auto-detection, please disconnect all
interfaces before pressing 'a' to begin the process.

Enter the WAN interface name or 'a' for auto-detection
(vtnet0 vtnet1 or a): vtnet0

Enter the LAN interface name or 'a' for auto-detection
NOTE: this enables full Firewalling/NAT mode.
(vtnet1 a or nothing if finished): vtnet1

The interfaces will be assigned as follows:

WAN -> vtnet0
LAN -> vtnet1

Do you want to proceed [y/n]? y
```

Figura 25: Pantalla confirmar selección de interfaces en pfSense.

24. Esperamos que se configure:

```
n

If the names of the interfaces are not known, auto-detection can
be used instead. To use auto-detection, please disconnect all
interfaces before pressing 'a' to begin the process.

Enter the WAN interface name or 'a' for auto-detection
(vtnet0 vtnet1 or a): vtnet0

Enter the LAN interface name or 'a' for auto-detection
NOTE: this enables full Firewalling/NAT mode.
(vtnet1 a or nothing if finished): vtnet1

The interfaces will be assigned as follows:

WAN -> vtnet0
LAN -> vtnet1

Do you want to proceed [y/n]? y

Writing configuration...done.
One moment while the settings are reloading... done!
Configuring loopback interface...done.
Configuring LAN interface...done.
Configuring WAN interface...█
```

Figura 26: Pantalla de procedimientos realizados en la configuración de interfaces.

25. Se nos presenta la siguiente pantalla mostrándonos la configuración que recibieron las interfaces:

```
Starting syslog...done.
Starting CRON... done.
pfSense 2.7.0-RELEASE amd64 Wed Jun 28 03:53:34 UTC 2023
Bootup complete

FreeBSD/amd64 (pfSense.home.arp) (ttyv0)

QEMU Guest - Netgate Device ID: fa197762f7c69c606522

*** Welcome to pfSense 2.7.0-RELEASE (amd64) on pfSense ***

WAN (wan)      -> vtnet0      -> v4/DHCP4: 192.168.16.141/24
LAN (lan)      -> vtnet1      -> v4: 192.168.1.1/24

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults  13) Update from console
5) Reboot system              14) Enable Secure Shell (sshd)
6) Halt system                 15) Restore recent configuration
7) Ping host                   16) Restart PHP-FPM
8) Shell

Enter an option: █
```

Figura 27: Pantalla de opciones en pfSense.

26. Seleccionamos la opción 2 para configurar la dirección IP de la interfaz LAN:

```
QEMU Guest - Netgate Device ID: fa197762f7c69c606522
*** Welcome to pfSense 2.7.0-RELEASE (amd64) on pfSense ***

WAN (wan)      -> vtnet0      -> v4/DHCP4: 192.168.16.141/24
LAN (lan)      -> vtnet1      -> v4: 192.168.1.1/24

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults  13) Update from console
5) Reboot system              14) Enable Secure Shell (sshd)
6) Halt system                 15) Restore recent configuration
7) Ping host                   16) Restart PHP-FPM
8) Shell

Enter an option: 2

Available interfaces:

1 - WAN (vtnet0 - dhcp, dhcp6)
2 - LAN (vtnet1 - static)

Enter the number of the interface you wish to configure: 2
```

Figura 28: Pantalla para elegir la interfaz a modificar en pfSense.

27. Podemos configurar a través de DHCP ingresando ‘y’:

```
WAN (wan)      -> vtnet0      -> v4/DHCP4: 192.168.16.141/24
LAN (lan)      -> vtnet1      -> v4: 192.168.1.1/24

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults  13) Update from console
5) Reboot system              14) Enable Secure Shell (sshd)
6) Halt system                 15) Restore recent configuration
7) Ping host                   16) Restart PHP-FPM
8) Shell

Enter an option: 2

Available interfaces:

1 - WAN (vtnet0 - dhcp, dhcp6)
2 - LAN (vtnet1 - static)

Enter the number of the interface you wish to configure: 2

Configure IPv4 address LAN interface via DHCP? (y/n) y
Configure IPv6 address LAN interface via DHCP6? (y/n) y
```

Figura 29: Pantalla de opciones de interfaz a modificar en pfSense.

28. Ingresamos 'y' para utilizar el protocolo http:

```
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults  13) Update from console
5) Reboot system             14) Enable Secure Shell (sshd)
6) Halt system               15) Restore recent configuration
7) Ping host                 16) Restart PHP-FPM
8) Shell

Enter an option: 2

Available interfaces:

1 - WAN (vtnet0 - dhcp, dhcp6)
2 - LAN (vtnet1 - static)

Enter the number of the interface you wish to configure: 2

Configure IPv4 address LAN interface via DHCP? (y/n) y

Configure IPv6 address LAN interface via DHCP6? (y/n) y
Disabling IPv4 DHCPD...
Disabling IPv6 DHCPD...

Do you want to revert to HTTP as the webConfigurator protocol? (y/n) y
```

Figura 30: Pantalla para confirmar modificación de interfaz en pfSense.

29. Reiniciamos la máquina virtual.

30. Ahora podemos observar que tenemos una nueva IP para la vtnet1:

```
Starting syslog...done.
Starting CRON... done.
pfSense 2.7.0-RELEASE amd64 Wed Jun 28 03:53:34 UTC 2023
Bootup complete

FreeBSD/amd64 (pfSense.home.arp) (ttyv0)

QEMU Guest - Netgate Device ID: fa197762f7c69c606522

*** Welcome to pfSense 2.7.0-RELEASE (amd64) on pfSense ***

WAN (wan)      -> vtnet0      -> v4/DHCP4: 192.168.16.141/24
LAN (lan)      -> vtnet1      -> v4/DHCP4: 192.168.16.79/24

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults  13) Update from console
5) Reboot system             14) Enable Secure Shell (sshd)
6) Halt system               15) Restore recent configuration
7) Ping host                 16) Restart PHP-FPM
8) Shell

Enter an option: █
```

Figura 31: Pantalla mostrada una vez realizada la modificación en pfSense.

31. En un navegador, visitamos, en nuestro caso, la dirección 192.168.16.79. Deberán adaptar de acuerdo a la asignada en su ambiente. Utilizar para el usuario *admin* y para la contraseña *pfSense*:

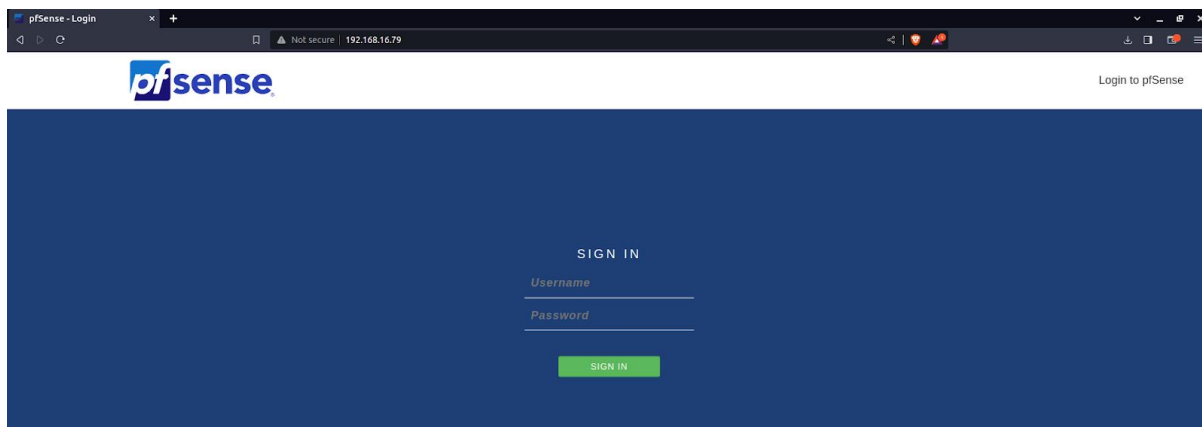


Figura 32: Formulario de Login del panel de pfSense.

32. Observamos el dashboard y damos clic en Next hasta llegar al paso 2:

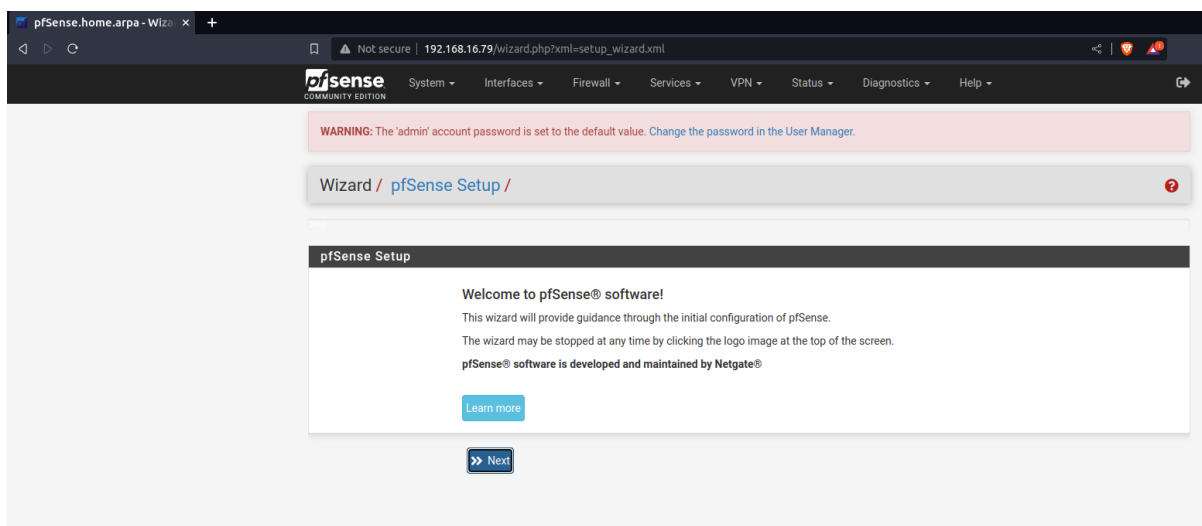


Figura 33: Wizard de configuración de pfSense.

33. Pueden configurar con los siguientes valores para el paso 2:

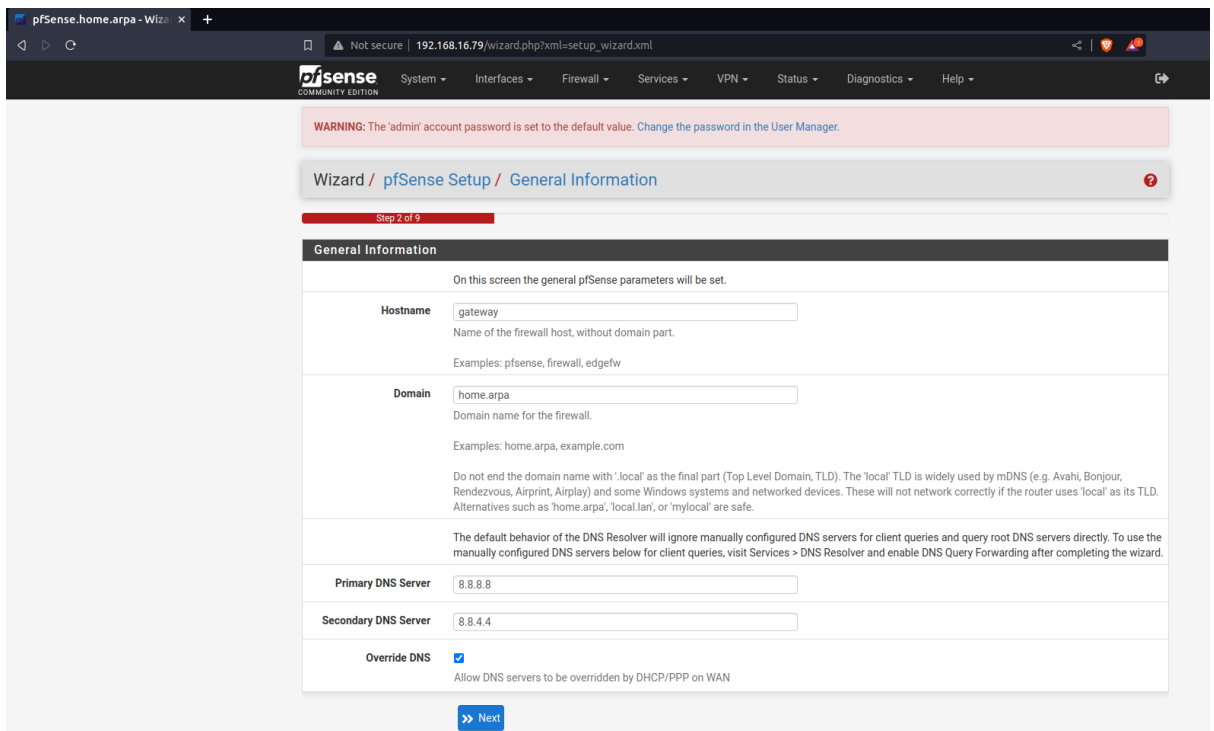


Figura 34: Configuraciones generales del Wizard de pfSense.

34. Escogemos la opción deseada para el siguiente paso:

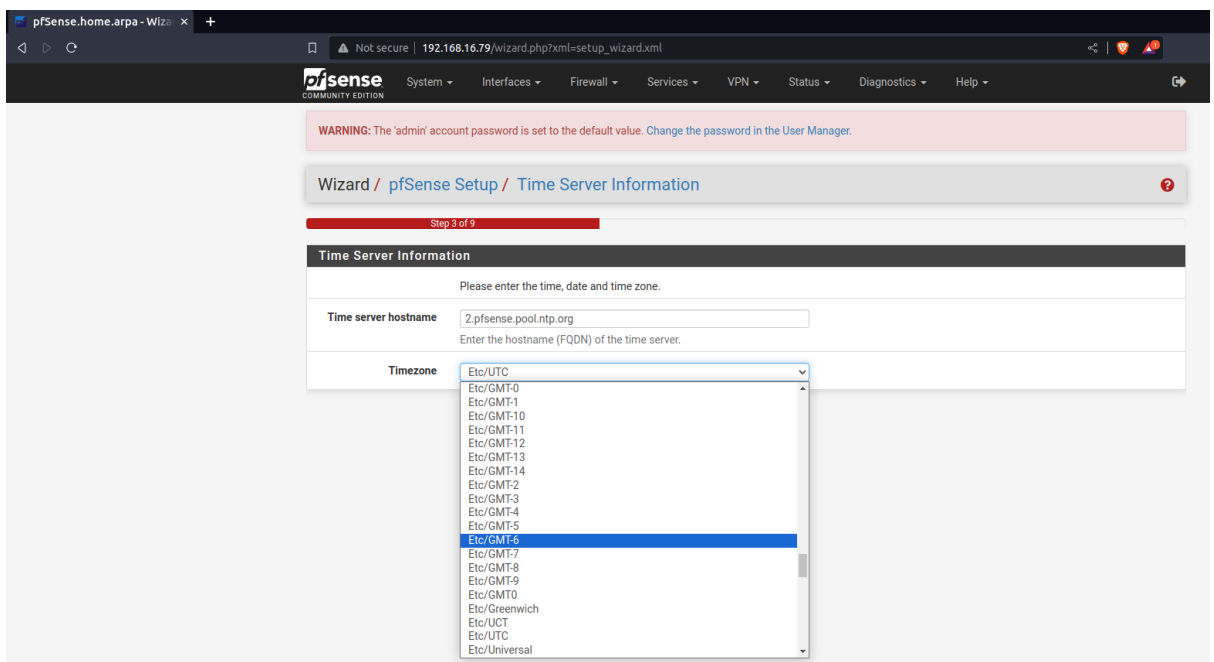


Figura 35: Configuraciones NTP de pfSense.

35. Para los pasos 4 no debe modificarse ninguna opción:

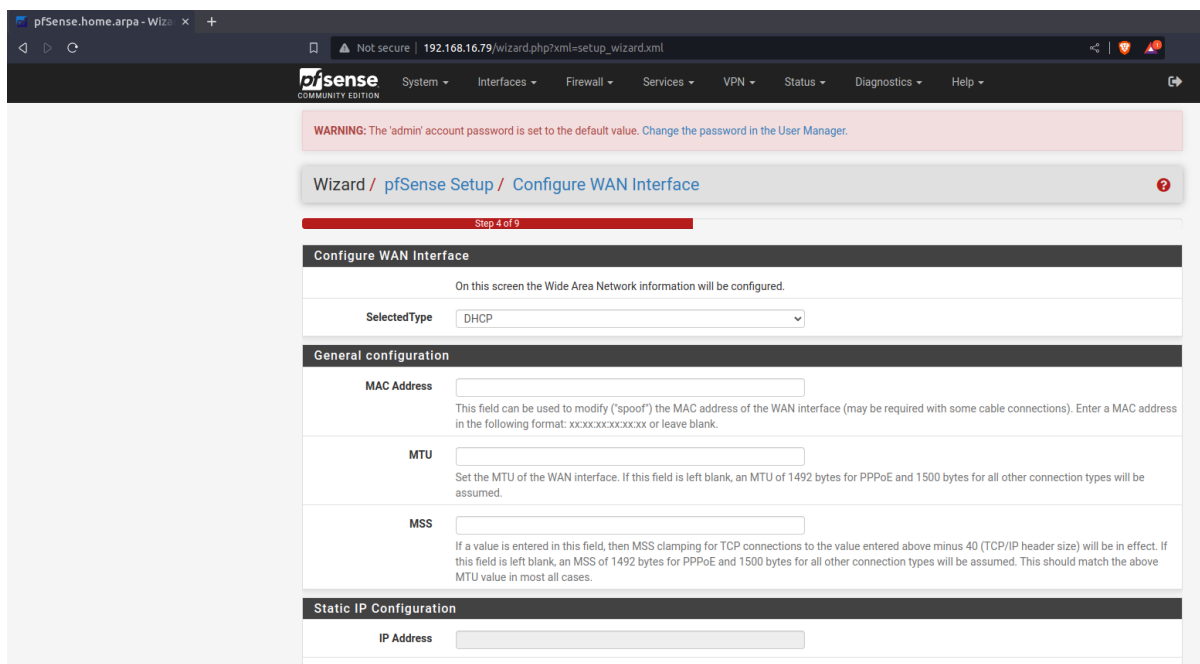


Figura 36: Configuraciones de interfaz WAN de pfSense.

36. Modificar para el paso 5:

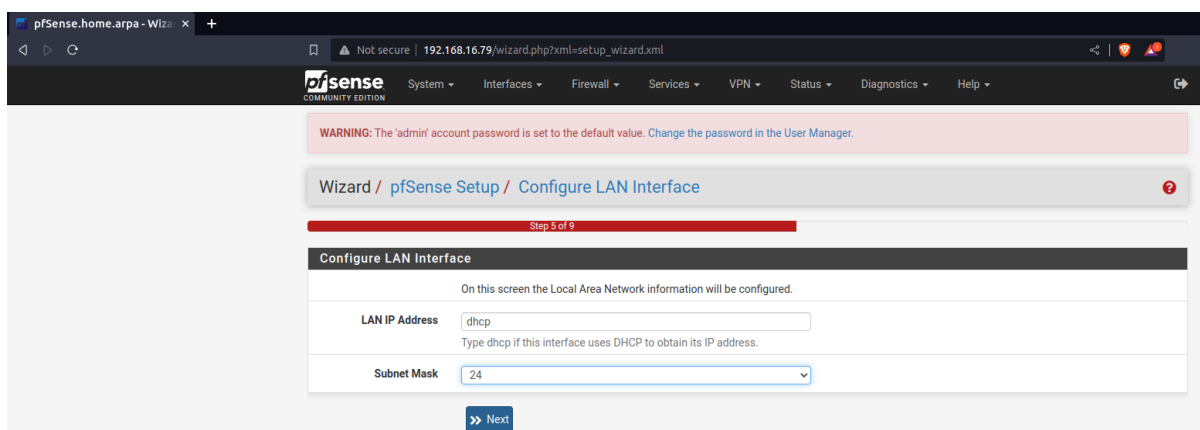


Figura 37: Configuraciones de interfaz LAN de pfSense.

37. Cambiamos la contraseña para nuestro usuario admin:

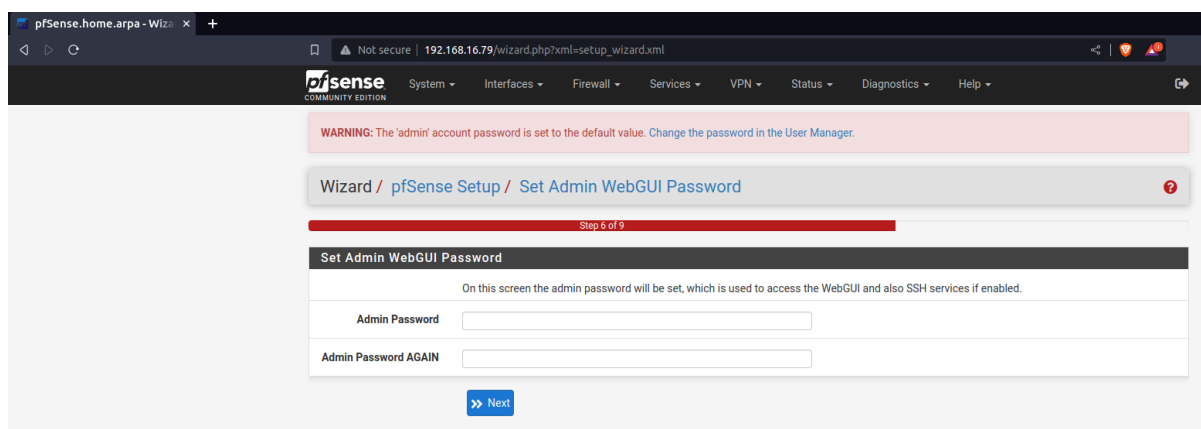


Figura 38: Configuraciones de autenticación del admin de pfSense.

38. Recargamos la configuración y posteriormente podremos ver que el wizard se ha completado:

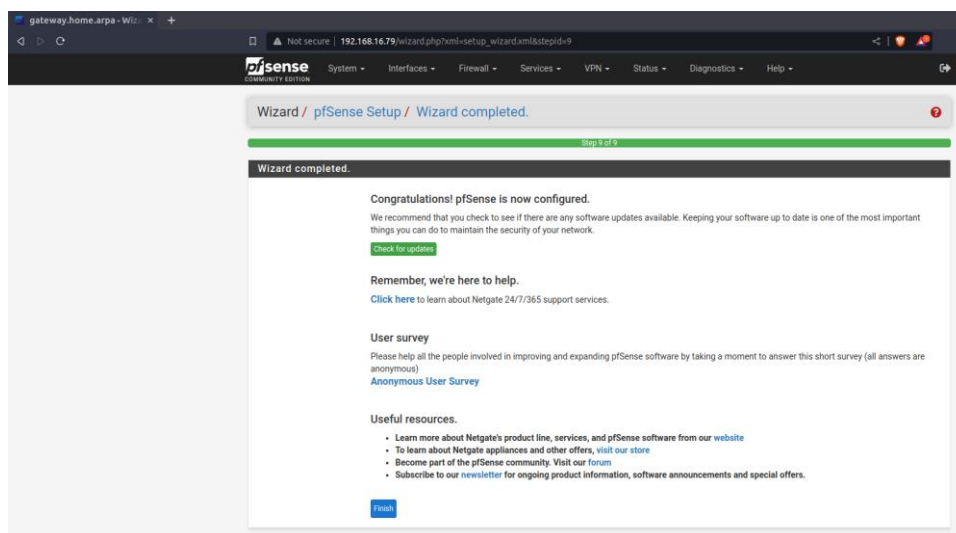


Figura 39: Pantalla de configuraciones realizadas por el Wizard de pfSense.

3. Máquinas virtuales con Ubuntu.

1. Abrir VMM y dar clic en el siguiente botón “Create a new virtual machine” (Crear nueva máquina virtual):

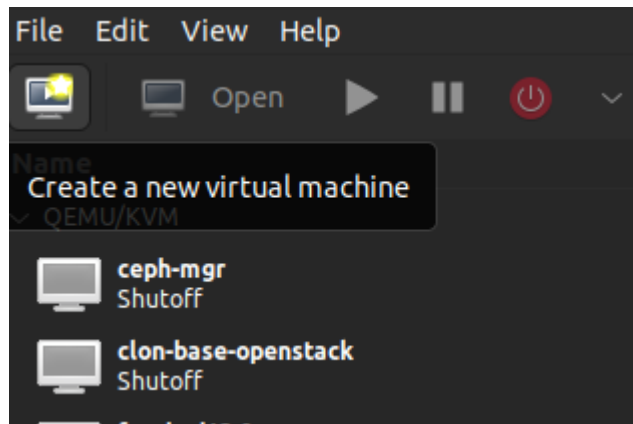


Figura 40: Botón para crear máquina virtual en VMM.

2. Damos clic en “Browse” para buscar la imagen de Ubuntu y esperamos a que sea reconocido el sistema operativo. Una vez sea reconocido daremos clic en “Forward”:

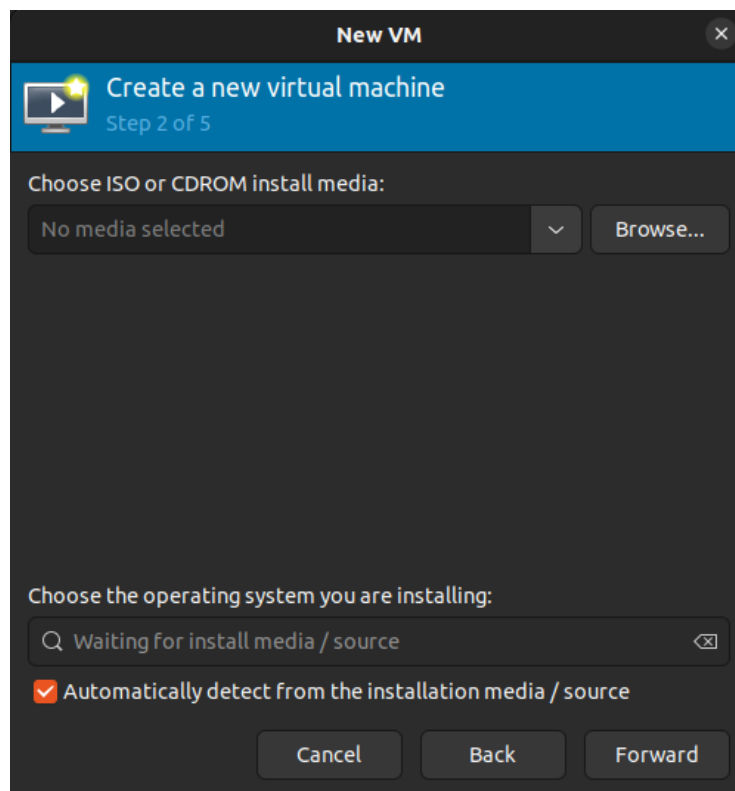


Figura 41: Formulario para seleccionar imagen en VMM.

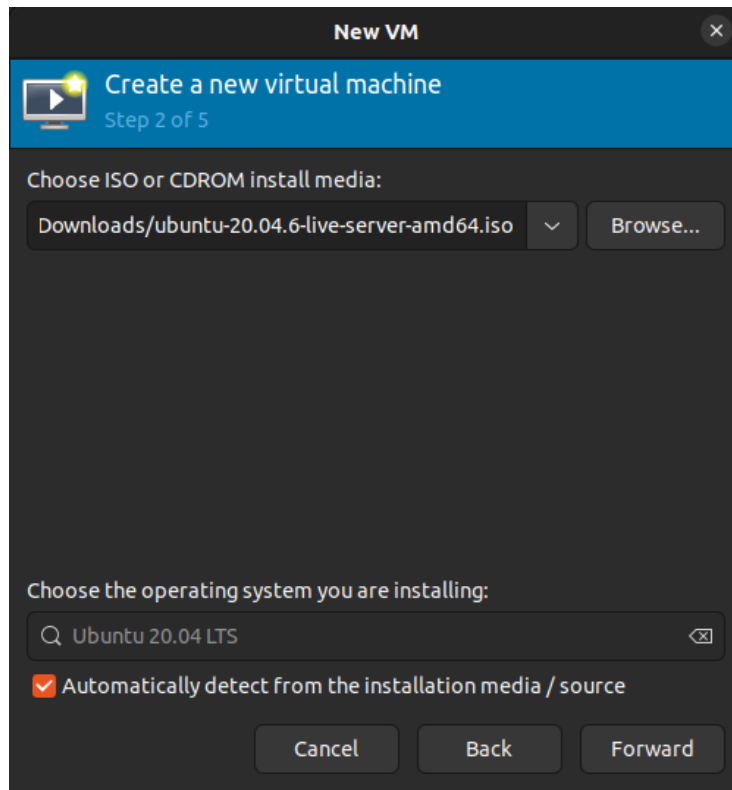


Figura 42: Formulario para seleccionar imagen una vez elegida en VMM.

3. Indicamos la memoria RAM y la cantidad de CPUs que tendrá la máquina virtual, y damos clic en “Forward”:

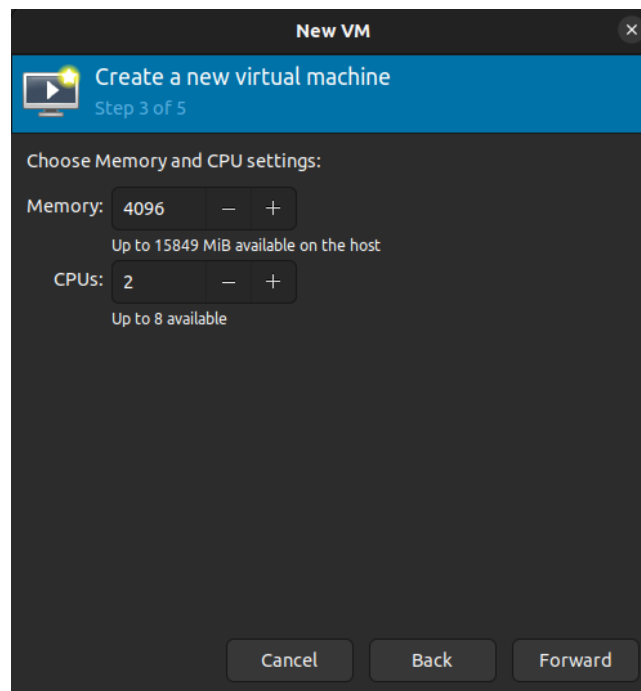


Figura 43: Formulario para seleccionar imagen en VMM.

4. Configuramos el espacio del disco de la siguiente manera, especificando la cantidad de acuerdo al nodo que estemos trabajando. Damos clic en “Forward”:

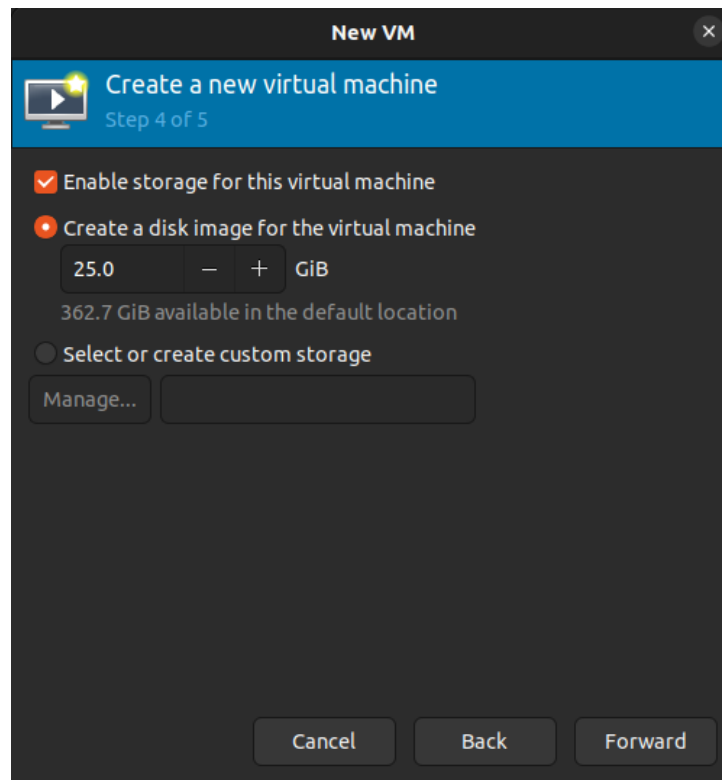


Figura 44: Formulario para ingresar tamaño de disco en VMM.

5. Nombramos al nodo en cuestión y marcamos la opción “Customize configuration before install”. Además nos aseguramos que “Network Selection” tenga la opción “Bridge device...” y el nombre del bridge que ha sido configurado para la máquina host. Damos clic en “Finish”:

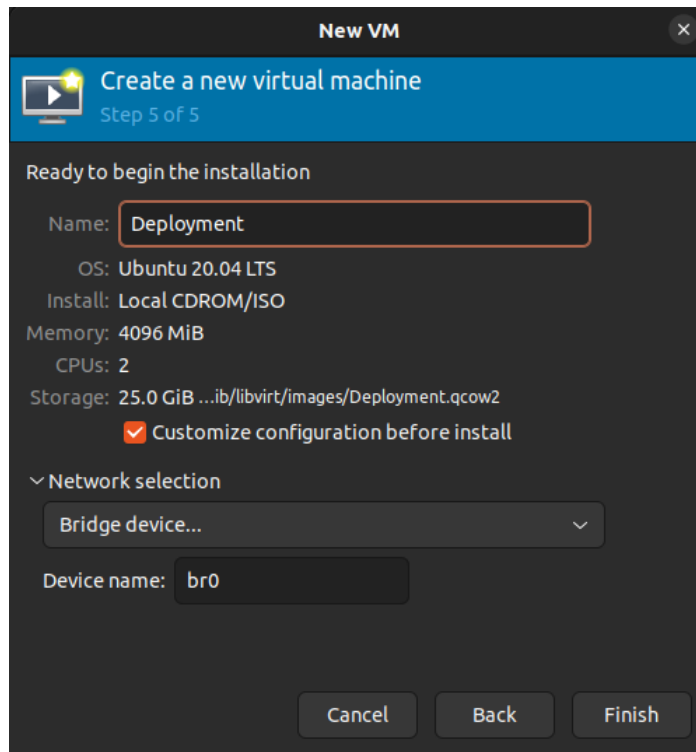


Figura 45: Formulario para ingresar nombre de instancia e interfaz de red en VMM.

6. Se nos presentará una pantalla donde podremos agregar, por ejemplo, nuevos discos e interfaces de red.
7. Si el nodo necesita dispositivos adicionales, dar clic en “Add Hardware” en la esquina inferior izquierda:

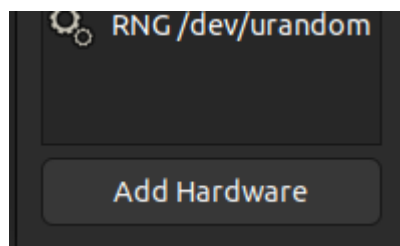


Figura 46: Botón para agregar nuevo hardware en VMM.

- Si es un disco nuevo, ajustar la capacidad que tendrá y dar clic en “Finish”:

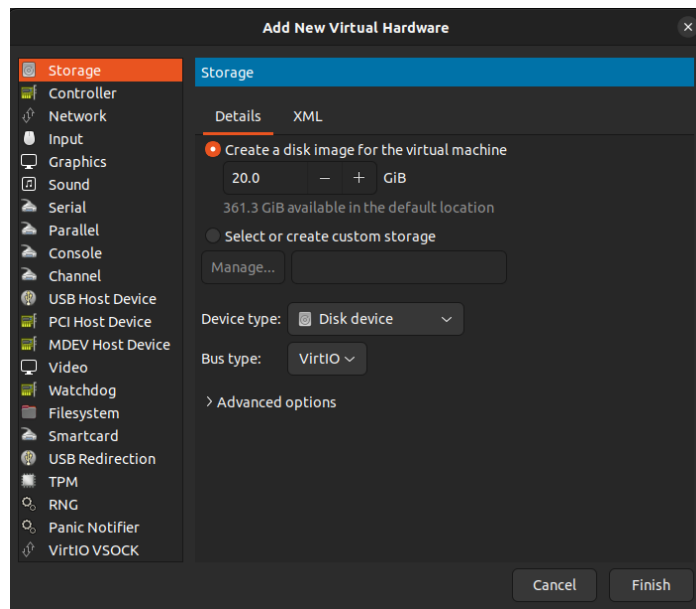


Figura 47: Formulario para agregar un nuevo disco a instancia en VMM.

- Si es una interfaz de red nueva, especificar la fuente y dar clic en “Finish”:

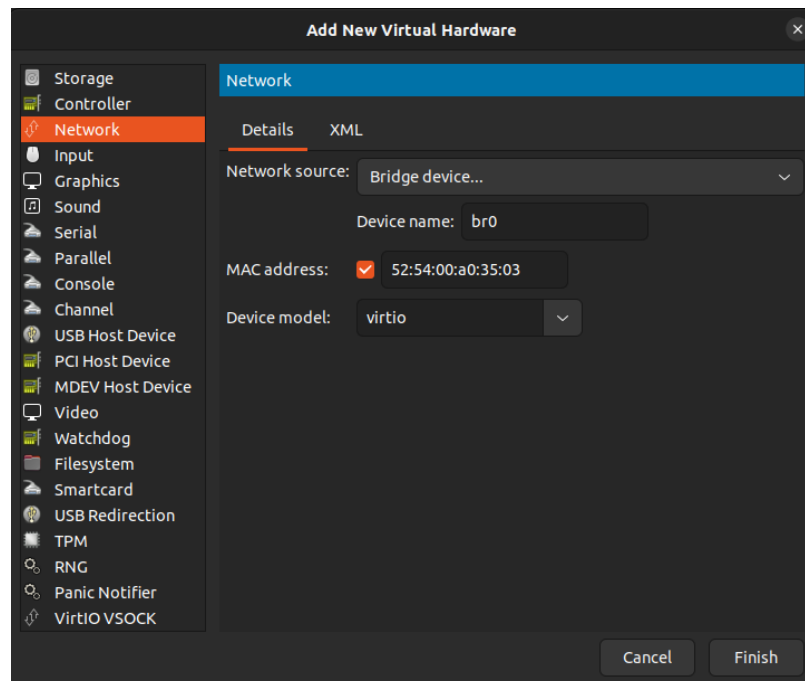


Figura 48: Formulario para agregar una nueva interfaz de red a instancia en VMM.

10. Dar clic en el botón “Begin Installation” en la esquina superior izquierda:

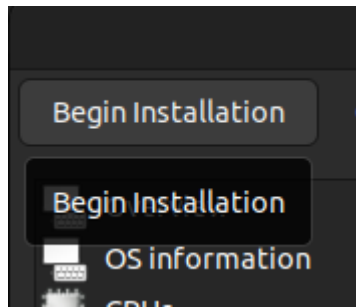


Figura 49: Botón para iniciar instalación en VMM.

11. Esperamos que se realicen las comprobaciones iniciales:

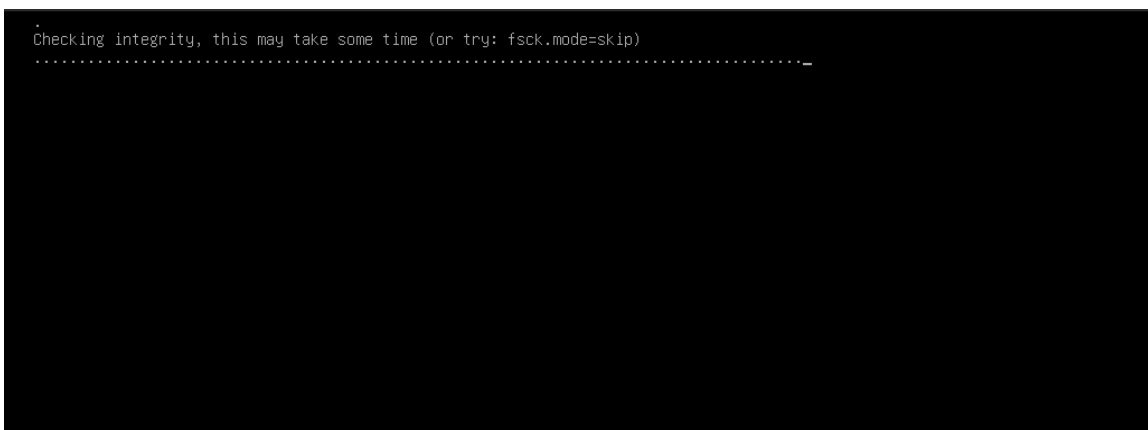


Figura 50: Pantalla de comprobaciones iniciales de Ubuntu Server 20.04.

12. Escogemos el idioma de nuestra preferencia. Para este caso, escogeremos “Español” utilizando las flechas del teclado y damos enter:

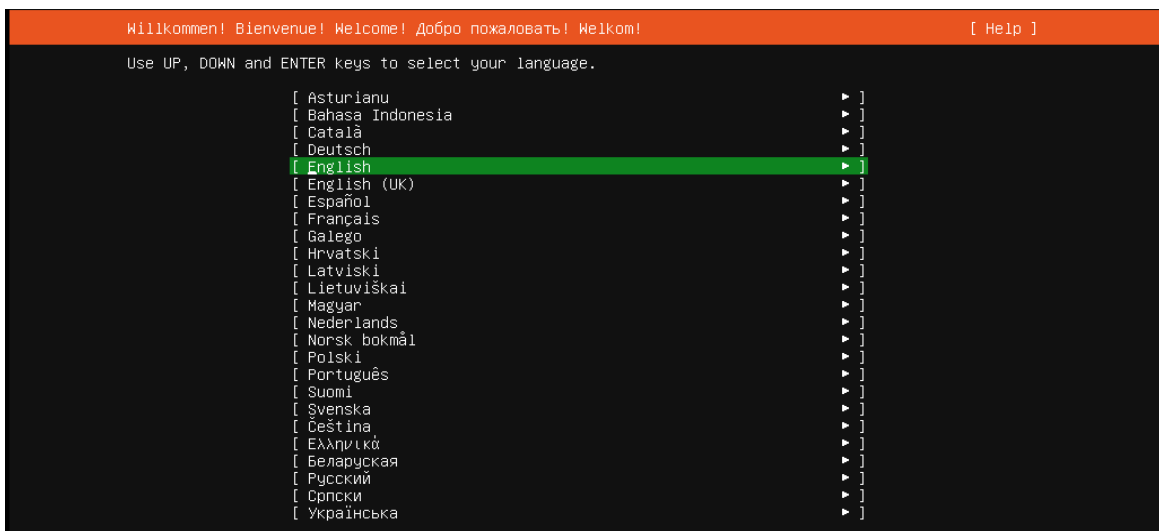
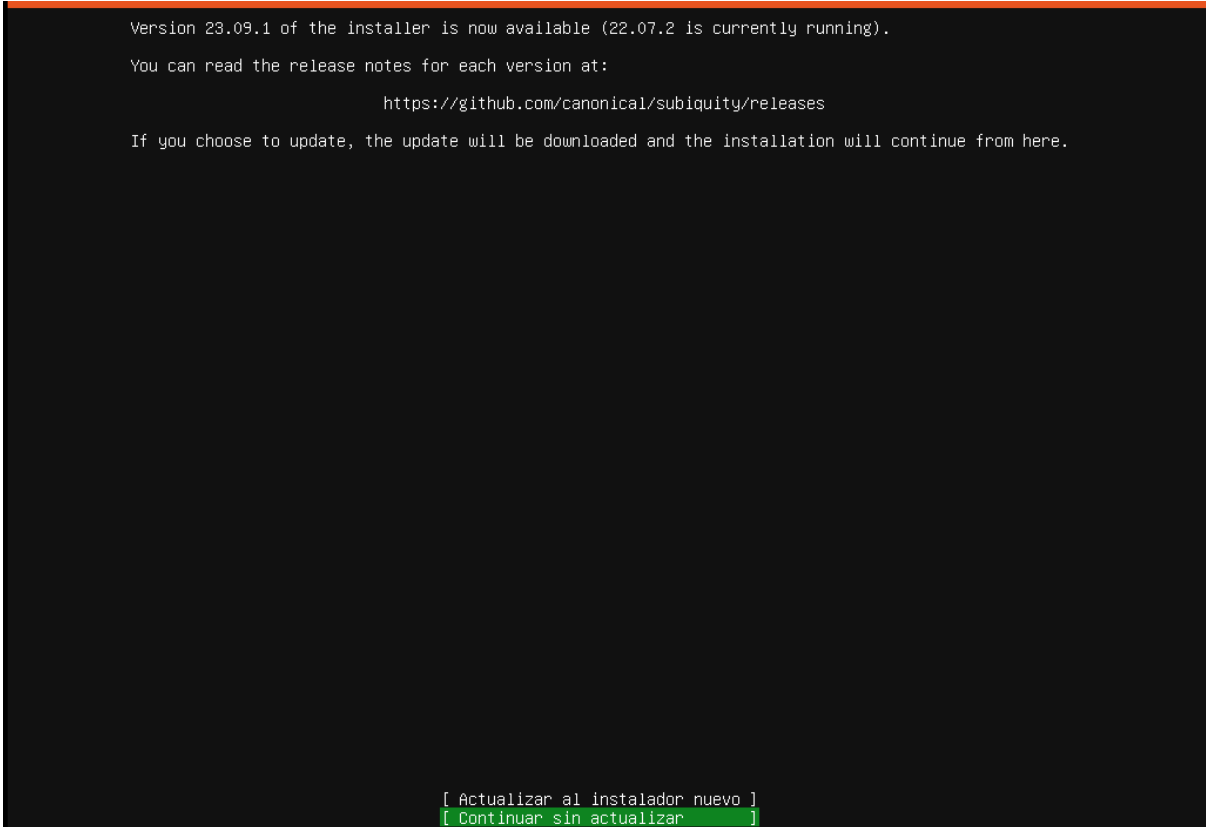


Figura 51: Pantalla de elección de idioma de Ubuntu Server 20.04.

13. Si se nos presenta una pantalla para actualizar el instalador, dar enter en la opción “Continuar sin actualizar”:



```
Version 23.09.1 of the installer is now available (22.07.2 is currently running).
You can read the release notes for each version at:
    https://github.com/canonical/subiquity/releases
If you choose to update, the update will be downloaded and the installation will continue from here.

[ Actualizar al instalador nuevo ]
[ Continuar sin actualizar ]
```

Figura 52: Pantalla de elección de instalador de Ubuntu Server 20.04.

14. En la pantalla de selección referente a la disposición del teclado, damos enter en Disposición y buscamos la opción “Spanish (Latin American)”. Una vez realizado, dar enter en “Hecho”:

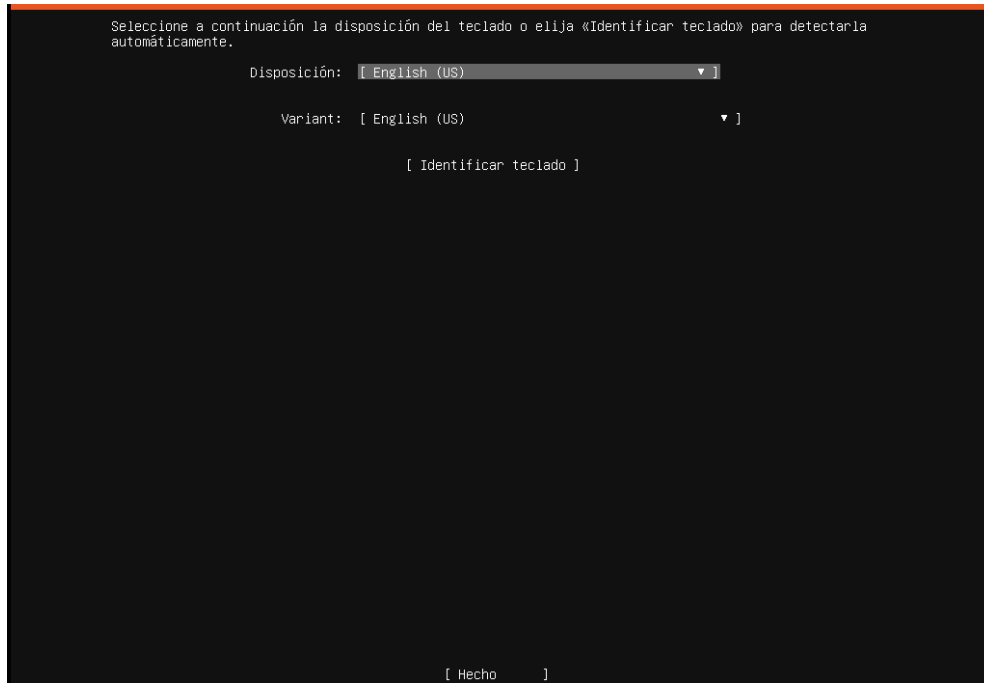


Figura 53: Pantalla de elección de idioma de teclado de Ubuntu Server 20.04.

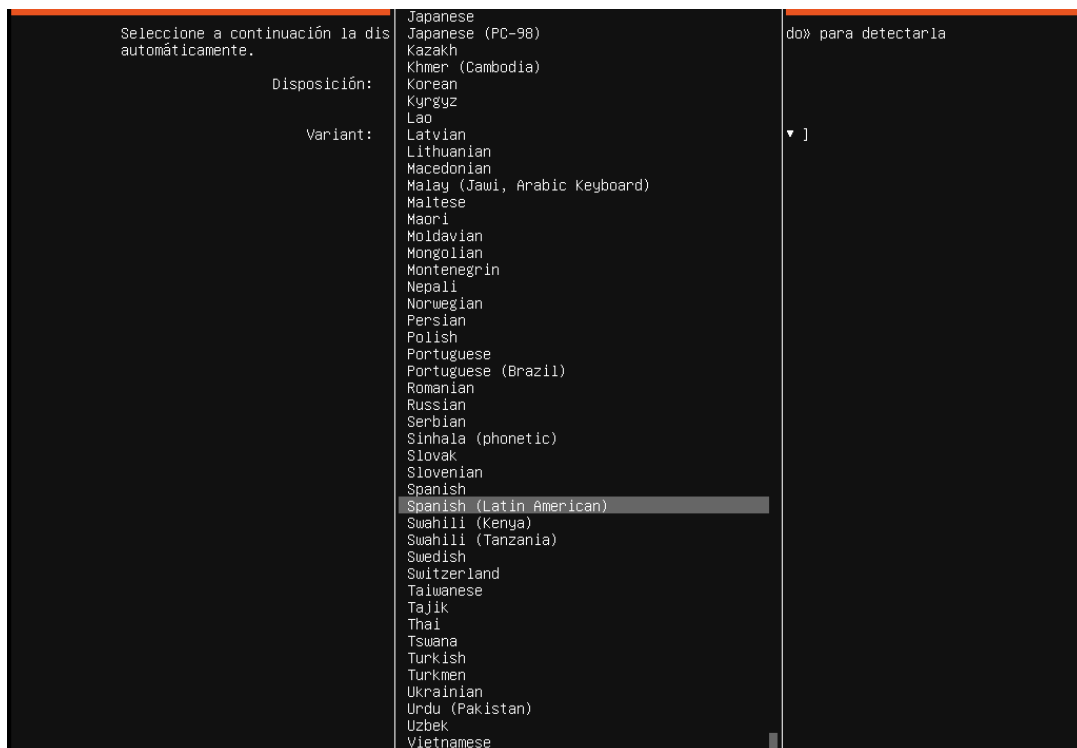


Figura 54: Pantalla de opciones de idioma de teclado de Ubuntu Server 20.04.

15. Damos enter en “Hecho” para la parte de Conexiones de red:

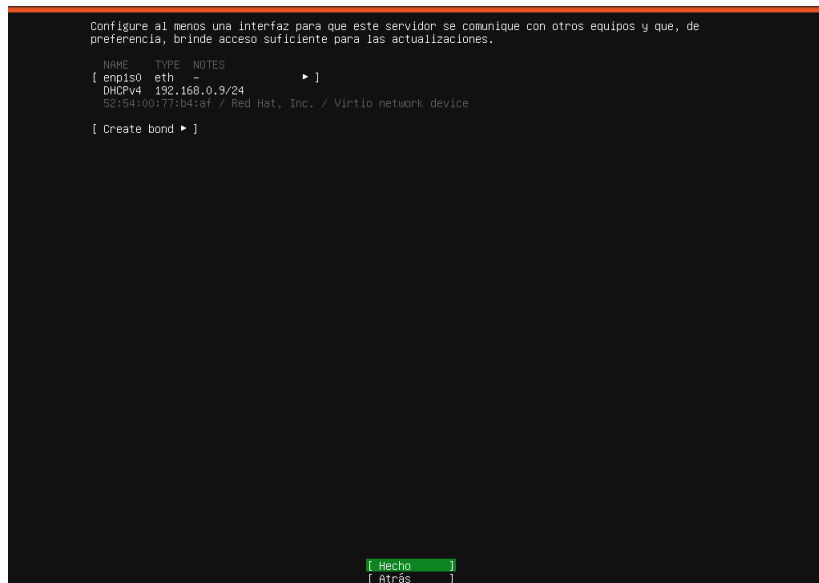


Figura 55: Pantalla de interfaces de Ubuntu Server 20.04.

16. Para la configuración de proxy, lo dejaremos vacío y damos enter en “Hecho”:

17. Damos enter en “Hecho” para la configuración del mirror:

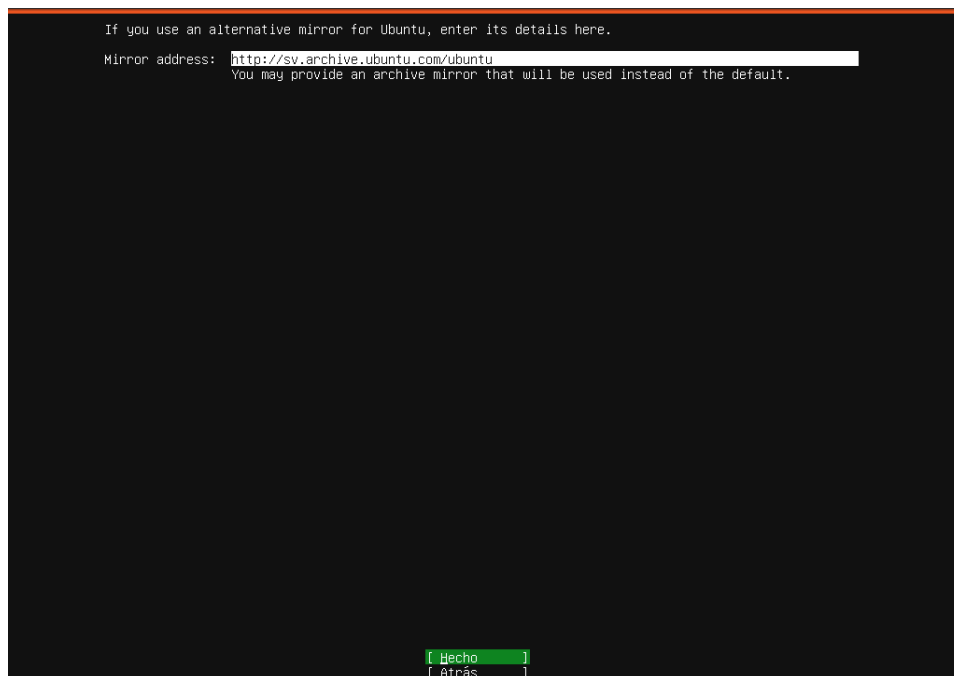


Figura 56: Pantalla de elección del mirror de Ubuntu Server 20.04.

18. Para la configuración de almacenamiento, dejar marcada la opción de utilizar la totalidad del disco y la de configurarlo como LVM. Damos enter en “Hecho”:

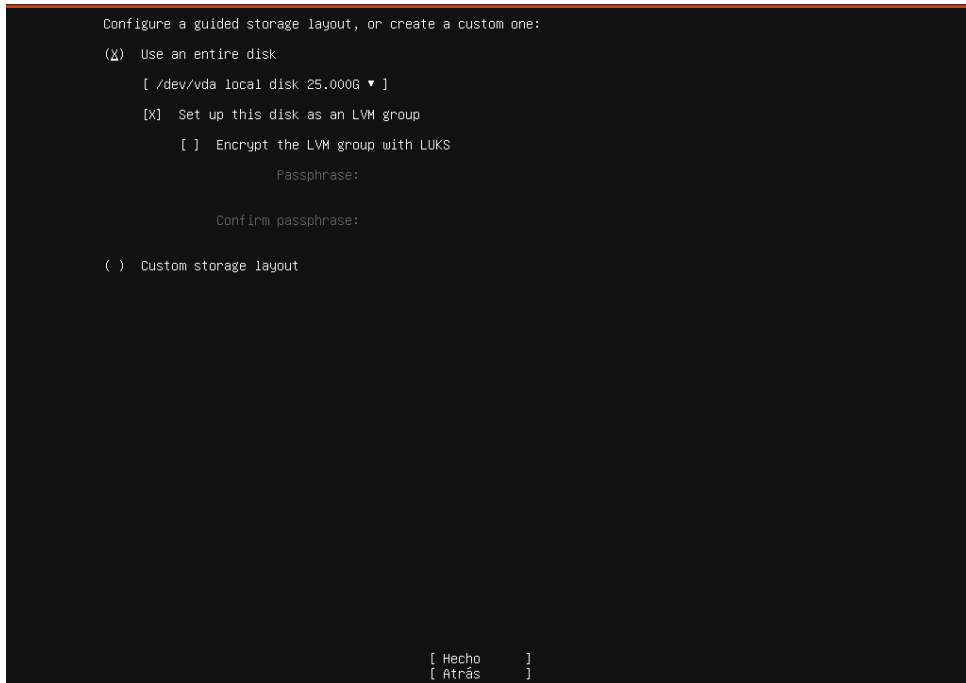


Figura 57: Pantalla de opciones de almacenamiento de Ubuntu Server 20.04.

19. En la siguiente pantalla, nos ubicamos en “ubuntu-lv”, damos enter, nos movemos a la opción “Edit” y damos enter:

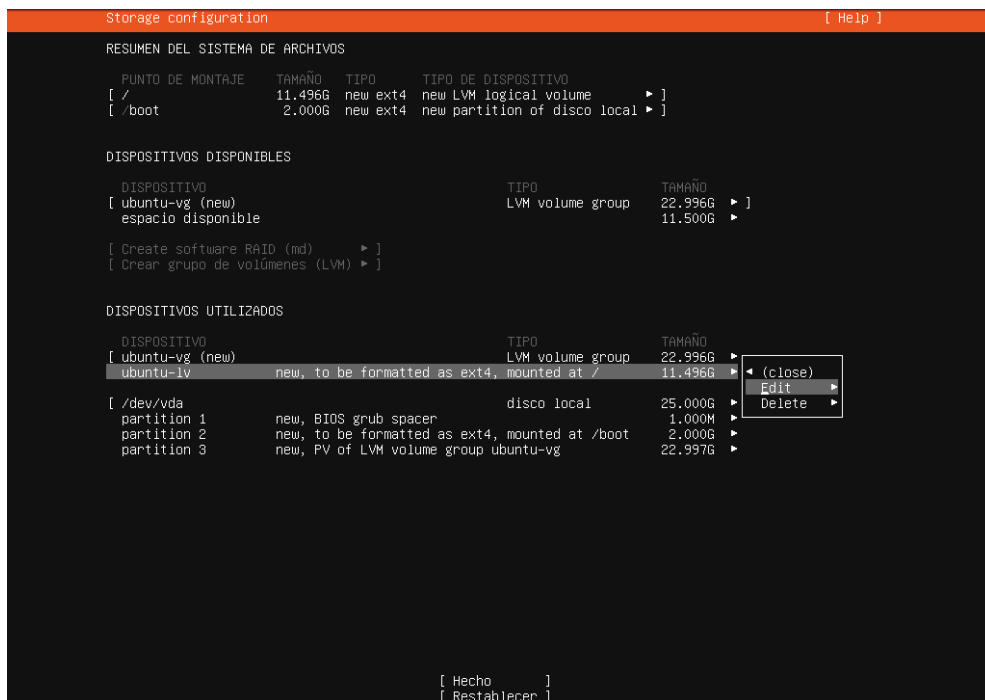


Figura 58: Pantalla de configuración de almacenamiento de Ubuntu Server 20.04.

20. Modificamos el parámetro “Size” para poder aprovechar la máxima capacidad del disco. Nos ubicamos en la opción “Guardar” y damos enter:

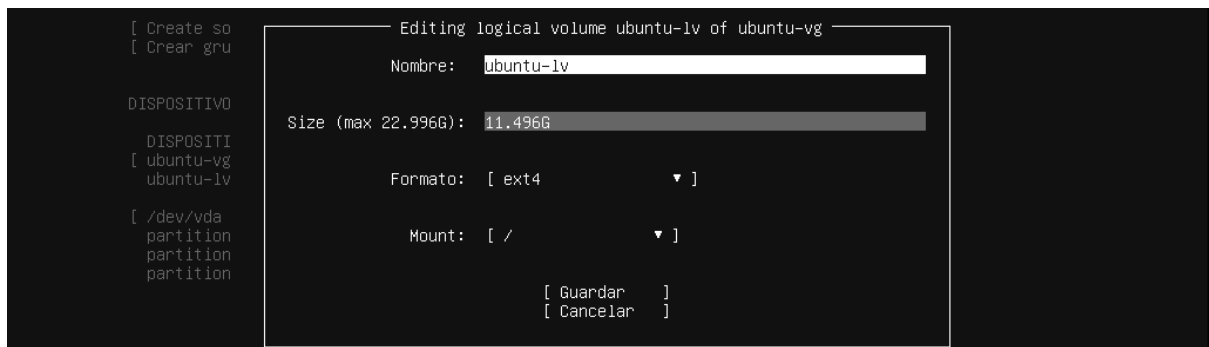


Figura 59: Pantalla de edición de almacenamiento de Ubuntu Server 20.04.

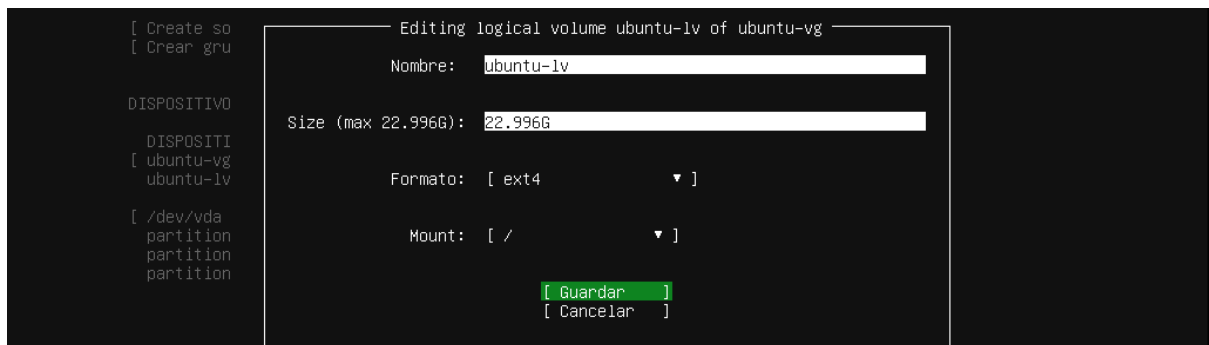


Figura 60: Pantalla de edición de almacenamiento una vez cambiada.

21. Damos enter en la opción “Hecho”:

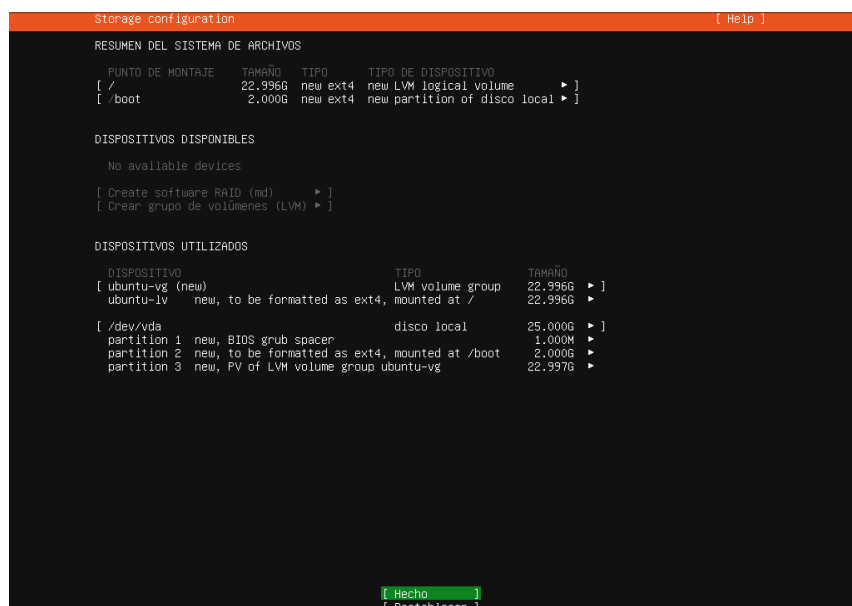


Figura 61: Pantalla de configuración de almacenamiento una vez cambiada.

22. Seleccionamos la opción “Continuar” y damos enter:

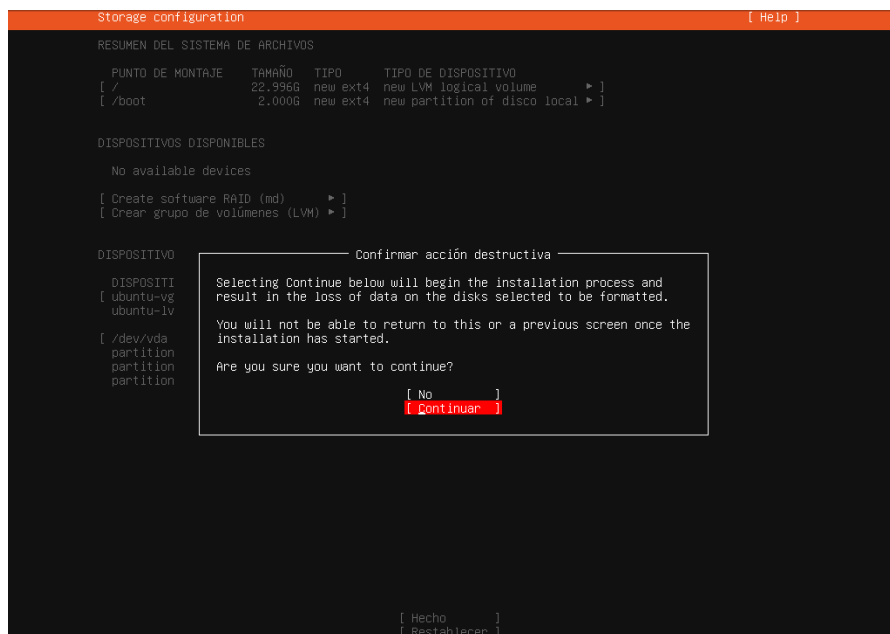


Figura 62: Pantalla de confirmación de opciones escogidas de almacenamiento.

23. En la configuración del usuario, lo llenamos a nuestra conveniencia. Anotar el nombre de usuario y contraseña. Damos enter en la opción “Hecho”:

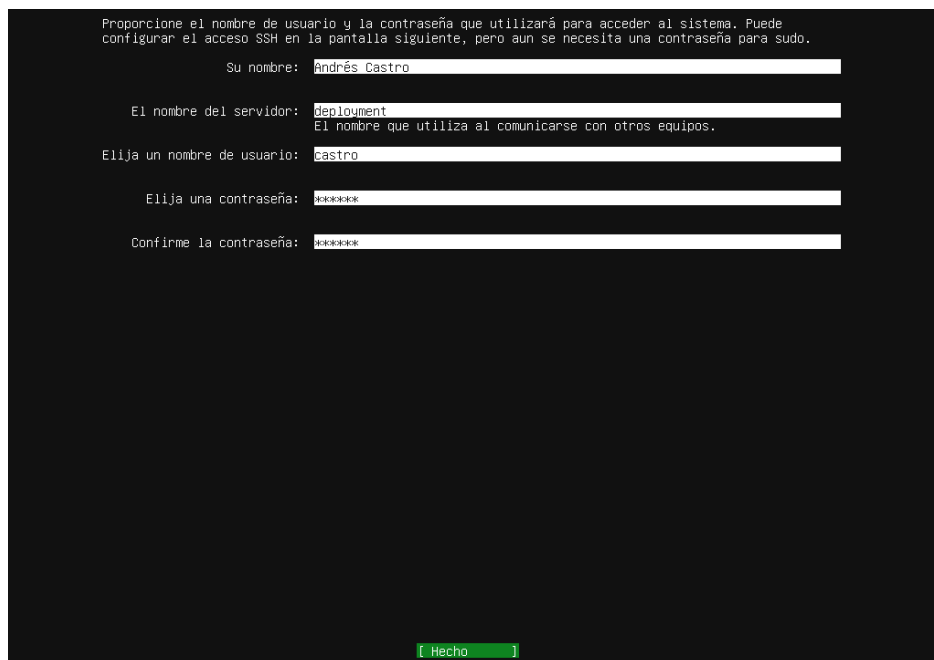


Figura 63: Pantalla de creación de usuario de Ubuntu Server 20.04.

24. En la pantalla de configuración SSH, marcamos la opción para instalar OpenSSH y damos enter en “Hecho”:

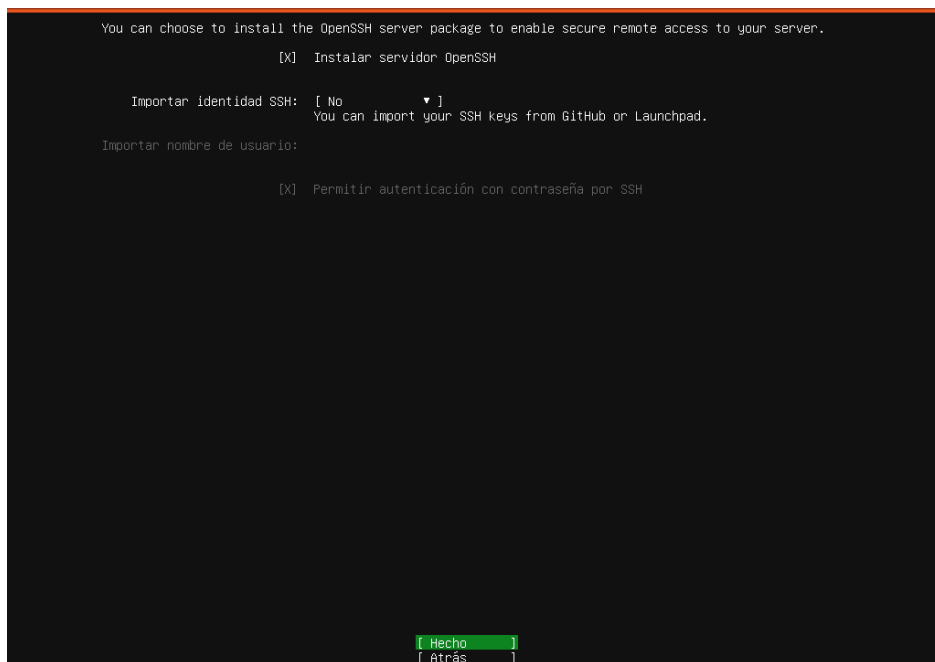


Figura 64: Pantalla de configuración de OpenSSH.

25. En la opción de paquetes adicionales, no seleccionamos ninguna opción y damos enter en “Hecho”:

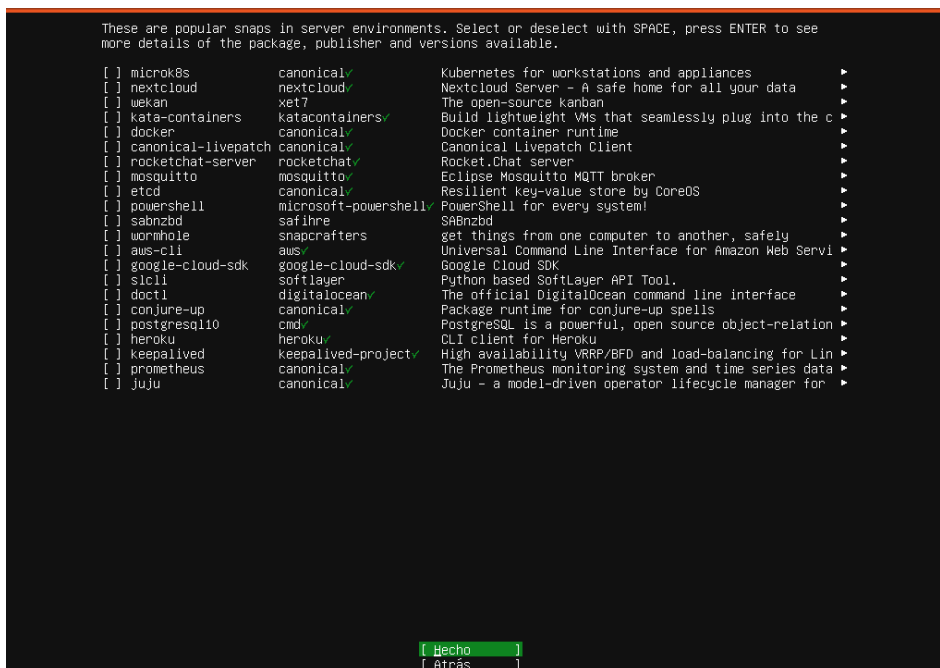


Figura 65: Listado de paquetes adicionales para instalar en Ubuntu Server 20.04.

26. Esperamos que se realice la instalación:

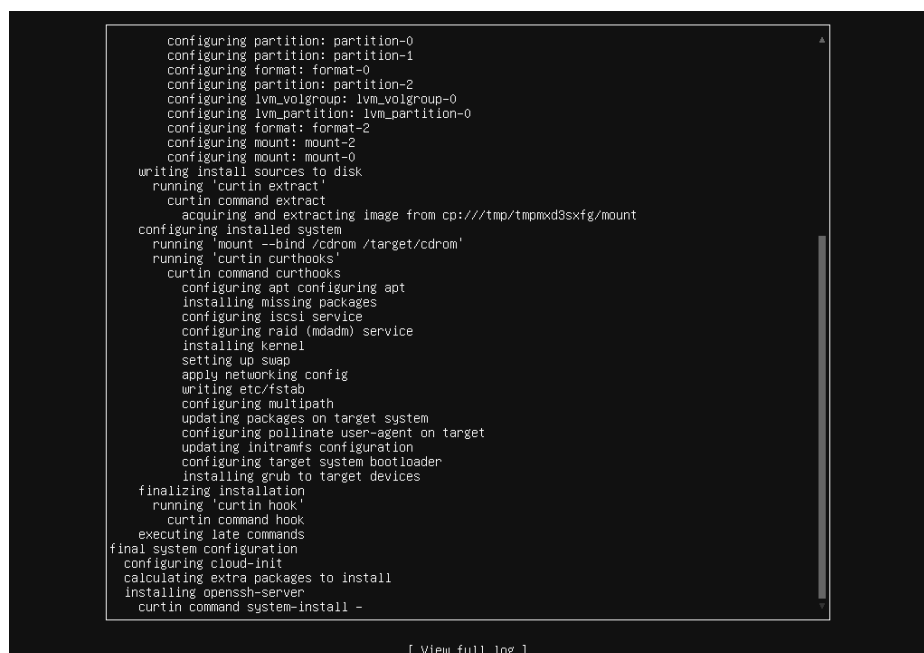
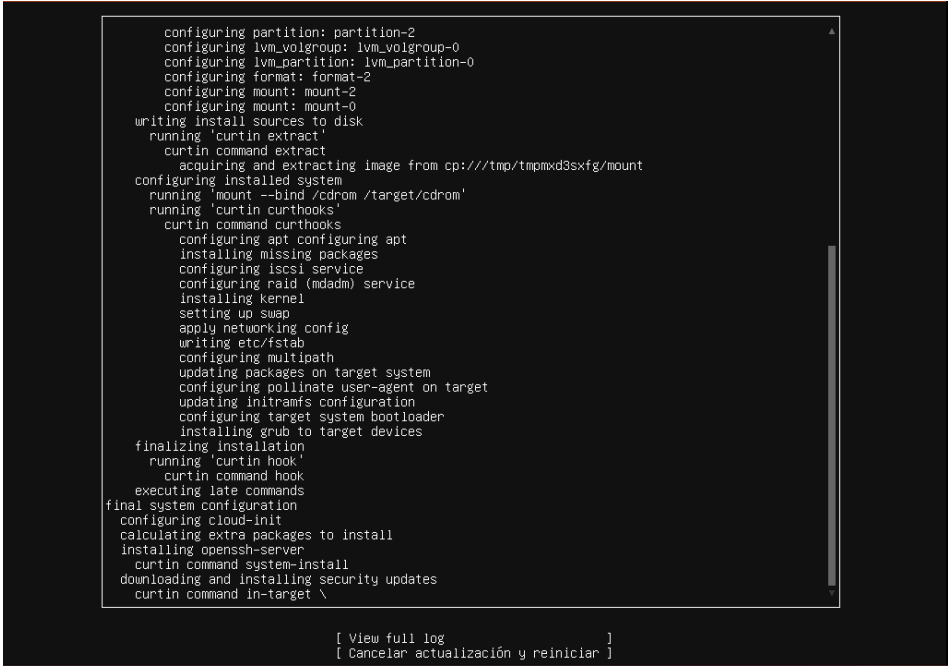


Figura 66: Pantalla de acciones realizadas durante la instalación de Ubuntu Server 20.04.

27. Dejamos que la actualización se lleve a cabo, es decir, **no** dar enter en la opción

“Cancelar actualización y reiniciar”:

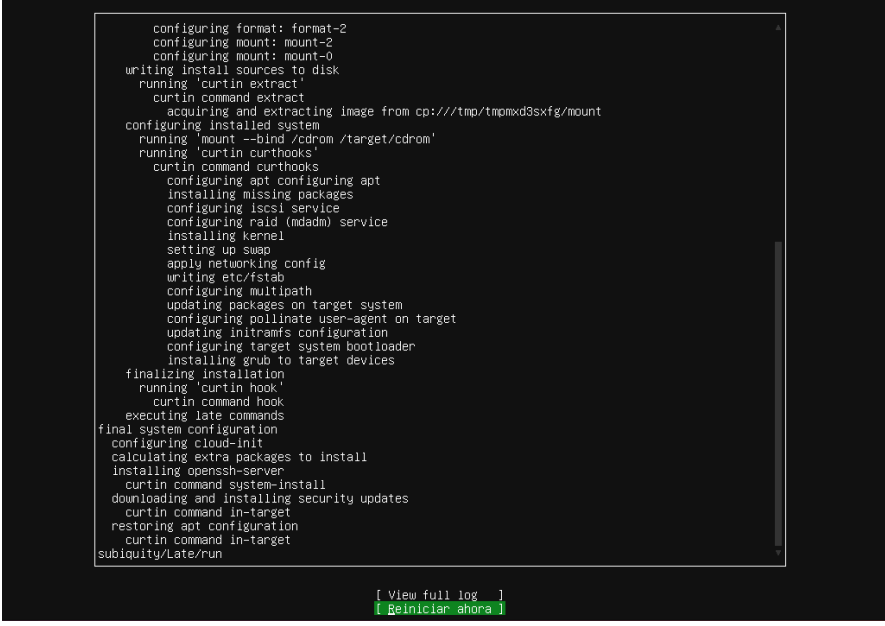


```
configuring partition: partition-2
configuring lvm_voigroup: lvm_voigroup-0
configuring lvm_partition: lvm_partition-0
configuring format: format-2
configuring mount: mount-2
configuring mount: mount-0
writing install sources to disk
running 'curtin extract'
  curtin command extract
  acquiring and extracting image from cp:///tmp/tmpmxd3sxfg/mount
configuring installed system
running 'mount --bind /cdrom /target/cdrom'
running 'curtin curthooks'
  curtin command curthooks
  configuring apt
  installing missing packages
  configuring iscsi service
  configuring raid (mdadm) service
  installing kernel
  setting up swap
  apply networking config
  writing etc/fstab
  configuring multipath
  updating packages on target system
  configuring pollinate user-agent on target
  updating inittarafs configuration
  configuring target system bootloader
  installing grub to target devices
finalizing installation
running 'curtin hook'
  curtin command hook
executing late commands
final system configuration
configuring cloud-init
calculating extra packages to install
installing openssh-server
  curtin command system-install
downloading and installing security updates
  curtin command in-target \

[ View full log ]
[ Cancelar actualización y reiniciar ]
```

Figura 67: Menú de opciones luego de instalación básica de Ubuntu Server 20.04.

28. Una vez terminada la instalación y actualización, damos enter en la opción “Reiniciar ahora”:



```
configuring format: format-2
configuring mount: mount-2
configuring mount: mount-0
writing install sources to disk
running 'curtin extract'
  curtin command extract
  acquiring and extracting image from cp:///tmp/tmpmxd3sxfg/mount
configuring installed system
running 'mount --bind /cdrom /target/cdrom'
running 'curtin curthooks'
  curtin command curthooks
  configuring apt
  installing missing packages
  configuring iscsi service
  configuring raid (mdadm) service
  installing kernel
  setting up swap
  apply networking config
  writing etc/fstab
  configuring multipath
  updating packages on target system
  configuring pollinate user-agent on target
  updating inittarafs configuration
  configuring target system bootloader
  installing grub to target devices
finalizing installation
running 'curtin hook'
  curtin command hook
executing late commands
final system configuration
configuring cloud-init
calculating extra packages to install
installing openssh-server
  curtin command system-install
downloading and installing security updates
  curtin command in-target
restoring apt configuration
  curtin command in-target
subiquity/Late/run

[ View full log ]
[ Reiniciar ahora ]
```

Figura 68: Menú luego de instalación de Ubuntu Server 20.04.

29. Al aparecer el siguiente mensaje de error, dar clic en el botón de Información en la esquina superior derecha:

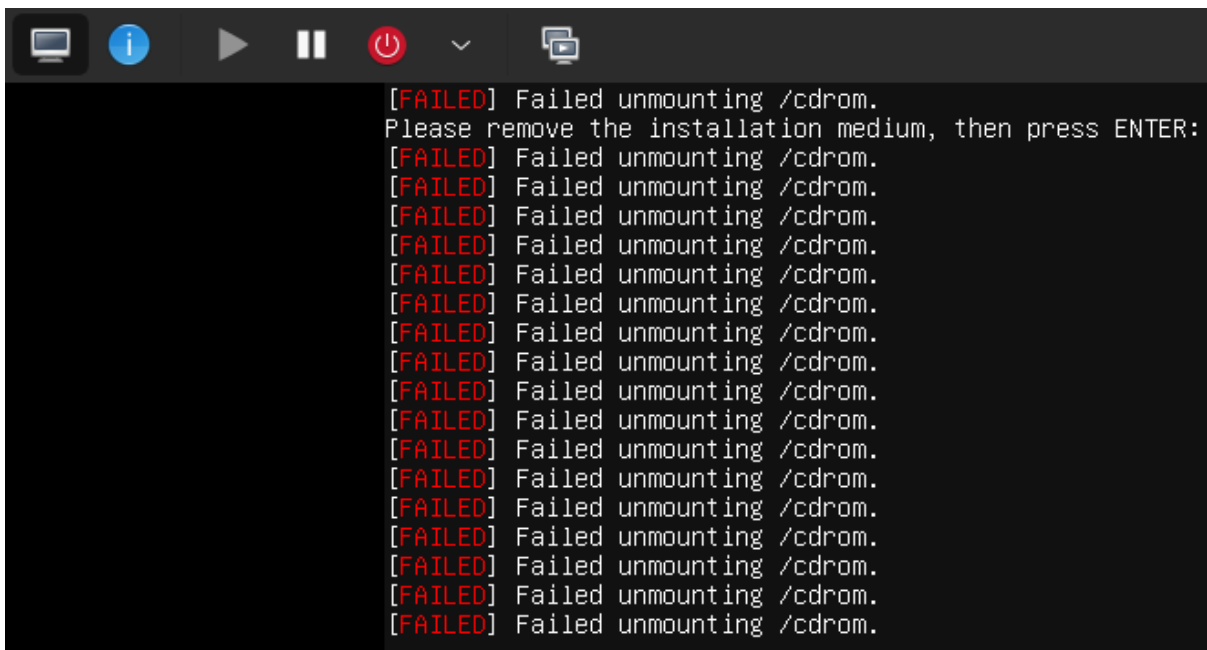


Figura 69: Pantalla de consola luego de instalación de Ubuntu Server 20.04.

30. Seleccionamos la opción de SATA CDROM y quitamos la imagen ISO en el botón que se indica con la flecha:

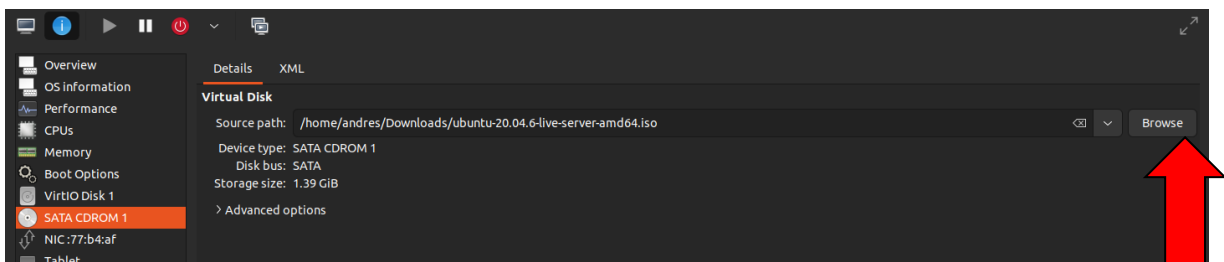


Figura 70: Opciones del dispositivo virtual óptico de la instancia.

31. Una vez realizado el paso anterior, damos clic en el botón “Apply” en la esquina inferior derecha:

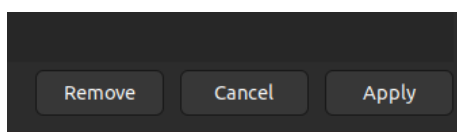


Figura 71: Botones de acciones para la instancia.

32. Damos clic en el botón de la pantalla para volver a consola:



Figura 72: Botones de pantallas de la instancia.

33. Damos enter para reiniciar la máquina virtual.

34. Esperamos a que reinicie la máquina virtual. Una vez esté arriba de nuevo, iniciamos sesión con el usuario creado previamente:

```
deployment login: [ 19.158181] cloud-init[1471]: Generating locales (this might take a while)...
[ 20.780552] cloud-init[1471]: es_ES.UTF-8... done
[ 20.784624] cloud-init[1471]: Generation complete.
[ 21.191269] cloud-init[1471]: Cloud-init v. 23.1.2-0ubuntu0~20.04.2 running 'modules:config' at Wed, 11 Oct 2023 01:17:38 +00:00. Up 18.99 seconds.
ci-info: no authorized SSH keys fingerprints found for user castro.
<14>Oct 11 01:17:41 cloud-init: #####
<14>Oct 11 01:17:41 cloud-init: -----BEGIN SSH HOST KEY FINGERPRINTS-----
<14>Oct 11 01:17:41 cloud-init: 1024 SHA256:YtoEdepo+RPPdKX4b9hJVMgdSIhDoGSMNrfj/bSzmQ root@deployment (DSA)
<14>Oct 11 01:17:41 cloud-init: 256 SHA256:3Pkyutj7A111eJbYnJgMjUuUR4GMk2J8Shp1bN7B8 root@deployment (ECDSA)
<14>Oct 11 01:17:41 cloud-init: 256 SHA256:gwwbTYBuLgHmjdJtpM2qe1IghXj4+1F+1yCB609TQ9M root@deployment (ED25519)
<14>Oct 11 01:17:41 cloud-init: 3072 SHA256:fob9bFsUarDX0/UXhAD0mRCJZD384JXGv6rRfp1gb6A root@deployment (RSA)
<14>Oct 11 01:17:41 cloud-init: -----END SSH HOST KEY FINGERPRINTS-----
<14>Oct 11 01:17:41 cloud-init: #####
-----BEGIN SSH HOST KEY KEYS-----
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoVTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBHLjMDPg3TKGMBRw3NxoUSAFtvVR/+HBnY9Wm3VCUsZK1V21LW/irwJP
hvqCsEAYmU3C02SngC8aIySwIK2pmtc= root@deployment
ssh-ed25519 AAAAC3NzaC1lZDI1IbnE5AAAAIFeviziGSLj9R18GgzARY+LWVo8tsXmUQrwRTMkp1+1S root@deployment
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDQrX692Y3CYGn9iMX6XyznmnczUhu+QMz3323scPVHfuN0bpNTQQCeQINGJzLHPm1bCGv0vV7Ue+1+Kmmkc2ci7jw0
+qb/HU+8IPbHh72Q2HwUs8fWNYEoyanJKZMk04JgP+Q3e6z4pDVb6RJbFrXpRIFIJ7b0FE20cImEjL61ipPg5bLEspzbu27k3T40vweokhrs84rgYBp45pofZzvwitPyT
QA7h0vj1Y0797rhx8JVMUMNEifTqSao1S8Knnj6h0dKSihjv0MbyS0j4efu1wSv4o4UJ4w1VES4KMIOUkDjzjinpJR36XP44mi3J16jBwX1fecJWRes+6pgV+24hkURr
rGPt30kdnZkHEvk+LHh2DKd1Pj5EvcF3CooxkV9K41xf2pCjvLZX54FP1vjoG2XsDaR0XswECZka1DQKz6nD1T2nx6m5pSalmYcVn0B/qfp3Nn+F7ktZ3A1XRm6W44h
0x6Paau2Hcx4j884ktgxEGDIu44Rf9xx/yU1lgnE= root@deployment
-----END SSH HOST KEY KEYS-----
[ 21.951645] cloud-init[1513]: Cloud-init v. 23.1.2-0ubuntu0~20.04.2 running 'modules:final' at Wed, 11 Oct 2023 01:17:41 +0000. Up 21.72 seconds.
[ 21.956229] cloud-init[1513]: Cloud-init v. 23.1.2-0ubuntu0~20.04.2 finished at Wed, 11 Oct 2023 01:17:41 +0000. Datasource D
ataSourceNone. Up 21.94 seconds
[ 21.956418] cloud-init[1513]: 2023-10-11 01:17:41,634 - cc_final_message.py[WARNING]: Used fallback datasource

deployment login: castro
Password:
```

Figura 73: Login de Ubuntu Server 20.04.

35. Se nos mostrará la siguiente información en consola:

```
deployment login: castro
Password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-164-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of mié 11 oct 2023 01:19:13 UTC

System load:  0.2          Processes:            137
Usage of /:   27.9% of 22.47GB   Users logged in:    0
Memory usage: 5%          IPv4 address for enp1s0: 192.168.0.9
Swap usage:  0%           IPv6 address for enp1s0: ::5054:ff:fe77:b4af

Expanded Security Maintenance for Applications is not enabled.

50 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

castro@deployment:~$
```

Figura 74: Consola de Ubuntu Server 20.04.

En la sección de configuración e instalación se realizará la configuración de las interfaces.

IV. Configuración e Instalación

A. pfSense

1. Agregando VLANs.

1. Nos dirigimos a la opción Assignments del menú interfaces:

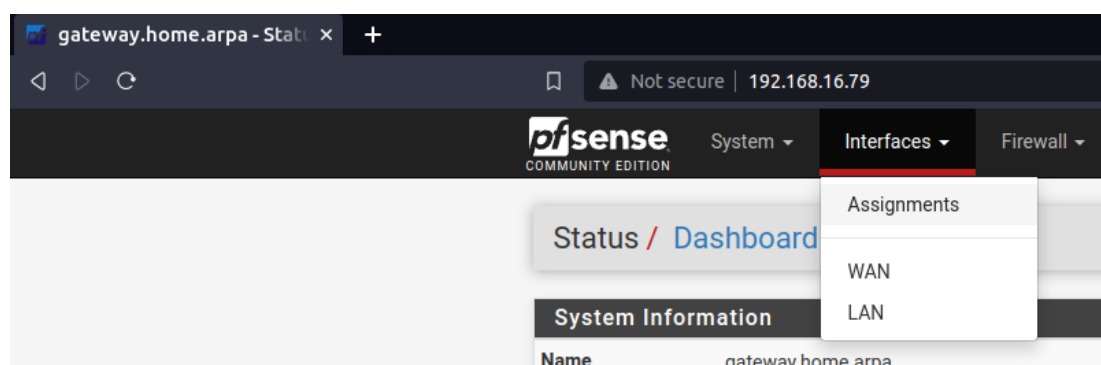


Figura 75: Menú de interfaces en pfSense.

2. Seleccionamos la pestaña VLANs:

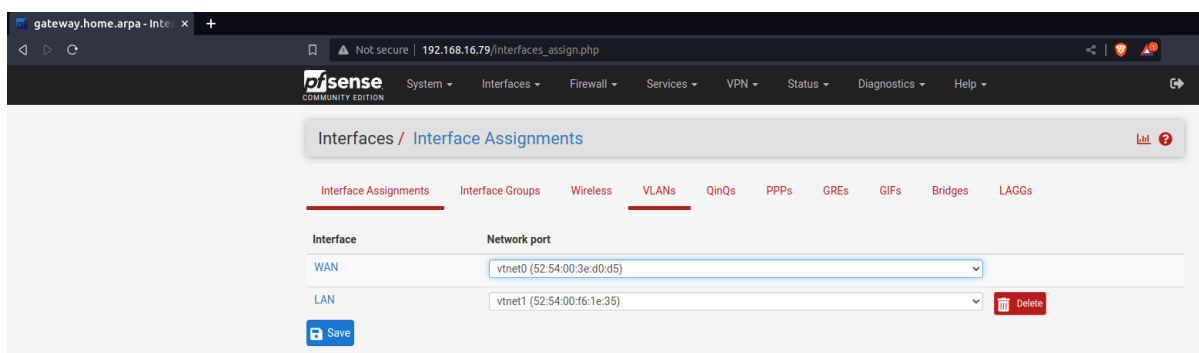


Figura 76: Listado de interfaces en pfSense.

3. Hacemos clic en Add:

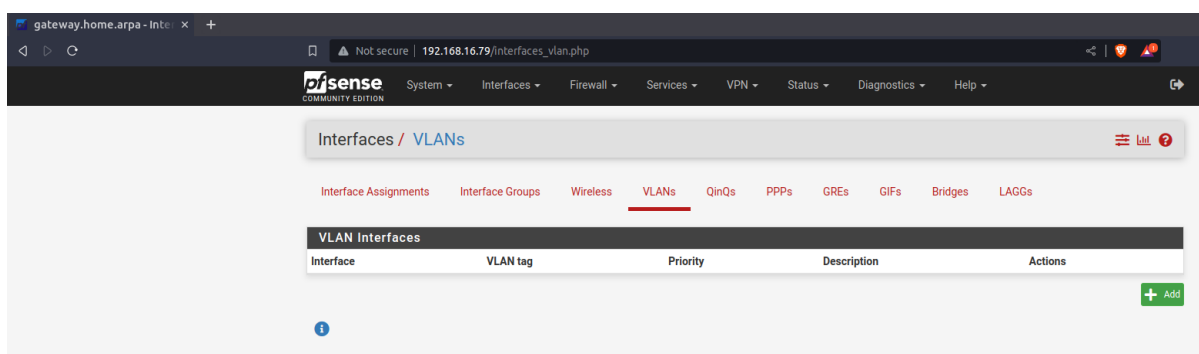


Figura 77: Listado de VLANs en pfSense.

4. Para agregar la VLAN de management cambiaremos la interfaz padre a la vtnet1, asignamos la etiqueta correspondiente e ingresamos una breve descripción. Finalmente, damos clic en Save:

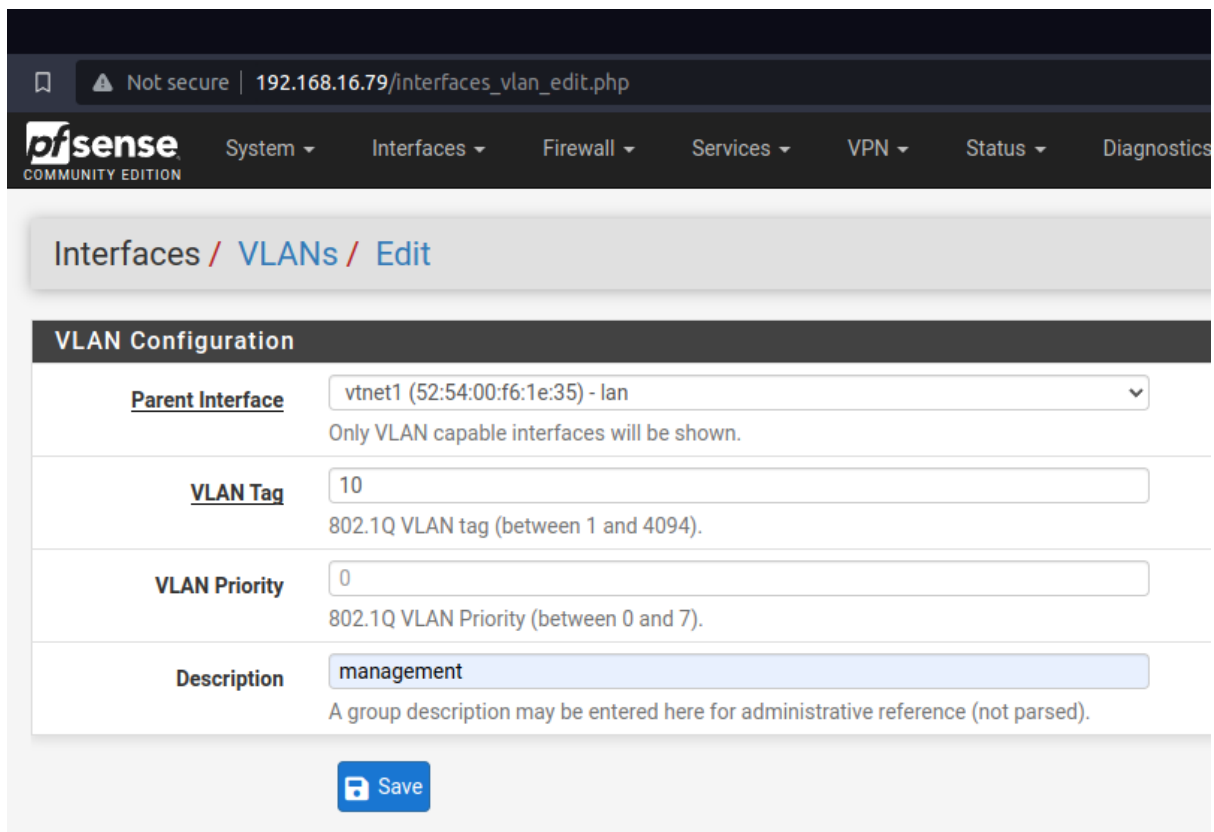


Figura 78: Pantalla de edición de VLAN en pfSense.

5. Repetir los pasos anteriores para crear las demás VLANs.

2. Agregando Interfaces.

1. En la sección de asignación de interfaces daremos clic en agregar:

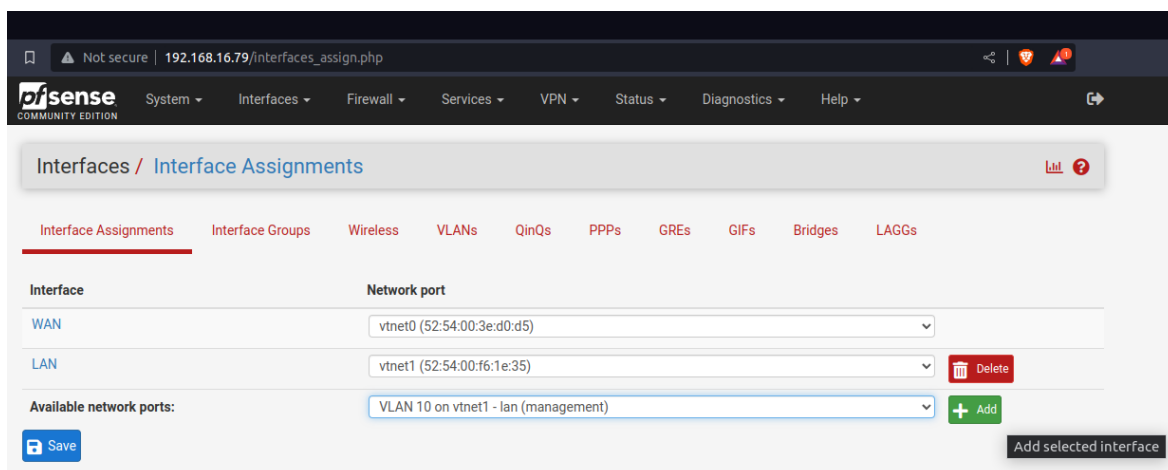


Figura 79: Listado de interfaces en pfSense.

2. Nos aparecerá la interfaz OPT1. Daremos clic sobre el nombre:

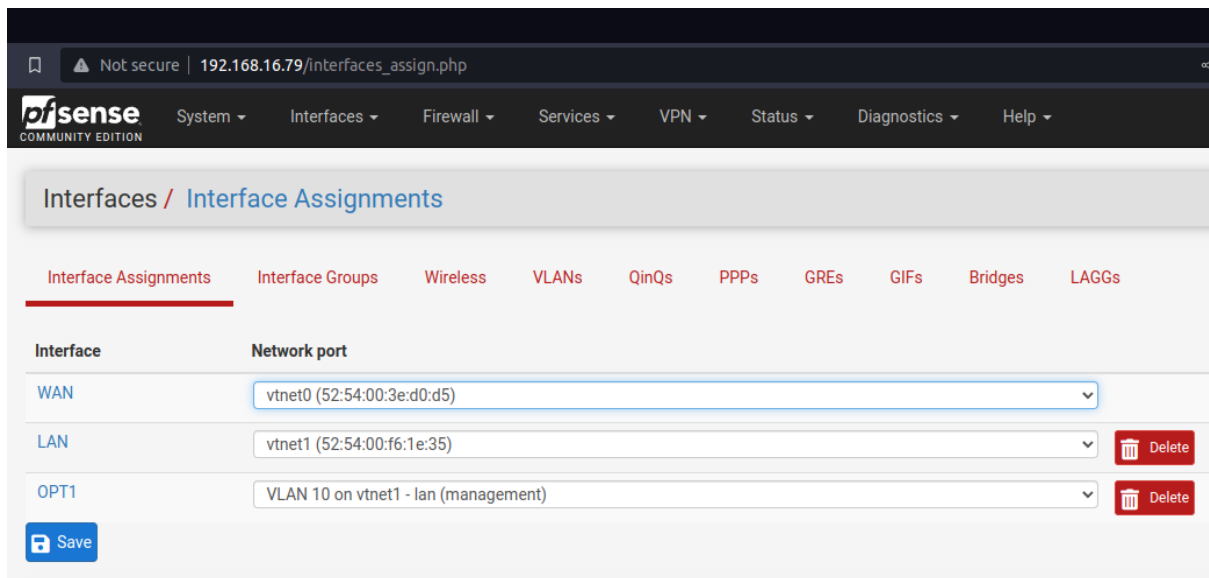


Figura 80: Listado de interfaces en pfSense una vez se agrega una nueva.

3. Cambiaremos las siguientes configuraciones:

- Enable interface
- Description
- IPv4 configuration type
- IPv4 address

Recordar que estamos configurando para la vlan management, por tanto para las demás vlans se debe cambiar el valor IPv4 “address”.

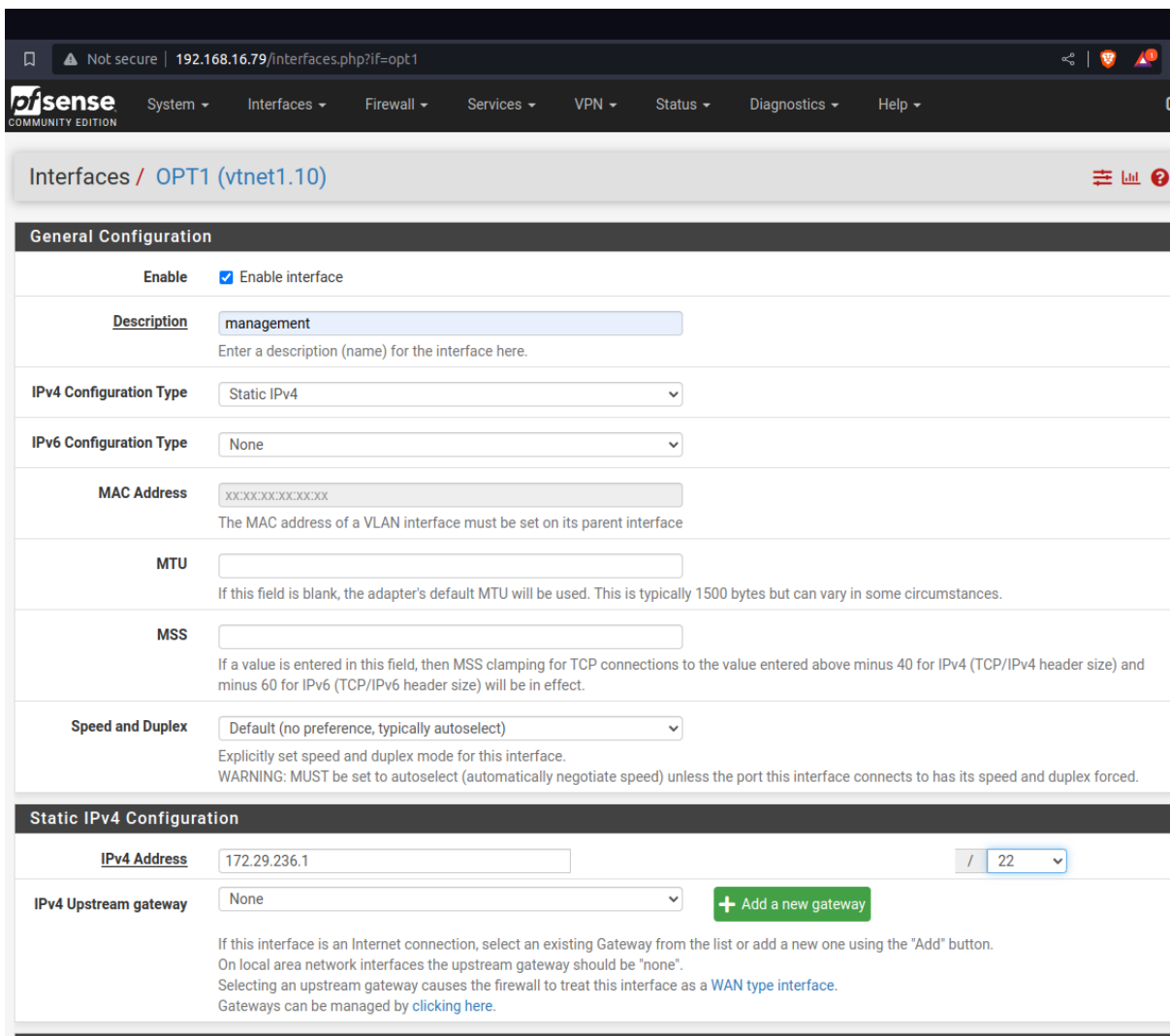


Figura 81: Modificación de interfaces en pfSense.

4. Haremos clic en Save al final de la página.
 5. Damos clic en Apply Changes para que se recargue la configuración.
 6. Repetir los pasos anteriores para las demás interfaces.
- 3. Agregando Reglas.**
1. Seleccionamos el menú Firewall opción Rules:

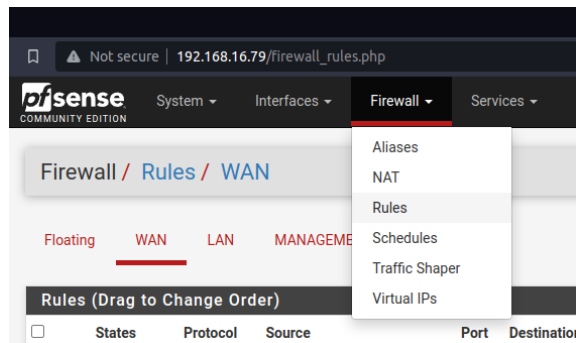


Figura 82: Menú de Firewall en pfSense.

2. En la pestaña MANAGEMENT damos clic en Add ↑ :

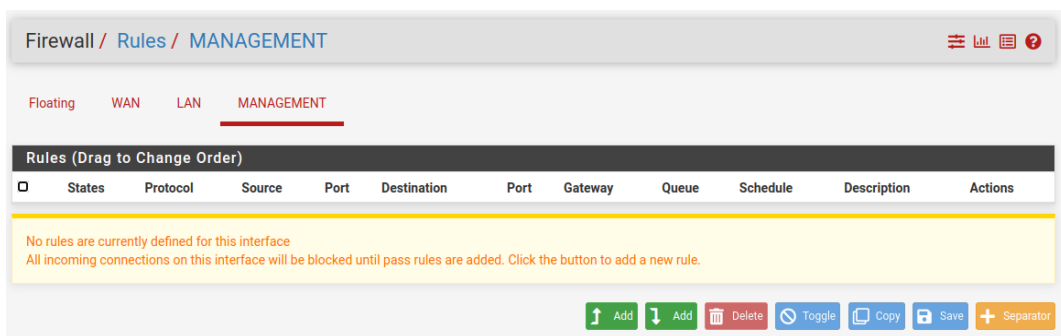


Figura 83: Listado de reglas por interfaz en pfSense.

3. Agregaremos la configuración para ICMP. Modificamos el protocolo, damos clic en guardar y aplicamos los cambios. Considerar que el valor del parámetro action es *pass*:

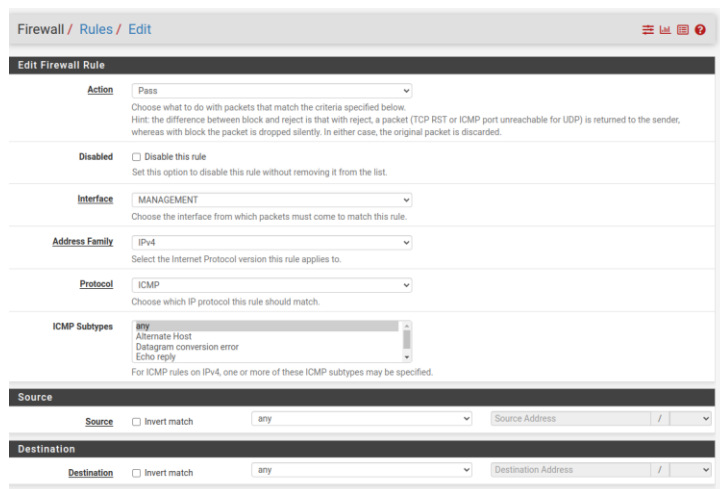


Figura 84: Edición de regla en pfSense.

4. Repetimos los pasos de acuerdo a nuestras necesidades y para las otras VLANs.

B. Ceph

1. Preparando nodos.

1. En cada nodo del clúster de ceph modificar el archivo /etc/hosts:

```
172.29.244.40 ceph-admin
172.29.244.41 ceph-mon
172.29.244.42 ceph-osd1
172.29.244.43 ceph-osd2
172.29.244.44 ceph-osd3
```

2. Ejecutar los siguientes comandos en todos los nodos:

```
apt install chrony openssh-server curl -y
systemctl enable --now chrony
systemctl enable --now sshd
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg

# Add the repository to Apt sources:
echo \
  "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu \
  "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin
```

3. En el nodo ceph-admin creamos el siguiente usuario:

```
useradd -m -s /bin/bash cephadmin
```

4. Establecemos la contraseña del usuario creado (la que se desee):

```
passwd cephadmin
```

5. Hacemos sudo a dicho usuario:

```
echo "cephadmin ALL=(ALL:ALL) NOPASSWD:ALL" >>  
/etc/sudoers.d/cephadmin  
chmod 0440 /etc/sudoers.d/cephadmin
```

6. En los nodos mon y osd modificaremos el archivo /etc/ssh/sshd_config y modificamos la línea referente a permitir una sesión con el usuario root:

```
PermitRootLogin yes
```

7. Reiniciamos el servicio:

```
service sshd restart
```

8. Establecer una contraseña para el usuario root y anotarla:

```
passwd
```

2. *Iniciando el clúster.*

1. Descargamos el paquete de ceph-adm y le damos los permisos con los comandos siguientes:

```
sudo su  
wget -q https://github.com/ceph/ceph/raw/quincy/src/cephadm/cephadm -P /usr/bin/  
chmod +x /usr/bin/cephadm
```

2. Cambiamos de usuario:

```
su - cephadmin
```

3. Inicializamos el clúster con el siguiente comando:

```
sudo cephadm bootstrap --mon-ip 172.29.244.41
```

4. Esperamos que termine el proceso y anotamos la contraseña que sea desplegada en consola, la url para visualizar el panel administrativo y el usuario :

```
Verifying podman|docker is present...
Verifying lvm2 is present...
Verifying time synchronization is in place...
Unit systemd-timesyncd.service is enabled and running
Repeating the final host check...
docker (/usr/bin/docker) is present
systemctl is present
lvcreate is present
Unit systemd-timesyncd.service is enabled and running
Host looks OK
```

Ceph Dashboard is now available at:

```
URL: https://172.29.244.41:8443/
User: admin
Password: 8964vjghsi
```

```
Enabling client.admin keyring and conf on hosts with "admin" label
Enabling autotune for osd_memory_target
You can access the Ceph CLI with:
```

```
sudo /usr/bin/cephadm shell --fsid f959b65e-91c2-11ec-9776-abbffb8a52a1
-c /etc/ceph/ceph.conf -k /etc/ceph/ceph.client.admin.keyring
```

Please consider enabling telemetry to help improve Ceph:

```
ceph telemetry on
```

For more information see:

```
https://docs.ceph.com/docs/quincy/mgr/telemetry/
```

```
Bootstrap complete.
```

5. Instalamos el paquete ceph-common en el nodo ceph-admin:

```
sudo apt install ceph-common
```

3. *Registrando nodos al clúster.*

1. Copiamos la llave ssh desde el nodo ceph-admin a los demás nodos:

```
sudo ssh-copy-id -f -i /etc/ceph/ceph.pub root@ceph-mon  
sudo ssh-copy-id -f -i /etc/ceph/ceph.pub root@ceph-osd1  
sudo ssh-copy-id -f -i /etc/ceph/ceph.pub root@ceph-osd2  
sudo ssh-copy-id -f -i /etc/ceph/ceph.pub root@ceph-osd3
```

2. Añadimos los nodos al clúster:

```
sudo ceph orch host add ceph-mon  
sudo ceph orch host add ceph-osd1  
sudo ceph orch host add ceph-osd2  
sudo ceph orch host add ceph-osd3
```

3. Añadimos una etiqueta a los nodos agregados:

```
sudo ceph orch host label add ceph-mon mon  
sudo ceph orch host label add ceph-osd1 osd  
sudo ceph orch host label add ceph-osd2 osd  
sudo ceph orch host label add ceph-osd3 osd
```

4. *Inflando el clúster.*

1. En cada nodo OSD realizamos los siguientes comandos:

```
vgcreate vg01 /dev/vdb  
lvcreate -L 99G -n lv01 vg01
```

2. Desde el nodo ceph-admin agregamos los volúmenes:

```
sudo ceph orch daemon add osd ceph-osd1:vg01/lv01  
sudo ceph orch daemon add osd ceph-osd2:vg01/lv01  
sudo ceph orch daemon add osd ceph-osd3:vg01/lv01
```

3. Revisamos los datos del clúster con el siguiente comando:

```
sudo ceph -s
```

Observamos la salida en consola, donde podremos ver el uso del espacio.

5. *Creando pools.*

Ejecutar los siguientes comandos para crear los pools:

```
ceph osd pool create volumes  
ceph osd pool create images  
ceph osd pool create backups  
ceph osd pool create vms  
  
rbd pool init volumes  
rbd pool init images  
rbd pool init backups  
rbd pool init vms
```

Configuramos el número de réplicas a 2:

```
ceph osd pool set volumes size 2  
ceph osd pool set images size 2  
ceph osd pool set backups size 2  
ceph osd pool set vms size 2
```


6. *Creando usuarios y keyrings.*

Realizamos los siguientes comandos para generar los keyrings.

```
ceph auth get-or-create client.glance mon 'profile rbd' osd 'profile rbd pool=images' mgr 'profile rbd pool=images'
ceph auth get-or-create client.cinder mon 'profile rbd' osd 'profile rbd pool=volumes, profile rbd pool=vms, profile rbd-read-only pool=images' mgr 'profile rbd pool=volumes, profile rbd pool=vms'
ceph auth get-or-create client.cinder-backup mon 'profile rbd' osd 'profile rbd pool=backups' mgr 'profile rbd pool=backups'
```

Anotar los resultados que generan los comandos.

C. **Infraestructura**

1. *Preparando nodo.*

1. Instalar paquete vlan y activar módulo 8021q:

```
apt install vlan
modprobe 8021q
```

2. Modificar el archivo de netplan, considerando el nombre de la interfaz:

```
ip a #para revisar el nombre de las interfaces
nano /etc/netplan/00-installer-config.yaml
```

```
network:
  ethernets:
    enp1s0:
      addresses: [192.168.x.x/24] #adaptar de acuerdo a la LAN de su caso
      nameservers:
        addresses: [8.8.8.8,8.8.4.4]
      routes:
        - to: default
          via: 192.168.0.1
          metric: 100
  vlans:
```

```
#red management
enp1s0.10:
  addresses: [172.29.236.5/22]
  id: 10
  link: enp1s0
# red storage
enp1s0.20:
  addresses: [172.29.244.5/22]
  id: 20
  link: enp1s0
version: 2
```

2. *Chrony.*

1. Instalamos el paquete de chrony y editamos su archivo de configuración:

```
apt install chrony
nano /etc/chrony/chrony.conf
```

2. En chrony.conf modificar/agregar de [acuerdo a la guía oficial](#):

```
server ntp.ues.edu.sv iburst
allow 172.29.236.0/22
```

3. Ejecutamos el siguiente comando para reiniciar el servicio:

```
service chrony restart
```

4. Comprobamos su funcionamiento con el siguiente comando:

```
chronyc sources
```

3. *Repositorio de OpenStack.*

Ejecutar los siguientes comandos:

```
add-apt-repository cloud-archive:wallaby
apt install python3-openstackclient
```

4. *MariaDB.*

1. Ejecutar los siguientes comandos para instalar el servidor de MariaDB:

```
apt install curl python3-pymysql -y
curl -LsS https://r.mariadb.com/downloads/mariadb_repo_setup | sudo bash
apt update
apt install mariadb-server
apt install rsync
```

2. Modificar el archivo */etc/mysql/mariadb.conf.d/50-server.cnf*:

```
bind-address      = 0.0.0.0 #o en su defecto 172.29.236.5
```

5. *RabbitMQ.*

1. Ejecutar los siguientes para instalar RabbitMQ:

```
apt install rabbitmq-server
rabbitmqctl add_user openstack icc115
rabbitmqctl set_permissions openstack ".*" ".*" ".*"
```

6. *Memcached.*

1. Instalar el paquete de Memcached:

```
apt install memcached python3-memcache
```

2. Modificamos el archivo */etc/memcached.conf* :

```
-l 172.29.236.5
```

3. Ejecutar el siguiente comando para reiniciar el servicio:

```
service memcached restart
```

7. *Keystone.*

1. Entramos a la consola de MariaDB:

```
mariadb
```

2. Dentro de MariaDB ejecutamos los siguientes comandos para crear el usuario y la base:

```
CREATE DATABASE keystone;  
GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'localhost' IDENTIFIED BY  
'icc115';  
GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%' IDENTIFIED BY  
'icc115';
```

3. Instalamos el paquete de Keystone:

```
apt install keystone
```

4. Modificamos el archivo /etc/keystone/keystone.conf :

```
[database]  
connection = mysql+pymysql://keystone:icc115@172.29.236.5/keystone  
  
[token]  
provider = fernet
```

5. Poblamos la base de datos con el siguiente comando:

```
su -s /bin/sh -c "keystone-manage db_sync" keystone
```

6. Inicializamos los repositorios para los token Fernet:

```
keystone-manage fernet_setup --keystone-user keystone --keystone-group keystone
keystone-manage credential_setup --keystone-user keystone --keystone-group keystone
```

7. Inicializamos el servicio de Keystone. Considerar que el nombre “RegionZero” puede cambiarse, pero siempre deberá utilizarse el mismo nombre que se decida en la configuración de los demás servicios:

```
keystone-manage bootstrap --bootstrap-password icc115 \
--bootstrap-admin-url http://172.29.236.5:5000/v3/ \
--bootstrap-internal-url http://172.29.236.5:5000/v3/ \
--bootstrap-public-url http://172.29.236.5:5000/v3/ \
--bootstrap-region-id RegionZero
```

8. Creamos el archivo admin-openrc:

```
nano ~/admin-openrc
```

```
export OS_PROJECT_DOMAIN_NAME=Default
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=icc115
export OS_AUTH_URL=http://172.29.236.5:5000/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2
export OS_REGION_NAME=RegionZero
#export OS_REGION_NAME=RegionOne
```

9. Cargamos las variables del archivo:

```
. admin-openrc
```

10. Creamos el proyecto service:

```
openstack project create --domain default --description "Service Project" service
```

11. En este momento crearemos todos los usuarios de los demás servicios de la región uno, como el servicio de glance de prueba para la región cero:

```
nano ~/script.sh
```

```
#Keystone para RegionOne
openstack endpoint create --region RegionOne identity public http://172.29.236.5:5000
openstack endpoint create --region RegionOne identity internal http://172.29.236.5:5000
openstack endpoint create --region RegionOne identity admin http://172.29.236.5:5000

#Glance, que tiene 6 endpoint por estar en dos regiones
openstack user create --domain default --password icc115 glance
openstack role add --project service --user glance admin
openstack service create --name glance --description "OpenStack Image" image
openstack endpoint create --region RegionOne image public http://172.29.236.11:9292
openstack endpoint create --region RegionOne image internal http://172.29.236.11:9292
openstack endpoint create --region RegionOne image admin http://172.29.236.11:9292
openstack endpoint create --region RegionZero image public http://172.29.236.5:9292
openstack endpoint create --region RegionZero image internal http://172.29.236.5:9292
openstack endpoint create --region RegionZero image admin http://172.29.236.5:9292

#Placement
openstack user create --domain default --password icc115 placement
openstack role add --project service --user placement admin
openstack service create --name placement --description "Placement API" placement
openstack endpoint create --region RegionOne placement public http://172.29.236.11:8780
openstack endpoint create --region RegionOne placement internal
http://172.29.236.11:8780
openstack endpoint create --region RegionOne placement admin http://172.29.236.11:8780

#Nova
openstack user create --domain default --password icc115 nova
openstack role add --project service --user nova admin
openstack service create --name nova --description "OpenStack Compute" compute
openstack endpoint create --region RegionOne compute public
http://172.29.236.11:8774/v2.1
openstack endpoint create --region RegionOne compute internal
http://172.29.236.11:8774/v2.1
openstack endpoint create --region RegionOne compute admin
http://172.29.236.11:8774/v2.1

#Neutron
```

```
openstack user create --domain default --password icc115 neutron
openstack role add --project service --user neutron admin
openstack service create --name neutron --description "OpenStack Networking" network
openstack endpoint create --region RegionOne network public http://172.29.236.11:9696
openstack endpoint create --region RegionOne network internal http://172.29.236.11:9696
openstack endpoint create --region RegionOne network admin http://172.29.236.11:9696

#Cinder
openstack user create --domain default --password icc115 cinder
openstack role add --project service --user cinder admin
openstack service create --name cinderv3 --description "OpenStack Block Storage"
volumev3
openstack endpoint create --region RegionOne volumev3 public
http://172.29.236.11:8776/v3/%\(\project_id\)s
openstack endpoint create --region RegionOne volumev3 internal
http://172.29.236.11:8776/v3/%\(\project_id\)s
openstack endpoint create --region RegionOne volumev3 admin
http://172.29.236.11:8776/v3/%\(\project_id\)s
```

12. Cambiamos los permisos del archivo y lo ejecutamos:

```
chmod +x ~/script-openstack.sh

cd ~

./script-openstack.sh
```

13. Reiniciamos el servidor de apache:

```
service apache2 restart
```

8. *Glance.*

Considerar que este servicio está siendo instalado en la infraestructura para poder tener el selector de regiones activado en el panel de administración web. Los pasos son los siguientes:

1. Entramos a la consola de MariaDB:

```
mariadb
```

2. Instalamos el paquete de Glance:

```
apt install glance
```

3. Modificamos el archivo de configuración de Glance:

```
nano /etc/glance/glance-api.conf
```

```
[database]
connection = mysql+pymysql://glance:icc115@172.29.236.5/glance

[keystone_authtoken]
www_authenticate_uri = http://172.29.236.5:5000
auth_url = http://172.29.236.5:5000
memcached_servers = 172.29.236.5:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = glance
password= icc115

[paste_deploy]
flavor = keystone

[glance_store]
stores = file,http
default_store = file
filesystem_store_datadir = /var/lib/glance/images/
```

4. Poblamos la base de Glance con el siguiente comando:

```
su -s /bin/sh -c "glance-manage db_sync" glance
```

5. Reiniciamos el servicio:

```
service glance-api restart
```


9. *Horizon.*

1. Instalamos los paquetes con el siguiente comando:

```
apt install openstack-dashboard
```

2. Modificamos el archivo de configuración del dashboard:

```
DEBUG = True

SESSION_ENGINE = 'django.contrib.sessions.backends.cache'

CACHES = {
    'default': {
        'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',
        'LOCATION': '172.29.236.5:11211',
    }
}

OPENSTACK_HOST = "172.29.236.5"
OPENSTACK_KEYSTONE_URL = "http://%s:5000/identity/v3" %
OPENSTACK_HOST

TIME_ZONE = "America/El_Salvador"

OPENSTACK_API_VERSIONS = {
    "identity": 3,
    "image": 2,
    "volume": 3,
}

OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = "Default"

OPENSTACK_KEYSTONE_DEFAULT_ROLE = "user"

OPENSTACK_ENDPOINT_TYPE = "publicURL"

REST_API_REQUIRED_SETTINGS = ['OPENSTACK_HYPERVISOR_FEATURES',
                              'LAUNCH_INSTANCE_DEFAULTS',
                              'OPENSTACK_IMAGE_FORMATS']
IMAGES_ALLOW_LOCATION = False

HORIZON_IMAGES_UPLOAD_MODE = 'legacy'
```

3. Reiniciamos el servicio de apache:

```
systemctl restart apache2.service
```

D. Región

1. Preparando nodos.

a) Nodo Deployment.

Ejemplo de configuración de Netplan:

```
network:
  ethernets:
    enp1s0:
      addresses: [192.168.16.175/24]
      nameservers:
        addresses: [8.8.8.8]
      routes:
        - to: default
          via: 192.168.16.1
          metric: 1000
  vlans:
    enp1s0.10:
      id: 10
      link: enp1s0
      macaddress: 00:11:00:00:bb:aa
  bridges:
    br-mgmt:
      addresses: [172.29.236.2/22]
      interfaces: [enp1s0.10]
      macaddress: 60:f4:12:dd:70:cc
      nameservers:
        addresses: [8.8.8.8]
      routes:
        - to: default
          via: 172.29.236.1
          metric: 100
  version: 2
```

Se recomienda actualizar los paquetes del sistema y kernel:

```
apt update
```

```
apt dist-upgrade
```

Reiniciamos el host de despliegue. Luego, instalamos los siguientes paquetes y activamos el módulo 8021q:

```
apt install build-essential git chrony openssh-server python3-dev sudo vlan  
modprobe 8021q  
lsmod | grep 8021q
```

(1) Configurando llaves SSH.

1. Ejecutar en el nodo de despliegue, considerando no incluir passphrases para la llave, con el objetivo de reducir la intervención del usuario:

```
ssh-keygen
```

2. Copiar llave a los nodos objetivo:

```
ssh-copy-id root@172.29.236.11  
ssh-copy-id root@172.29.236.21  
ssh-copy-id root@172.29.236.31
```

3. Probar conexión con los nodos control-plane y compute.

(2) Instalar dependencias y repositorio.

1. Clonar el repositorio de OpenStack-Ansible Git en /opt/openstack-ansible:

```
git clone -b 23.4.3 https://opendev.org/openstack/openstack-ansible /opt/openstack-ansible
```

2. Nos dirigimos a la carpeta creada y ejecutamos el comando:

```
cd /opt/openstack-ansible  
scripts/bootstrap-ansible.sh
```

b) Nodos Control-Plane y Compute.

Ejemplo de configuración de Netplan para el Control-Plane:

```
network:
  ethernets:
    enp10s0:
      dhcp4: false
      dhcp6: false
    enp1s0:
      addresses: [192.168.16.145/24]
      nameservers:
        addresses: [8.8.8.8]
      routes:
        - to: default
          via: 192.168.16.1
          metric: 1000
    enp9s0:
      dhcp4: false
      dhcp6: false
  vlans:
    enp1s0.10:
      id: 10
      link: enp1s0
      macaddress: 00:01:00:00:00:10
    enp1s0.20:
      id: 20
      link: enp1s0
      macaddress: 00:02:00:00:00:20
    enp1s0.30:
      id: 30
      link: enp1s0
      macaddress: 00:01:00:00:00:30
  bridges:
    br-mgmt:
      addresses: [172.29.236.11/22]
      interfaces: [enp1s0.10]
      macaddress: 60:f4:12:ac:32:aa
      nameservers:
        addresses: [8.8.8.8]
      routes:
        - to: default
          via: 172.29.236.1
          metric: 100
    br-storage:
      interfaces: [enp1s0.20]
      addresses: [172.29.244.11/22]
      macaddress: 60:f4:12:bb:ff:55
    br-vxlan:
```

```
interfaces: [enp1s0.30]
addresses: [172.29.240.11/22]
macaddress: 60:f4:12:ac:34:89
br-vlan:
  interfaces: [enp10s0]
  macaddress: 60:f4:12:ac:34:52
version: 2
```

Ejemplo de configuración de Netplan para nodos Compute:

```
network:
  ethernets:
    enp1s0:
      dhcp4: false
      dhcp6: false
      optional: true
    enp9s0:
      dhcp4: false
      dhcp6: false
  vlans:
    enp1s0.10:
      link: enp1s0
      id: 412
      macaddress: 10:10:12:12:11:11
    enp1s0.30:
      link: enp1s0
      id: 30
      macaddress: 20:20:21:21:22:22
    enp1s0.20:
      link: enp1s0
      id: 20
      macaddress: 30:30:31:31:32:32
  bridges:
    br-mgmt:
      interfaces: [enp1s0.10]
      addresses: [172.29.236.21/22]
      macaddress: 12:12:11:11:10:10
      nameservers:
        addresses: [8.8.8.8]
      routes:
        - to: default
          via: 172.29.236.1
          metric: 100
    br-vxlan:
      interfaces: [enp1s0.30]
      addresses: [172.29.240.21/22]
      macaddress: 22:22:21:21:20:20
```

```
br-storage:
  interfaces: [enp1s0.20]
  addresses: [172.29.244.21/22]
  macaddress: 32:32:31:31:30:30
br-vlan:
  interfaces: [enp1s0]
  macaddress: 42:42:41:41:4e:4f
version: 2
```

(1) Instalando paquetes.

Se recomienda actualizar los paquetes del sistema y kernel:

```
apt update
apt dist-upgrade
```

Reiniciamos el host de despliegue. Luego, instalamos los siguientes paquetes y activamos el módulo de soporte para vlans:

```
apt install bridge-utils debootstrap openssh-server tcpdump vlan python3
apt install linux-modules-extra-$(uname -r)
modprobe 8021q
lsmod | grep 8021q
```

2. Configurando despliegue de Ansible.

OpenStack-Ansible depende de varios archivos que son utilizados para construir el inventario de Ansible. Realizaremos los siguientes pasos:

```
cp -r /opt/openstack-ansible/etc/openstack_deploy /etc
cd /etc/openstack_deploy
cp openstack_user_config.yml.example openstack_user_config.yml
```

Luego, modificaremos el archivo **openstack_user_config.yml** de la siguiente manera:

```
---
cidr_networks:
```

container: 172.29.236.0/22
tunnel: 172.29.240.0/22
storage: 172.29.244.0/22

used_ips:

- "172.29.236.1,172.29.236.50"
- "172.29.240.1,172.29.240.50"
- "172.29.244.1,172.29.244.50"

global_overrides:

external_lb_vip_address: 192.168.16.99
internal_lb_vip_address: 172.29.236.11
management_bridge: "br-mgmt"

provider_networks:

- network:
 - group_binds:
 - all_containers
 - hosts
 - type: "raw"
 - container_bridge: "br-mgmt"
 - container_interface: "eth1"
 - container_type: "veth"
 - ip_from_q: "container"
 - is_container_address: true
- network:
 - group_binds:
 - glance_api
 - cinder_api
 - cinder_volume
 - nova_compute
 - type: "raw"
 - container_bridge: "br-storage"
 - container_type: "veth"
 - container_interface: "eth2"
 - ip_from_q: "storage"
- network:
 - group_binds:
 - neutron_linuxbridge_agent
 - container_bridge: "br-vxlan"
 - container_type: "veth"
 - container_interface: "eth10"
 - ip_from_q: "tunnel"
 - type: "vxlan"
 - range: "1:1000"
 - net_name: "vxlan"
- network:
 - group_binds:
 - neutron_linuxbridge_agent

```
    container_bridge: "br-vlan"
    container_type: "veth"
    container_interface: "eth11"
    type: "vlan"
    range: "1:4094"
    net_name: "vlan"
- network:
  group_binds:
    - neutron_linuxbridge_agent
    container_bridge: "br-vlan"
    container_type: "veth"
    container_interface: "eth12"
#la interfaz depende de la que se tenga en la vm
  host_bind_override: "enp9s0"
  type: "flat"
  net_name: "flat"

shared-infra_hosts:
  infra2:
    ip: 172.29.236.11

repo-infra_hosts:
  infra2:
    ip: 172.29.236.11

os-infra_hosts:
  infra2:
    ip: 172.29.236.11

identity_hosts:
  infra2:
    ip: 172.29.236.11

network_hosts:
  infra2:
    ip: 172.29.236.11

compute_hosts:
  compute2:
    ip: 172.29.236.21
  compute3:
    ip: 172.29.236.22

storage-infra_hosts:
  infra2:
    ip: 172.29.236.11

storage_hosts:
  infra2:
```



```
ip: 172.29.236.11
```

```
haproxy_hosts:
```

```
  infra2:
```

```
    ip: 172.29.236.11
```

Modificaremos el archivo de **user_variables.yml** agregando el nombre de la región:

```
---
#Configuraciones generales
debug: true
service_region: RegionOne
install_method: source
memcached_servers: 172.29.236.5

#Configuración de Ceph

ceph_keyrings_dir: /etc/openstack_deploy/ceph-keyrings
ceph_conf_file: |
  [global]
    fsid = eac4e7b4-562c-11ee-a68a-bb385a8f3276
    mon_host =
[v2:172.29.244.41:3300/0,v1:172.29.244.41:6789/0],[v2:172.29.244.42:3300/0,v1:172.29.2
44.42:6789/0],[v2:172.29.244.43:3300/0,v1:172.29.244.43:6789/0],[v2:172.29.244.44:330
0/0,v1:172.29.244.44:6789/0],[v2:172.29.244.45:3300/0,v1:172.29.244.45:6789/0]

glance_ceph_client: glance
glance_default_store: rbd
glance_rbd_store_pool: images

cinder_ceph_client: cinder
cinder_backends:
  rbd:
    volume_driver: cinder.volume.drivers.rbd.RBDDriver
    rbd_pool: volumes
    rbd_ceph_conf: /etc/ceph/ceph.conf
    rbd_store_chunk_size: 8
    volume_backend_name: rbd
    rbd_user: "{{ cinder_ceph_client }}"
    rbd_secret_uuid: "{{ cinder_ceph_client_uuid }}"
    report_discard_supported: true

cinder_service_backup_program_enabled: true
cinder_service_backup_driver: cinder.backup.drivers.ceph.CephBackupDriver
cinder_service_backup_ceph_user: cinder-backup
cinder_service_backup_ceph_pool: backups
```

```
nova_libvirt_images_rbd_pool: vms
```

Finalmente, ejecutamos las siguientes órdenes para crear el archivo de contraseñas:

```
cd /opt/openstack-ansible  
./scripts/pw-token-gen.py --file /etc/openstack_deploy/user_secrets.yml
```

Para configurar los keyrings de Ceph que generamos en la sección anterior, realizamos los siguientes comandos:

```
mkdir -p /etc/openstack_deploy/ceph-keyrings  
  
cat <<EOF > /etc/openstack_deploy/ceph-keyrings/cinder.keyring  
[client.cinder]  
    key = AQCNDwplsnU4CBAAAn7rVAoNKxzm4q2EOIaxy6g==  
  
EOF  
  
cat <<EOF > /etc/openstack_deploy/ceph-keyrings/glance.keyring  
[client.glance]  
    key = AQCODwplLckMGhAA3rngP59cvEcXZ0kLWVXRkQ==  
  
EOF  
  
cat <<EOF > /etc/openstack_deploy/ceph-keyrings/cinder-backup.keyring  
[client.cinder-backup]  
    key = AQAAt+ydlbovzEBAATI5m/6FXNlArxbFU9ZCZDA==  
  
EOF
```

Para configurar cinder como contenedor en lugar de instalarlo directamente sobre el nodo, realizamos los siguientes comandos:

```
rm /etc/openstack_deploy/env.d/cinder.yml  
cat <<EOF > /etc/openstack_deploy/env.d/cinder.yml  
---  
container_skel:  
  cinder_volumes_container:  
    properties:  
      is_metal: false  
EOF
```

3. *Ejecutando playbooks.*

Son tres los playbooks ejecutados:

1. setup-hosts.yml
2. setup-infrastructure.yml
3. setup-openstack.yml

Debemos considerar que es un proceso que lleva cierto tiempo para completarse. De forma aproximada, OpenStack comparte los siguientes tiempos:

Tabla 4

Tiempo estimado del despliegue del prototipo.

Equipo	Tiempo
Equipos con disco SSD	30 - 50 minutos
Máquina virtual con disco SSD	45 - 60 minutos
Equipos con HDD	90 - 120 minutos

La tabla 4 muestra el tiempo estimado según el disco duro que se encuentre en los recursos de la propia maquina o máquina virtual para el despliegue del prototipo

Ejecutar los siguientes comandos para correr los playbooks:

```
cd /opt/openstack-ansible/playbooks
openstack-ansible setup-infrastructure.yml --syntax-check
openstack-ansible setup-hosts.yml
openstack-ansible setup-infrastructure.yml
openstack-ansible setup-openstack.yml
```

4. *Verificación del despliegue.*

Para ingresar a ver los contenedores del nodo Control-Plane, en consola ejecutamos los siguientes comandos:

```
lxc-ls -f
```

En el contenedor de utilidades se encuentra el archivo **openrc** para utilizar los comandos de OpenStack en cli:

```
lxc-attach -n `lxc-ls -1 | grep utility | head -n 1`  
. ~/openrc
```

Por ejemplo, para listar a los usuarios debemos ejecutar:

```
openstack user list
```

Para ingresar a otro contenedor nos apoyamos en la órden `lxc-attach`, listando previamente los contenedores para saber el nombre:

```
lxc-ls  
lxc-attach -n nombre_del_contenedor
```

Para verificar el dashboard, accedemos en el navegador a la IP externa especificada en el archivo `/etc/openstack_deploy/openstack_user_config.yml`. Considerar que el dashboard emplea https. En nuestro caso, sería lo siguiente:

<https://192.168.16.145/>

Para autenticarnos, utilizaremos `admin` como usuario, y la contraseña puede revisarse en el archivo `/etc/openstack_deploy/user_secrets.yml` o en el archivo `openrc` que se encuentra en el contenedor `utility`, realizando un `cat` o `nano`.

```
cat /etc/openstack_deploy/user_secrets.yml | grep keystone
```

V. Integración

A. Modificando configuración de servicios de la región

1. *HAProxy*.

1. Dirigirse al archivo `/etc/haproxy/haproxy.cfg` del contenedor correspondiente a modificar el modo para el backend y frontend del servicio de Glance:

```
nano /etc/haproxy/haproxy.cfg
```

```
#mode http
mode tcp
```

2. Reiniciar el servicio:

```
service haproxy restart
```

2. *Glance.*

1. Ingresamos al contenedor de glance en el nodo control-plane de la región:

```
lxc-attach -n `lxc-ls -1 | grep glance | head -n 1`
```

2. Instalamos nano o el editor de nuestra preferencia:

```
apt install nano
```

3. Modificamos el archivo `/etc/glance/glance-api-conf`:

```
[keystone_authtoken]
#auth_url = http://172.29.236.11:5000/v3
#www_authenticate_uri = http://172.29.236.11:5000
auth_url = http://172.29.236.5:5000/v3
www_authenticate_uri = http://172.29.236.5:5000
#memcached_servers = 172.29.237.146:11211
memcached_servers = 172.29.236.5:11211
```

3. *Placement.*

1. Ingresamos al contenedor de placement en el nodo control-plane de la región:

```
lxc-attach -n `lxc-ls -1 | grep placement | head -n 1`
```

2. Instalamos nano o el editor de nuestra preferencia:

```
apt install nano
```

3. Modificamos el archivo `/etc/placement/placement.conf`

```
[keystone_authtoken]
#auth_url = http://172.29.236.11:5000/v3
#www_authenticate_uri = http://172.29.236.11:5000
auth_url = http://172.29.236.5:5000/v3
www_authenticate_uri = http://172.29.236.5:5000
#memcached_servers = 172.29.237.146:11211
memcached_servers = 172.29.236.5:11211
```

4. *Nova.*

a) **Nodo control-plane.**

1. Ingresamos al contenedor de nova en el nodo control-plane de la región:

```
lxc-attach -n `lxc-ls -1 | grep nova | head -n 1`
```

2. Instalamos nano o el editor de nuestra preferencia:

```
apt install nano
```

3. Modificamos el archivo `/etc/nova/nova.conf`

```
[cache]
#memcache_servers = 172.29.237.146:11211
memcache_servers = 172.29.236.5:11211

[neutron]
#auth_url = http://172.29.236.11:5000/v3
auth_url = http://172.29.236.5:5000/v3

[placement]
#auth_url = http://172.29.236.11:5000/v3
auth_url = http://172.29.236.5:5000/v3

[keystone_authtoken]
#auth_url = http://172.29.236.11:5000/v3
#www_authenticate_uri = http://172.29.236.11:5000
auth_url = http://172.29.236.5:5000/v3
www_authenticate_uri = http://172.29.236.5:5000
#memcached_servers = 172.29.237.146:11211
memcached_servers = 172.29.236.5:11211
```

b) Nodo compute.

1. Ingresamos al nodo compute de la región. Puede ser desde el nodo deployment de dicha región:

```
ssh root@172.29.236.22
```

2. Modificamos el archivo `/etc/nova/nova.conf`

```
#memcache_servers = 172.29.237.146:11211
memcache_servers = 172.29.236.5:11211

[neutron]
#auth_url = http://172.29.236.11:5000/v3
auth_url = http://172.29.236.5:5000/v3

[placement]
#auth_url = http://172.29.236.11:5000/v3
auth_url = http://172.29.236.5:5000/v3

[keystone_authtoken]
#auth_url = http://172.29.236.11:5000/v3
#www_authenticate_uri = http://172.29.236.11:5000
auth_url = http://172.29.236.5:5000/v3
www_authenticate_uri = http://172.29.236.5:5000
#memcached_servers = 172.29.237.146:11211
memcached_servers = 172.29.236.5:11211
```

5. Neutron.

a) Nodo control-plane.

1. Ingresamos al contenedor de neutrón en el nodo control-plane de región dos:

```
lxc-attach -n `lxc-ls -1 | grep neutron | head -n 1`
```

2. Instalamos nano o el editor de nuestra preferencia:

```
apt install nano
```

3. Modificamos el archivo `/etc/neutron/neutron.conf`

```
[placement]
#auth_url = http://172.29.236.11:5000/v3
auth_url = http://172.29.236.5:5000/v3

[nova]
#auth_url = http://172.29.236.11:5000/v3
auth_url = http://172.29.236.5:5000/v3

[keystone_authtoken]
#auth_url = http://172.29.236.11:5000/v3
#www_authenticate_uri = http://172.29.236.11:5000
auth_url = http://172.29.236.5:5000/v3
www_authenticate_uri = http://172.29.236.5:5000
#memcached_servers = 172.29.237.146:11211
memcached_servers = 172.29.236.5:11211
```

b) Nodo compute.

No se realizan cambios.

6. Cinder.

1. Ingresamos al contenedor de cinder en el nodo control-plane de la región:

```
lxc-attach -n `lxc-ls -1 | grep cinder | head -n 1`
```

2. Instalamos nano o el editor de nuestra preferencia:

```
apt install nano
```

3. Modificamos el archivo `/etc/cinder/cinder.conf`

```
[keystone_authtoken]
#auth_url = http://172.29.236.11:5000/v3
#www_authenticate_uri = http://172.29.236.11:5000
auth_url = http://172.29.236.5:5000/v3
www_authenticate_uri = http://172.29.236.5:5000
#memcached_servers = 172.29.237.146:11211
memcached_servers = 172.29.236.5:11211

[nova]
#auth_url = http://172.29.236.11:5000/v3
auth_url = http://172.29.236.5:5000/v3
```


B. Agregando el servicio swift a la región

1. Dirigirse al nodo ceph-admin y ejecutar el siguiente comando para verificar que ningún servicio se encuentre ejecutando en el puerto 9000:

```
netstat -tulnp | grep 9000
```

Debe haber una respuesta vacía del comando:

```
root@deployment:~# netstat -tulnp | grep 9000
root@deployment:~#
```

Figura 85: Ejecución del comando netstat.

De existir un servicio corriendo en ese puerto, escoger otro.

2. En el panel de administración de Ceph nos dirigimos a la sección “Cluster->Services:

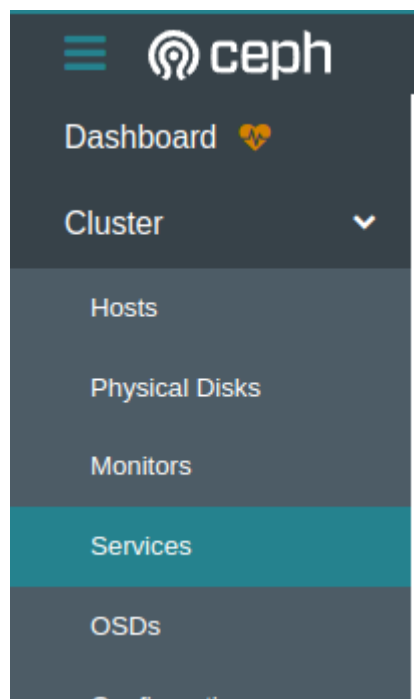


Figura 86: Opciones del menú Cluster de Ceph.

3. Damos clic en la opción “Create”:

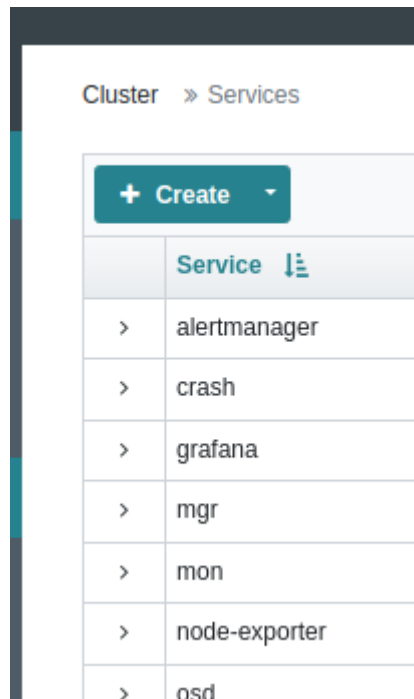


Figura 87: Listado de servicios desplegados de Ceph.

4. En la ventana desplegada creamos el servicio rgw con los siguientes parámetros (el valor de id puede variar). Escoger a conveniencia el hosts que tendrá el servicio:

The screenshot shows the 'Create Service' dialog box. The 'Type' field is set to 'rgw'. The 'Id' field is set to 'region-one' and has a green checkmark. There is an 'Unmanaged' checkbox which is unchecked. The 'Placement' field is set to 'Hosts'. Below this, there is a 'Hosts' section with a message 'There are no hosts.' and a list of hostnames: ceph-admin, ceph-mgr, ceph-mon, ceph-osd1, ceph-osd2, and ceph-osd3. The 'ceph-admin' host is selected with a green checkmark. At the bottom right, there are 'Cancel' and 'Create Service' buttons.

Figura 88: Formulario para creación de servicio en Ceph.

- Al dar clic en el botón “Create Service” será creado el servicio y se nos hará saber en el panel a través de una notificación:

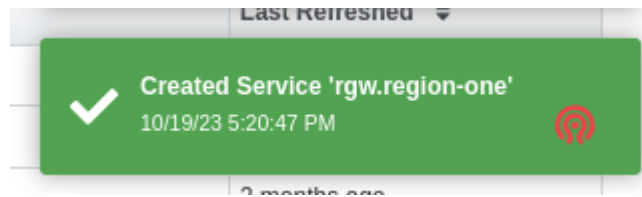


Figura 89: Notificación de servicio desplegado de Ceph.

- Probar el acceso del nodo ceph-admin al servicio Keystone del nodo infraestructura:

```
curl http://172.29.244.5:5000
```

- Configurar con los siguientes comandos en la consola de ceph-admin Keystone como método de autorización:

```
ceph config set client.rgw rgw_keystone_api_version 3
ceph config set client.rgw rgw_keystone_accepted_roles "admin,service"
ceph config set client.rgw rgw_keystone_admin_user swift
ceph config set client.rgw rgw_keystone_admin_password icc115
ceph config set client.rgw rgw_keystone_admin_domain default
ceph config set client.rgw rgw_keystone_admin_project service
ceph config set client.rgw rgw_s3_auth_use_keystone true
ceph config set client.rgw rgw_keystone_url http://172.29.244.5:5000
ceph config set client.rgw rgw_keystone_implicit_tenants true
ceph config set client.rgw rgw_keystone_verify_ssl false
ceph config set client.rgw rgw_swift_account_in_url true
```

- Reiniciar nodo ceph-admin.
- Ejecutar los siguientes comandos en OpenStack (nodo infraestructura, recordar tener cargadas las variables de autenticación):

```
openstack user create --domain default --password icc115 swift
openstack service create --name=swift --description="Swift Service" object-store
openstack role add --project service --user swift admin
openstack endpoint create --region RegionOne object-store public
http://172.29.244.41:9000/swift/v1/AUTH_%(project_id)s
openstack endpoint create --region RegionOne object-store admin
http://172.29.244.41:9000/swift/v1/AUTH_%(project_id)s
```

```
openstack endpoint create --region RegionOne object-store internal  
http://172.29.244.41:9000/swift/v1/AUTH_%\project_id\
```