UNIVERSIDAD DE EL SALVADOR

FACULTAD DE INGENIERÍA Y ARQUITECTURA

ESCUELA DE INGENIERÍA DE SISTEMAS INFOMÁTICOS

PROTOTIPO DE INFRAESTRUCTURA DE NUBE COMUNITARIA MULTI-REGIÓN ORIENTADA A PROPORCIONAR SERVICIOS FUNDAMENTALES

PRESENTADO POR:

VILMA ARELY BÁRCENAS CRUZ

DANIEL ALBERTO BAÑOS LÓPEZ

NOÉ SALVADOR PONCE MARTÍNEZ

PARA OPTAR AL TÍTULO DE:

INGENIERO DE SISTEMAS INFORMÁTICOS

CIUDAD UNIVERSITARIA, NOVIEMBRE DE 2023

UNIVERSIDAD DE EL SALVADOR

RECTOR:

MSC. JUAN ROSA QUINTANILLA QUINTANILLA

SECRETARIO GENERAL:

LIC. PEDRO ROSALÍO ESCOBAR CASTANEDA

FACULTAD DE INGENIERIA Y ARQUITECTURA

DECANO:

ING. LUIS SALVADOR BARRERA MANCÍA

SECRETARIO:

ARQ. RAUL ALEXANDER FABIAM ORELLANA

ESCUELA DE INGENIERÍA DE SISTEMAS INFORMÁTICOS

DIRECTOR:

ING. CESAR AUGUSTO GONZÁLEZ RODRÍGUEZ

UNIVERSIDAD DE EL SALVADOR

FACULTAD DE INGENIERIA 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 INFRAESTRUCTURA DE NUBE COMUNITARIA MULTI-REGIÓN ORIENTADA A PROPORCIONAR SERVICIOS FUNDAMENTALES

Presentado por:

VILMA ARELY BÁRCENAS CRUZ

DANIEL ALBERTO BAÑOS LÓPEZ

NOÉ SALVADOR PONCE MARTÍNEZ

Trabajo de Graduación Aprobado por:

Docente Asesor:

MSc. JULIO DAMIÁN MORALES AYALA

SAN SALVADOR, NOVIEMBRE DE 2023

Trabajo de Graduación Aprobado por:

Docente Asesor:

MSc. JULIO DAMIÁN MORALES AYALA

Agradecimientos

A mi madre y padre, por su apoyo incondicional y su esfuerzo extraordinario, que me ha permitido formarme como profesional. A mis hermanas y hermano, quienes también han contribuido al logro de esta meta y son parte importante de mi vida. A mi mejor amiga, por brindarme constante motivación para no rendirme y seguir adelante, recordándome siempre que «*todo estará bien».* A mis compañeros y docentes que han sido parte de este proceso.

Vilma Arely Bárcenas Cruz

Agradezco infinitamente a Dios que me ha ayudado a lo largo de todo este camino, sin él no habría llegado a esta etapa. Doy gracias a mis padres que lo han dado todo por mí y el apoyo a lo largo de mi formación académica. Gracias a mi familia, personas especiales de mi vida y amigos que siempre me apoyaron y motivaron a seguir adelante. Por último, me doy gracias a mí mismo por no haberme rendido y haber luchado hasta culminar esta etapa.

Daniel Alberto Baños López

Primeramente, gracias a Dios por darme la vida y permitirme lograr esta meta. Infinitas gracias a mis padres por todo el amor y la educación que me bridaron, por las noches de desvelo acompañándome y por las oraciones que hacían por mi futuro. Gracias a mi hermano por estar siempre cuando lo necesitaba. Por último, a todos los compañeros y compañeras que se convirtieron en amigos de por vida y me guiaron en mi camino universitario.

Noé Salvador Ponce Martínez

TABLA DE CONTENIDO

INDICE DE FIGURAS

INDICE DE TABLAS

1 Introducción

En la actualidad, donde la digitalización cada vez tiene más influencia en el mundo, la necesidad de infraestructuras de tecnología de la información (*TI*) eficientes, escalables y rentables ha experimentado un notable crecimiento. Las organizaciones buscan constantemente soluciones que, además de ser robustas y seguras, se adapten ágilmente a los cambios del panorama digital. En este marco, el presente Trabajo Final de Grado se centra en la construcción y evaluación de una infraestructura de nube comunitaria fundamentada en *OpenStack*, una plataforma reconocida por su flexibilidad, robustez y habilidad para integrarse con eficiencia en diferentes escenarios operativos.

El prototipo de infraestructura diseñado actúa como un nodo central, un epicentro de recursos y servicios que son accesibles y utilizables desde diversas regiones geográficas. Mediante la implementación de servicios federados de identificación, persistencia de sesión, una base de datos central y una interfaz de gestión intuitiva, esta infraestructura permite además de conectar regiones, facilitar la colaboración y proporcionar un acceso fluido a los recursos fundamentales para su operatividad.

2 Contexto

La implementación y gestión de infraestructuras de nube representan desafíos cruciales en el ámbito actual de la tecnología de la información. Con la adopción cada vez más generalizada de la computación en la nube, las organizaciones y comunidades están experimentando transformaciones significativas en el uso de recursos informáticos, logrando una escalabilidad, eficiencia y flexibilidad sin precedentes. En este escenario, el presente trabajo final de grado se sumerge en un proyecto ambicioso: la creación de un prototipo de infraestructura de nube comunitaria basada en *OpenStack*, diseñada para servir a múltiples regiones geográficas.

2.1. Descripción

En el presente proyecto, el prototipo que se ha propuesto está destinado a servir como núcleo central para diversas regiones geográficas. La infraestructura busca proveer de servicios fundamentales que las regiones puedan consumir para su operatividad.

El objetivo principal es una integración transparente entre las regiones, que permita la colaboración e integración de sus recursos para formar una nube comunitaria unificada en un sitio central.

A través del desarrollo y análisis de este proyecto, se busca desvelar los desafíos, oportunidades y ventajas que implica concebir una infraestructura de tal magnitud.

Para alcanzar este objetivo, la infraestructura debe incluir los siguientes componentes esenciales que actuarán como *API* para las regiones:

Servicio de Identificación: Implementar un servicio de identificación que sirva para la autenticación de usuarios.

Servicio de federación de usuarios: Es un componente importante para que las regiones cuenten con usuarios federados, ofreciendo accesos seguros y unificados para todas las regiones que serán parte de la nube comunitaria.

Persistencia de Sesión: Instaurar un servicio de persistencia de sesiones que garantice a los usuarios una actividad constante y sin interrupciones entre las distintas regiones y servicios.

Base de datos centralizada: Debe establecerse una base de datos que actúe de forma centralizada para todas las operaciones de la nube comunitaria.

Interfaz de Gestión: Con el fin de brindar a los administradores una herramienta eficaz para monitorizar y gestionar los recursos y operaciones de las regiones, se debe proveer de una interfaz gráfica.

2.2. Antecedentes

En la era actual de la tecnología de la información, la computación en la nube se ha convertido en un pilar fundamental para empresas, organizaciones y comunidades que buscan maximizar los recursos informáticos de manera eficiente y escalable. La posibilidad de desplegar recursos rápidamente y a bajo costo ha impulsado un crecimiento exponencial en la adopción de la nube a nivel mundial.

Sin embargo, al intentar capitalizar los beneficios de la nube, las organizaciones y comunidades enfrentan desafíos complejos en gestión, seguridad y colaboración entre múltiples entornos en la nube (Insider, 2021).

Ante este escenario, surgió la necesidad de diseñar una infraestructura de nube que funcione como un nodo central, conectando y coordinando múltiples regiones para compartir recursos de manera efectiva, independientemente de la ubicación geográfica o configuraciones específicas.

Considerando que la infraestructura propuesta facilita la colaboración y la integración de recursos entre múltiples regiones, resulta importante comprender detalles específicos, como el momento en el que la computación en la nube comenzó a soportar múltiples regiones y las razones detrás de esta evolución.

La computación en la nube comenzó a admitir regiones cuando los proveedores de este servicio empezaron a organizar sus infraestructuras en diferentes regiones geográficas, conocidas como regiones de nube. Las regiones son esencialmente áreas geográficas arbitrarias, lo que significa que cada proveedor puede tener regiones distintas con nombres y límites variables. Una región es un área geográfica en la que se encuentra un centro de datos. Los proveedores mantienen centros de datos en diferentes ubicaciones y permiten a los clientes elegir entre ellos cuando despliegan una carga de trabajo. Las regiones permiten ubicar los recursos en la nube cerca de los clientes, tanto internos como externos. Cuanto más cerca estén los clientes de la región en la que se encuentran los recursos, más rápida y mejor será su experiencia (Estrin, 2020) .

La capacidad de utilizar múltiples regiones brinda ventajas importantes, como la redundancia, la alta disponibilidad, la baja latencia y la capacidad de cumplir con requisitos y regulaciones locales. Por lo tanto, la adopción de regiones en la nube en sus diferentes modelos de despliegue ha crecido a medida que las organizaciones buscan aprovechar estas ventajas. (Zhang, 2023).

En cuanto a *OpenStack*, de acuerdo con el histórico de sus lanzamientos, las múltiples regiones se comenzaron a soportar en el lanzamiento llamado *Grizzly en 2013*. (OpenStack, 2013)

Los beneficios de esta característica dentro de OpenStack (OpenStack Docs, 2020) incluyen:

Escalabilidad: Permite a los operadores escalar sus implementaciones de *OpenStack* agregando nuevas regiones. Esto se puede hacer para satisfacer la creciente demanda de recursos o para cumplir con los requisitos de rendimiento específicos de una aplicación.

Disponibilidad: Ayuda a garantizar la disponibilidad de los servicios de *OpenStack* al distribuir la carga de trabajo entre varias regiones. Esto puede ayudar a proteger contra las fallas de *hardware* o *software* en una región individual.

Seguridad: Mejora la seguridad de las implementaciones de *OpenStack* al aislar los recursos en diferentes regiones. Esto puede ayudar a proteger contra ataques dirigidos o la pérdida de datos.

A pesar de todos los avances y beneficios que han surgido con la adaptación de la computación en la nube y la implementación de soporte multi-región en plataformas como OpenStack, surgen inevitablemente retos y problemas específicos.

Estos desafíos, inherentes a la creciente complejidad y demanda de los sistemas modernos, deben ser abordados para que las organizaciones puedan aprovechar al máximo estas innovaciones. A continuación, se detalla el planteamiento del problema que esta investigación busca resolver.

2.3. Planteamiento del Problema

La creación de una nube comunitaria basada en *OpenStack*, que posea la capacidad de admitir un número ilimitado de regiones, representa un desafío tecnológico en el cual se necesitan abordar varios problemas y consideraciones críticas. Esto implica diseñar una arquitectura escalable que permita la adición y gestión sencilla de nuevas regiones, manteniendo al mismo tiempo un rendimiento óptimo y garantizar la adaptabilidad a cambios en la carga de trabajo y la demanda de recursos de manera dinámica.

Otro desafío es que debe admitir usuarios federados, lo que implica la gestión de identidades de usuarios de diferentes regiones. Asimismo, garantizar la persistencia de las sesiones de estos para brindar una experiencia de usuario coherente y sin interrupciones.

Estos desafíos deben abordarse de manera integral para garantizar el éxito y la eficiencia de la infraestructura y para proporcionar una experiencia de usuario robusta y segura.

Además, es esencial considerar el contexto empresarial y la relevancia de la implementación de una Nube Comunitaria basada en *OpenStack* para empresas del sector TI dedicadas a proveer servicios de *cloud computing*. Por lo tanto, la implementación de esta no solo es un desafío tecnológico, sino también una oportunidad estratégica para las empresas del sector, lo que permitiría fortalecer su posición en el mercado y brindar soluciones más avanzadas y personalizadas a sus clientes.

2.4. Delimitación del Problema

En el contexto de este trabajo de graduación, se ha delimitado el proyecto a enfocarse en el diseño y desarrollo de un prototipo de Infraestructura para una nube comunitaria, así como otros componentes mencionados anteriormente.

Para la construcción del prototipo se trabajó en un ambiente de máquinas virtuales sobre sistema operativo Ubuntu y no se consideran aspectos específicos sobre pruebas en hardware físico ni se presentan detalles sobre presupuestos concretos para una implementación en máquinas reales.

2.5. Justificación

La creación de una Nube Comunitaria basada en *OpenStack* con la capacidad de admitir un número ilimitado de regiones es un proyecto de considerable importancia y relevancia, tanto desde un punto de vista tecnológico como desde una perspectiva de satisfacción de necesidades específicas.

En la actualidad, las organizaciones, comunidades y usuarios individuales dependen cada vez más de servicios en la nube para almacenar datos, ejecutar aplicaciones y acceder a recursos computacionales. Esta creciente demanda de servicios en la nube subraya la importancia de crear una infraestructura que permita el despliegue eficiente y la gestión de recursos en la nube.

La idea de una Nube Comunitaria se basa en la colaboración y el uso compartido de recursos entre múltiples entidades o regiones (Marinos & Briscoe, 2009).

Lo anterior es especialmente relevante para comunidades, empresas y otras organizaciones que desean compartir recursos informáticos, almacenamiento de datos y servicios en la nube de manera colaborativa y eficiente.

La capacidad de admitir un número ilimitado de regiones y usuarios federados proporciona una flexibilidad inigualable. Esto permite a la infraestructura de la Nube Comunitaria adaptarse y crecer de manera dinámica a medida que aumenta la demanda de recursos informáticos y servicios en la nube, sin la necesidad de reemplazar sistemas o invertir en infraestructura adicional constantemente.

La Nube Comunitaria brinda un entorno ideal para la colaboración y la innovación entre regiones y entidades participantes. Al compartir recursos y servicios, se fomenta la creación conjunta de aplicaciones, el desarrollo de proyectos de investigación y el intercambio de conocimientos, lo que puede conducir a avances significativos en diversas disciplinas.

Este proyecto aborda desafíos tecnológicos y operativos clave que pueden tener un impacto significativo en la eficiencia y la capacidad de las organizaciones y comunidades para ofrecer servicios en la nube de alta calidad a sus usuarios finales.

3 Marco Teórico

Para poder comprender y adentrarse en el amplio mundo de la computación en la nube, es necesario establecer un sólido marco teórico que nos permita contextualizar y comprender los conceptos clave que están detrás de esta revolucionaria tecnología.

3.1. Virtualización

La virtualización es la creación, mediante *software*, de una versión virtual de recursos tecnológicos, como plataformas de *hardware,* sistemas operativos, dispositivos de almacenamiento o recursos de red (Energy, 2015), ya sean completos o parciales en un recurso físico. Cada tipo de virtualización tiene sus propias características y beneficios.

La virtualización de sistemas operativos en servidores permite reducir el número de servidores físicos necesarios para correr los servicios necesarios, lo que permite reducir el espacio necesario y la cantidad de equipo especializado para mantener en funcionamiento la infraestructura (IONOS, 2023) . Al resultado de virtualizar recursos físicos se le llama Máquinas Virtuales.

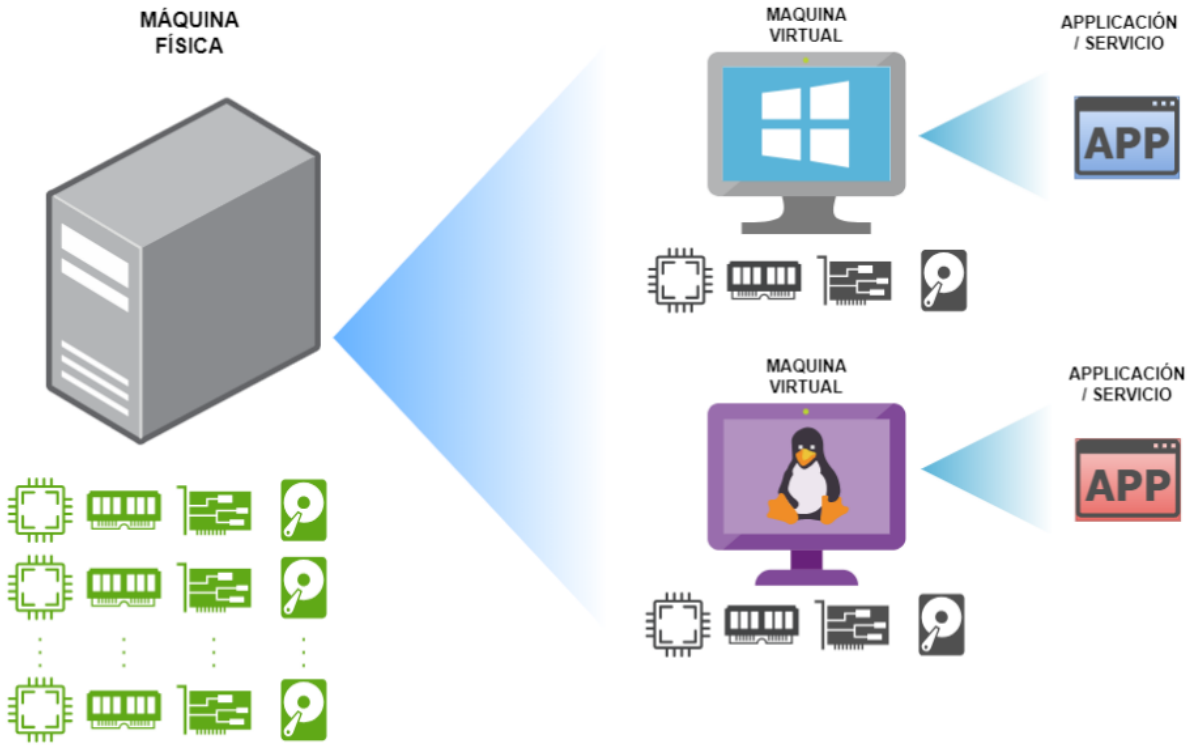


Figura 3.1 Conceptualización de Virtualización

La virtualización es una de las principales claves para la computación en la nube, tema que está explicado en el siguiente capítulo, y permite crear la cantidad de máquinas virtuales que se requieran dentro de un mismo equipo físico.

Dentro del lenguaje técnico de virtualización y computación en la nube se puede presentar el funcionamiento de la virtualización mediante capas de abstracción, para que sea más fácil comprender la separación de procesos y componentes que tienen que ver en la creación de recursos virtualizados.

En una configuración tradicional de maquina física se puede representar de la siguiente manera:

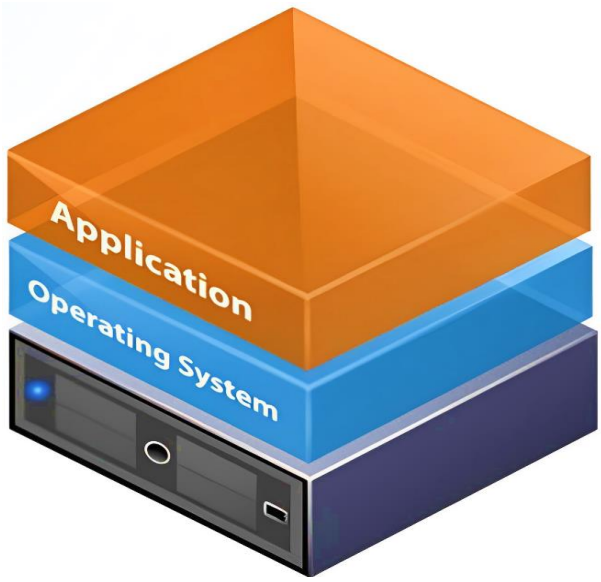


Figura 3.2. Abstracción de Arquitectura Tradicional. Adaptado de (Farina, 2021)

Está compuesta por el hardware del servidor o una maquina personal, sobre el cual corre el sistema operativo, y en la capa superior están funcionando las aplicaciones, programas o servicios. A esta arquitectura tradicional se le suele llamar *Bare-Metal Server*.

Para la arquitectura de virtualización tenemos siempre el *hardware* del servidor, pero ahora nos encontramos con una nueva capa de virtualización, la cual es el *software* que nos va a permitir emular máquinas virtuales y poder montar uno o varios sistemas operativos sobre los que correrán aplicaciones, programas o servicios que pueden estar completamente aislados unos de otros. A la nueva capa de virtualización se le llama Hipervisor.

Figura 3.3. Abstracción de Arquitectura de Virtualización. Tomada de (Desde Linux, 2023)

3.1.1. Hipervisores.

Como ya se dijo anteriormente, para crear entornos virtualizados se necesita la capa de virtualización, pero llamarle de esa manera, reduce la complejidad que se lleva a cabo para lograr crear máquinas virtuales. El hipervisor, es el *software* que está entre el *hardware* y los entornos virtuales.

Los hipervisores también son conocidos como Monitor de Máquina Virtual o VMM por sus siglas en inglés (*Virtual Machine Monitor*) y son los encargados de abstraer parte del *hardware* sobre el que se está ejecutando para que las máquinas virtuales usen una parte de éste al momento de estar encendida (Obasuyi & Sari, 2015). De esta manera, cada máquina virtual se ejecuta como si estuviera instalada en *hardware* real. Sin embargo, esto depende de la configuración y del tipo de hipervisor en el que esté corriendo.

3.1.2. Tipos de Hipervisores.

3.1.2.1. Hipervisor Tipo 1 o Bare-Metal.

Este tipo de hipervisor es un *software* que se ejecuta directamente en el *hardware* físico de una computadora o servidor y tiene acceso directo a la CPU, memoria y almacenamiento. Este es el más utilizado en entornos empresariales ya que es más eficiente para administrar recursos y dedicarlos a las máquinas virtuales. Ejemplos de hipervisor tipo 1 son *KVM, Microsoft Hyper-V, VMware vSphere, VMware ESX*, etc.

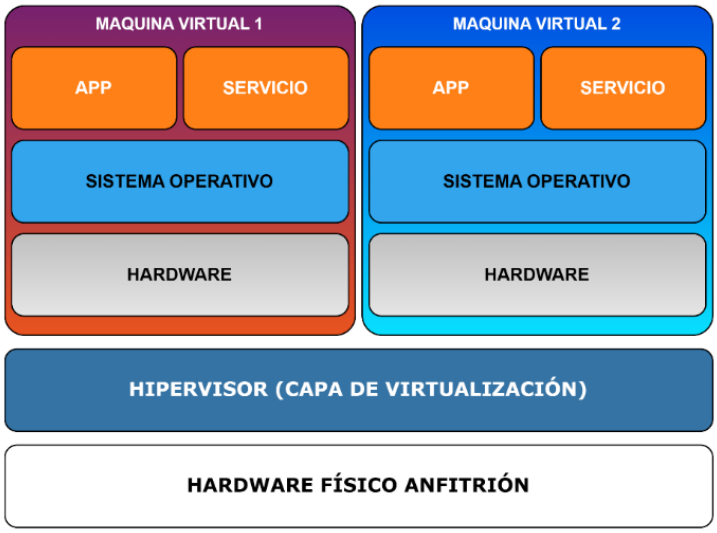


Figura 3.4. Arquitectura de Hipervisor Tipo 1

3.1.2.2. Hipervisor Tipo 2 o Alojado.

Este tipo de hipervisor necesita ejecutarse sobre un sistema operativo convencional ya instalado en el *hardware* como *Windows, MacOS o GNU/Linux*, por eso rara vez se puede ver en entornos empresariales, y está destinado a usuarios finales. Se ejecuta como un programa sobre el sistema operativo, y es utilizado por profesionales y usuarios casuales que necesitan tener más de un sistema operativo. Ejemplos de este tipo de hipervisor son *VMware Workstation y Player, VirtualBox*, etc.

Vienen con un set de herramientas adicionales para mejorar integración con el sistema operativo sobre el que se ejecutan, y mejoran la experiencia en términos de compartir archivos entre la máquina virtual y la maquina anfitrión. Aunque una desventaja es que cualquier problema que ocurra con el sistema operativo host afectará el sistema operativo de la máquina invitada y también afectará al propio hipervisor (Obasuyi & Sari, 2015).

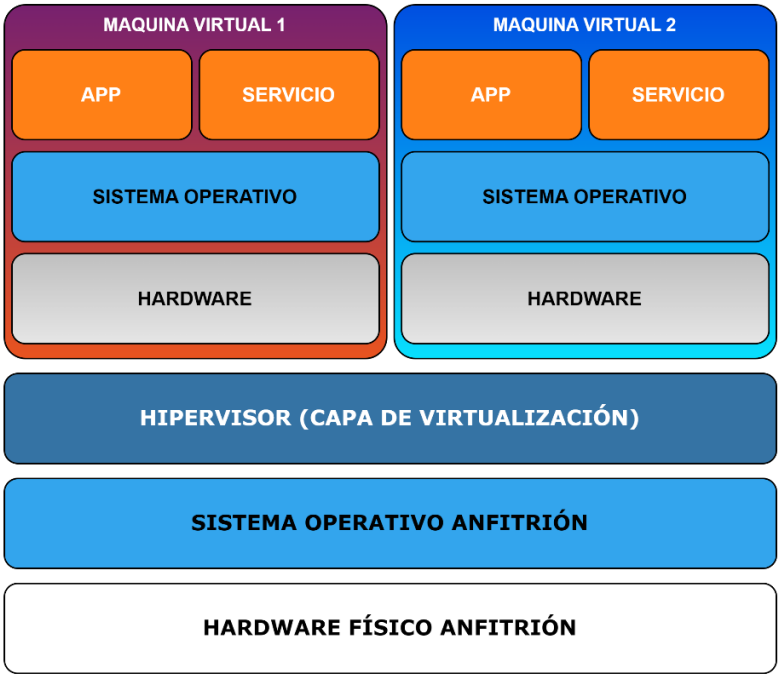


Figura 3.5. Arquitectura de Hipervisor Tipo 2

3.2. Computación en la Nube

La computación en la nube permite y transforma una infraestructura informática en una plataforma diversamente amplia en cuanto a utilidades que se le puedan dar, ya que se permite conectar a dicha infraestructura por internet y así poder usar todos los recursos informáticos que posee sin instalarlos ni mantenerlos en las instalaciones físicas (Vennam, s.f.).

La computación en la nube nos permite acceder a distintos tipos de recursos informáticos como: servidores, aplicaciones, almacenamiento de datos, herramientas de desarrollo, capacidades de red, sistemas operativos, entre otros.

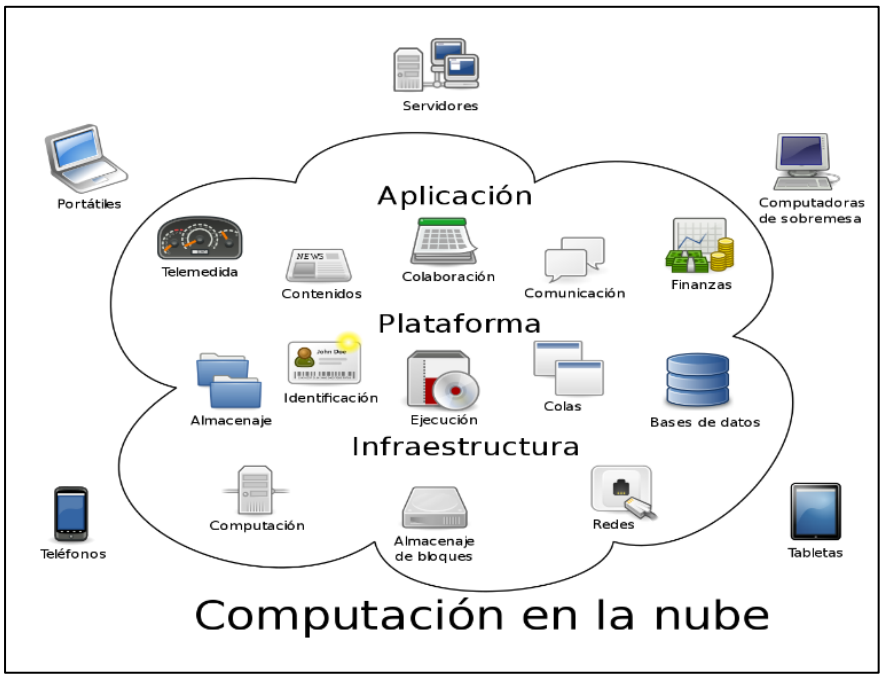


Figura 3.6. Computación en la nube. (Computación en la nube, 2023)

3.2.1. Estructura de Recursos en la Nube.

Al hablar de computación en la nube es importante conocer sobre los términos que se usan para describir los componentes.

3.2.1.1. Región.

Las regiones son subdivisiones de una nube que permiten la gestión de recursos y servicios de manera independiente dentro de una misma infraestructura. Las regiones se utilizan para separar física o lógicamente los recursos, lo que puede ser útil en los siguientes casos (RedHat, s.f.):

Geolocalización: Puedes configurar regiones para distribuir tus recursos en diferentes ubicaciones geográficas, lo que facilita la implementación de estrategias de recuperación ante desastres o la optimización de la latencia para los usuarios.

Separación de proyectos: Las regiones también se pueden utilizar para aislar proyectos o grupos de usuarios dentro de la misma infraestructura. Cada región puede tener su propio catálogo de servicios y cuotas.

Escala: Si se tiene una infraestructura muy grande, se puede dividir en regiones para facilitar la administración y escalabilidad.

3.2.1.2. Zona de disponibilidad.

Una zona de disponibilidad es parte de una región de una nube, y puede ser un centro de datos o un conjunto de centros de datos que están aislados físicamente, es decir, en locaciones distintas, pero dentro de una misma región (Zhang, 2023).

Dentro de una región puede haber más de dos, pero generalmente hay 3 zonas de disponibilidad que están aisladas, pero se conectan mediante redes troncales privadas de baja latencia que proporciona un menor coste y una latencia de red entre regiones (Medrano, 2023), y cada una de las zonas de disponibilidad tiene alimentación y seguridad física independiente, esto es para seguridad y poder aislarse si ocurre algún problema con algún centro de datos.

Los grandes proveedores de nube pública permiten tener redundancia entre zonas de disponibilidad, por si falla una, todos los servicios que proveía como redes, computo, aplicaciones, almacenamiento, etc., pueden seguir funcionando ya que se conmuta el servicio al siguiente centro de datos u otra zona de disponibilidad.

3.2.1.3. Dominio.

En el contexto de algunas plataformas de computación en la nube, especialmente en *OpenStack* (una plataforma de nube de código abierto), los dominios son contenedores de alto nivel para proyectos, usuarios y grupos. Administran los componentes de identidad basados en *Keystone*. Principalmente ayudan a dar el seguimiento a la actividad de los usuarios. Permiten la separación de responsabilidades y delegación de tareas. (openstack, 2023).

3.2.1.4. Proyecto.

Generalmente, cuando se habla de proyectos, se refiere a un contenedor organizativo donde se agrupan y gestionan recursos, permisos y usuarios. Un proyecto podría representar una aplicación, un departamento, un equipo o cualquier otra unidad organizativa que necesite recursos específicos en la nube.

En *OpenStack*, un proyecto es la unidad fundamental para la definición y propiedad de los recursos. Dividen lógicamente los dominios en unidades distintas más pequeñas. Tienen una estructura de árbol jerárquica, por lo que es posible crear una estructura con proyectos hijos y padres. Un proyecto tiene la capacidad de actuar como un dominio si se configura el atributo *is_domain* del proyecto en *true*. Es decir que, los dominios pueden entenderse como un tipo específico de proyectos. Sin embargo, un proyecto sólo puede actuar como dominio si su entidad padre o proyecto padre también es un dominio. (Vanecek, 2017).

3.2.2. Control y Accesos en la Nube.

3.2.2.1. Usuarios.

Los usuarios en la nube son individuos o sistemas que acceden y utilizan los recursos de la plataforma en la nube. Cada usuario tiene credenciales de autenticación, que suelen ser un nombre de usuario y una contraseña, aunque también se pueden utilizar otros métodos de autenticación, como tokens o certificados.

Los usuarios pueden ser administradores, operadores o inquilinos, dependiendo de sus responsabilidades y roles asignados.

3.2.2.2. Roles.

Los roles dentro de un ambiente de nube son conjuntos predefinidos de permisos que determinan las acciones que un usuario puede realizar en la plataforma. Algunos ejemplos comunes de roles en la nube incluyen:

Admin: Los administradores tienen control total sobre todos los recursos y pueden realizar cualquier acción en la plataforma. Tienen acceso a todas las funciones de administración de la nube.

Operador: Los operadores tienen permisos para gestionar recursos específicos, como máquinas virtuales o redes. Pueden realizar acciones relacionadas con la operación y el mantenimiento de la infraestructura.

Miembro: Los miembros son usuarios finales que pueden utilizar recursos de la plataforma, como crear máquinas virtuales o redes, pero tienen un conjunto limitado de acciones disponibles en comparación con los administradores y operadores.

Los roles determinan qué acciones pueden realizar los usuarios en la plataforma, y la asignación de roles se realiza en el contexto de proyectos, lo que permite una gestión granular de los permisos y accesos a recursos en la nube. Esto proporciona un control de seguridad y una flexibilidad significativos en la administración de la infraestructura.

3.2.2.3. RBAC.

El control de acceso basado en roles o *RBAC (Role Based Access Control)* es un paradigma de seguridad que se implementa en muchos ámbitos en tecnologías de la información. El principio básico de *RBAC* consiste en otorgar a cada usuario únicamente los permisos imprescindibles para desarrollar las funciones asociadas al puesto o rol que cumplen dentro de una organización, en el caso de la computación en la nube se utiliza para determinar el grado de acceso que tienen los usuarios dentro del sistema y las acciones que estos pueden llevar a cabo.

El modelo de control de acceso basado en roles se utiliza también en los sistemas operativos y otros tipos de *software*, como Directorio Activo (*Active Directory*) de *Microsoft* o sistemas *GNU/Linux* optimizados para tener mejor seguridad como sistemas *SELinux o Unix Solaris* (Role based access control (RBAC), 2023).

La metodología *RBAC* se basa en un conjunto de tres reglas principales que rigen el acceso a los sistemas seguros (Role-based access control - definition & overview, s.f.).

Asignación de roles: Cada transacción u operación sólo puede llevarse a cabo si el usuario ha asumido el rol apropiado. Los roles pueden ser asignados por un tercero o seleccionados por el usuario que intenta realizar la acción.

Autorización de roles: El propósito de la autorización de roles es asegurar que los usuarios sólo puedan asumir un rol para el cual se les ha dado la autorización apropiada. Cuando un usuario asume un rol, debe hacerlo con la autorización de un administrador.

Autorización de transacciones: Una operación sólo puede completarse si el usuario que intenta completar la transacción posee el rol apropiado.

3.3. Arquitectura en la Nube

El termino arquitectura en la nube hace referencia a la forma en que los diferentes componentes tecnológicos se unen y combinan para lograr la construcción y funcionamiento de una nube. Como se mencionaba antes, la arquitectura en la nube se auxilia de elementos claves como la virtualización para lograr la unión de los diferentes recursos mediante una red compartida.

De los elementos principales que incluye la arquitectura en la nube están:

Cliente: dispositivo utilizado para acceder a la nube y hacer uso y administración de ella.

Plataforma *Back-end*: esta se compone de servidores y almacenamiento.

Red: para la intercomunicación de los elementos que conforman la nube, así como la comunicación con el exterior (internet, servicios, etc.).

La combinación de estos elementos da como resultado los distintos recursos informáticos mencionados anteriormente en el tema de computación en la nube, permitiendo que usuarios finales puedan gozar de las ventajas, bondades y potencial de los recursos en la nube.

La arquitectura en la nube es una parte fundamental de la infraestructura tecnológica actual. A medida que las organizaciones buscan optimizar sus recursos y adaptarse a un entorno digital en constante evolución, han surgido varios modelos de arquitectura de nube para satisfacer diferentes necesidades y requisitos. En este contexto, se destacan cuatro conceptos esenciales: la nube pública, la nube privada, nube híbrida y nube comunitaria, cada una con sus propias características y ventajas distintivas, las cuales veremos a continuación.

3.3.1. Nube Publica.

Una nube pública ofrece servicios informáticos e infraestructura a los usuarios a través de un proveedor externo en la red pública de Internet. (Microsoft, s.f.). Sus recursos incluyen máquinas virtuales, aplicaciones y almacenamiento, entre otros.

Los servicios pueden incluir una variedad de cargas de trabajo que incluyen bases de datos, *firewalls*, equilibradores de carga, herramientas de administración y otros elementos de plataforma como servicio, o *software* como servicio. Estos pueden se gratuitos u ofrecerse a través de una variedad de esquemas de suscripción o precios bajo demanda, incluidos los modelos de pago por uso. (Bigelow., Neenan, & Casey, s.f.).

Básicamente, una nube pública se basa en un entorno virtualizado para proporcionar una extensión de la infraestructura de TI de una empresa, lo que permite a esa empresa alojar ciertos aspectos de su infraestructura y servicios en servidores virtuales que están fuera del sitio y son propiedad de un tercero. Los proveedores de servicios de nube pública tienen diferentes fortalezas y ofrecen una amplia variedad de servicios y modelos de precios. (vmware, s.f.)

3.3.2. Nube Privada.

Las nubes privadas, también conocidas como nubes internas, se caracterizan por su funcionamiento que se asemeja al de una red o centro de datos privado. En este escenario, la infraestructura de la nube está bajo el control y la gestión de una sola organización, ya sea directamente por sus propios recursos o a través de terceros, y puede ubicarse tanto dentro de las instalaciones de la organización (*on-premise*) como fuera de ellas (*off-premise*).

En el contexto de una nube privada, la organización cliente establece un entorno de virtualización en sus propios servidores, en alguno de sus centros de datos internos o en los de un proveedor de servicios especializado. Esta autonomía y control tienen un precio, ya que implica que la organización debe realizar inversiones para adquirir, construir y mantener la infraestructura de la nube. Como contrapartida, obtiene un nivel significativo de control sobre la misma, aunque esto conlleva costos más elevados en comparación con otros modelos de nube.

Las nubes privadas reducen las instancias de desperdicio de la capacidad. Permiten que la empresa configure una y otra vez los recursos de manera automática y según sea necesario, ya que no se ven limitados a las instalaciones físicas. De igual forma tienen otros beneficios como mayor capacidad de infraestructura para satisfacer grandes demandas de recursos informáticos y de almacenamiento, servicios por solicitud mediante el uso de interfaces de usuario de autoservicio y gestión basada en políticas, asignación eficiente de recursos según las necesidades del usuario, mayor supervisión de los recursos en toda la infraestructura, entre otros. (RedHat, 2023)

3.3.3. Nube Híbrida.

El termino nube híbrida hace referencia al modelo de arquitectura en la nube donde se fusionan los recursos de un entorno de nube privada con los de un entorno de nube pública. Esta opción surge en respuesta a las necesidades de empresas que ya tienen su propia infraestructura de TI, pero desean aprovechar las ventajas ofrecidas por los servicios de un proveedor externo de nube.

Las nubes híbridas brindan beneficios notables, como la capacidad de respuesta ágil y la reducción de costos, aunque esto puede implicar cierta pérdida de control sobre algunos aspectos de la infraestructura. La implementación de una nube híbrida puede ser un desafío debido a la necesidad de coordinar eficazmente recursos propios y recursos gestionados externamente, así como asegurar una conectividad robusta entre ambas plataformas.

A pesar de su complejidad, las nubes híbridas están ganando relevancia debido a su versatilidad y al conocimiento especializado que aportan algunos expertos en la integración de estos entornos. Se espera que tengan un papel importante en el panorama tecnológico del futuro, ofreciendo a las empresas la capacidad de aprovechar la infraestructura existente y los servicios de nube pública de manera conjunta para impulsar la eficiencia y la adaptabilidad en un entorno empresarial en constante evolución.

3.3.4. Nube Comunitaria.

Una nube comunitaria tiene muchas características de una nube híbrida, pero también puede tener acceso como nube publica, en la cual se construye por empresas que comparten intereses y obtienen ventajas de compartir el recurso de computación con otras (Marinos & Briscoe, 2009), o también pueden integrarse por personas que también quieran compartir poder de cómputo o simplemente archivos o información.

Este tipo de nube también tiene presente la preocupación por que muchos datos están siendo cedidos a grandes proveedores de nube publica como *Amazon, Google o Microsoft* y la posible falta de privacidad que tengan estas compañías con respecto a la información sensible que se puedan alojar en esos centros de datos (Marinos & Briscoe, 2009).

Este tipo de nube tiene sus propias ventajas frente a las demás nubes (Marinos & Briscoe, 2009), algunas de ellas son:

- Independencia de proveedores gigantes de servicios de nube.
- Autonomía de los nodos para ofrecer sus propias funciones y usar las técnicas y *software* que más le convenga.
- Identidad inherente de los usuarios en toda la nube y sus servicios.
- Escalabilidad económica mediante la optimización de recursos compartidos.

3.4. Modelos de Servicio de Nube

Para comprender el funcionamiento de la computación en la nube es necesario entender los modelos de servicio que esta ofrece ya que estos definen las funciones y responsabilidades tanto de los proveedores como de los usuarios, y constituyen la base de la computación en la nube. Existen tres tipos principales de modelos de servicios de computación en nube: infraestructura como servicio (*IaaS*), Plataforma como servicio (*PaaS*) y Software como servicio (*SaaS*).

3.4.1. Infraestructura como Servicio.

“La infraestructura como servicio (*IaaS*) es un modelo de servicio en la nube que ofrece recursos de infraestructura bajo demanda, como computación, almacenamiento, redes y virtualización, a empresas y particulares a través de la nube.

Se trata de un modelo muy atractivo en comparación con la manera tradicional de adquirir recursos de computación con los que ejecutar aplicaciones o almacenar datos, ya que esta requiere una mayor inversión de tiempo y dinero.

Las organizaciones deben comprar equipos a través de procesos de aprovisionamiento que pueden llevar meses. También deben invertir en instalaciones físicas (normalmente, salas especializadas con sistemas energéticos y de refrigeración). Además, las empresas necesitan profesionales de TI para gestionarlos y mantenerlos después de desplegar los sistemas.” (*¿Qué es IAAS (infraestructura como servicio)?* | *Google Cloud*, s. f.).

La Infraestructura como Servicio (*IaaS*) implica arrendar componentes de infraestructura de una empresa de servicios en la nube, como servidores, máquinas virtuales, redes y espacio de almacenamiento. Esto elimina la necesidad de construir y mantener físicamente una infraestructura en un centro de datos en las instalaciones, reduciendo así la complejidad y los gastos asociados.

Con *IaaS*, se puede acceder a estos recursos pagando solo por lo que se utiliza, lo que significa que se puede ajustar fácilmente la cantidad de recursos según las necesidades. En otras palabras, se puede gastar menos cuando no se necesita muchos recursos y aumentar o reducir la capacidad de manera instantánea para satisfacer la demanda en constante cambio.

3.4.2. Plataforma como Servicio

La plataforma como servicio (*PaaS*) es una capa de infraestructura en la nube que proporciona recursos para desarrollar herramientas y aplicaciones a nivel de usuario. Además, proporciona la infraestructura subyacente que incluye recursos de computación, red y almacenamiento, así como herramientas de desarrollo, sistemas de gestión de bases de datos y *middleware*.

Dado que la *PaaS* es una infraestructura basada en la nube, permite a las organizaciones evitar el coste y la complejidad de comprar y gestionar los recursos de infraestructura, incluidas las licencias de *software*, la infraestructura de aplicaciones y las herramientas de desarrollo. (Zettler, 2023).

Básicamente, el modelo *PaaS*, permite a los desarrolladores crear aplicaciones más rápido de lo que sería posible si ellos tuvieran que preocuparse de la creación, configuración y aprovisionamiento de sus propias plataformas e infraestructura de back-end, puesto que, toda la gestión del back-end tiene lugar en segundo plano. (CLOUDFLARE, s.f.)

3.5. Despliegues de Nube

Tras analizar en profundidad la computación en la nube, se abordaron sus tipos de despliegue: Nube Privada, Nube Pública, Nube Híbrida y Nube Comunitaria. También se examinaron algunos modelos de servicio, entre los cuales se incluyen Infraestructura como Servicio (IaaS) y Plataforma como Servicio (PaaS). En este apartado, presentaremos soluciones específicas para desplegar una nube y los componentes que la integran.

Las soluciones de despliegue de nube permiten a las organizaciones aprovechar la potencia de la nube para alojar, administrar y escalar sus recursos de TI de forma más eficiente. Ofrecen la infraestructura necesaria para ejecutar aplicaciones, almacenar datos y gestionar recursos de manera ágil. Estas soluciones pueden variar en cuanto a enfoque y funcionalidad: algunas se centran en la virtualización de servidores y la administración de máquinas virtuales, mientras que otras están diseñadas para gestionar contenedores y aplicaciones en entornos de nube.

Entre las plataformas para desplegar soluciones de nube, populares en el ámbito del software libre, se encuentran *OpenStack y OpenNebula*. Por otro lado, en el sector de proveedores de servicios en la nube, destacan *Amazon Web Services (AWS)*, *Google Cloud Platform y Microsoft Azure*, entre muchos otros. Cada uno de estos proveedores y plataformas ofrece un conjunto único de características y ventajas, permitiendo a las organizaciones adaptar su infraestructura de nube según sus requisitos.

A continuación, veremos un poco de las soluciones mencionadas anteriormente:

OpenStack: Es una plataforma de código abierto que permite construir y controlar infraestructuras de nube, ya sean públicas, privadas, comunitarias o híbridas. Ofrece servicios, como capacidad de cómputo, almacenamiento, *networking*, y gestión de imágenes. OpenStack es altamente adaptable y se puede ajustar según las necesidades específicas de una empresa.

OpenNebula: Similarmente, es de código abierto y se enfoca en la administración de infraestructuras de nube, con un énfasis en la virtualización de servidores y el manejo de máquinas virtuales (*VM*). *OpenNebula* destaca por su facilidad de uso y su capacidad para gestionar diversos hipervisores, como *KVM y VMware*.

***Amazon Web Services* (*AWS*):** Es la plataforma de servicios en la nube ofrecida por Amazon, y es una de las más extensas y utilizadas en el mundo. *AWS* proporciona una amplia variedad de servicios que van desde soluciones de cómputo, almacenamiento y bases de datos hasta herramientas de inteligencia artificial, Internet de las Cosas (*IoT*) y más. Debido a su gran ecosistema, *AWS* es adecuado para empresas de todos los tamaños, desde startups hasta grandes corporaciones, permitiendo la personalización y escalabilidad de recursos según las demandas.

***Google Cloud Platform* (*GCP*):** Es la propuesta de Google en el ámbito de los servicios en la nube. Similar a *AWS*, *GCP* ofrece una variedad de soluciones para cómputo, almacenamiento, análisis de datos, *machine learning* y otros campos. Su integración con otras herramientas y servicios de *Google*, como *Google Workspace* y *Google Analytics*, hace de *GCP* una opción atractiva para aquellas organizaciones ya familiarizadas con el ecosistema de *Google*.

Microsoft Azure: Es la plataforma de servicios de nube de *Microsoft*. *Azure* proporciona una gama de soluciones que abarcan desde la infraestructura como servicio (*IaaS*) hasta plataforma como servicio (*PaaS*) y Software como Servicio (*SaaS*). Azure se integra perfectamente con otros productos de *Microsoft*, como *Office 365* y *Windows Server*, y es conocido por su robustez en soluciones empresariales, así como por sus herramientas de desarrollo y gestión.

La elección de la solución adecuada no es una tarea trivial y debe ser realizada con cuidado. Depende en gran medida de las necesidades específicas de la organización. Factores como la escalabilidad, que determina cómo una solución puede adaptarse al crecimiento; la seguridad, que es crucial para proteger los datos y operaciones; y la compatibilidad con tecnologías existentes, para asegurar una integración sin problemas, son vitales al tomar una decisión. Es esencial que las organizaciones realicen un análisis exhaustivo para garantizar que la opción seleccionada se alinee con sus objetivos y requisitos.

3.5.1. Componentes de despliegue de Nube.

Las soluciones de nube mencionadas anteriormente están compuestas por una amplia gama de *software* que permite el funcionamiento de las nubes tal como las conocemos hoy en día. Estas soluciones representan solo una fracción del vasto universo de proyectos y componentes necesarios para su despliegue.

Dado el enorme número de componentes de *software* disponible, la *CNCF* (*Cloud Native Computing Foundation*) desempeña un papel crucial. Esta organización se encarga de compilar y categorizar todos los proyectos, tanto de código abierto como propietarios, que se pueden utilizar en la construcción y gestión en la nube.

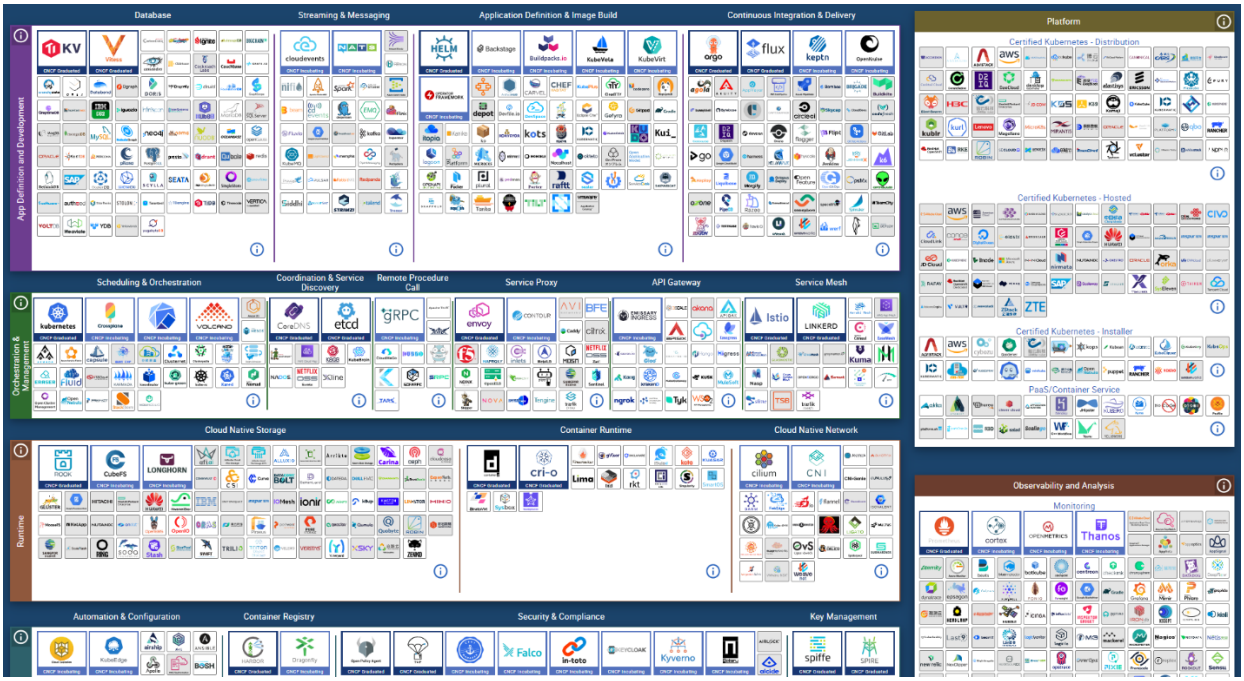


Figura 3.7. Captura del Landscape de la Nube por CNCF. Se puede ver con más detalle en la página web del CNCF (cncf.io)

Estos componentes incluyen, pero no se limitan a, servicios de identidad, cómputo y almacenamiento. A continuación, describimos estos componentes clave para cada una de las alternativas previamente mencionadas:

3.5.1.1. Identidad

***AWS**: AWS Identity and Access Management (IAM)* es una herramienta en línea que brinda una capa de seguridad para regular el acceso a los servicios de *Amazon Web Services (AWS)* de manera confiable. En esencia, *IAM* le permite gestionar de manera centralizada los derechos que determinan qué partes de *AWS* pueden ser utilizadas por usuarios específicos. Este servicio es invaluable para controlar quiénes pueden autenticarse (es decir, iniciar sesión) y quiénes tienen la autorización (permisos) para interactuar con los recursos. Además, permite establecer políticas de seguridad, definir roles y permisos detallados, y asegurar que los usuarios accedan solo a lo que les corresponde, contribuyendo así a una gestión segura y precisa de sus recursos en la nube de *AWS*.

Azure: *Azure Active Directory (Azure AD)* es un sistema basado en la nube que se encarga de gestionar la identidad y el acceso a los recursos en *Microsoft Azure* y servicios relacionados. *Azure AD* se encarga de autenticar a los usuarios, otorgarles permisos y controlar el acceso a los servicios en la nube. Además, permite la integración de aplicaciones y servicios en la nube, facilitando así la administración de identidades y accesos de forma segura y eficiente en el entorno de *Azure*.

Google Cloud: *Identity and Access Management (IAM)* es una herramienta que otorga a los administradores la capacidad de autorizar a quiénes se les permite llevar a cabo acciones en recursos específicos dentro de *GC*. Esto significa que tienes un control total y una visión completa para administrar tus activos en *GC* de manera centralizada. Proporciona una única perspectiva de las políticas de seguridad en toda la organización.

OpenNebula: *Cloud Authentication and Authorization (CAA)* desempeña un papel clave en asegurarse de que las personas adecuadas tengan acceso a los recursos correctos. Es como una especie de supervisor que se encarga de verificar quiénes son los usuarios y qué actividades pueden llevar a cabo en el entorno *OpenNebula*.

***OpenStack**: Keystone* es un servicio de identidad de código abierto utilizado por *OpenStack* que proporciona autenticación de cliente API, descubrimiento de servicios y autorización distribuida de varios inquilinos. El servicio administra bases de datos de usuario, así como catálogos de servicios de *OpenStack* y sus *endpoints* de API.

Los servicios internos se agrupan dentro de Keystone y se exponen en uno o varios *endpoints*. Los servicios internos se utilizan en combinación en el *frontend* para las acciones completadas. Por ejemplo, una llamada de autenticación validará las credenciales de usuario/proyecto con el servicio de identidad y, tras su validación correcta, creará y devolverá un token con el servicio de token. (OpenMetal, 2023)

Keystone proporciona la autenticación como se muestra en el siguiente diagrama:

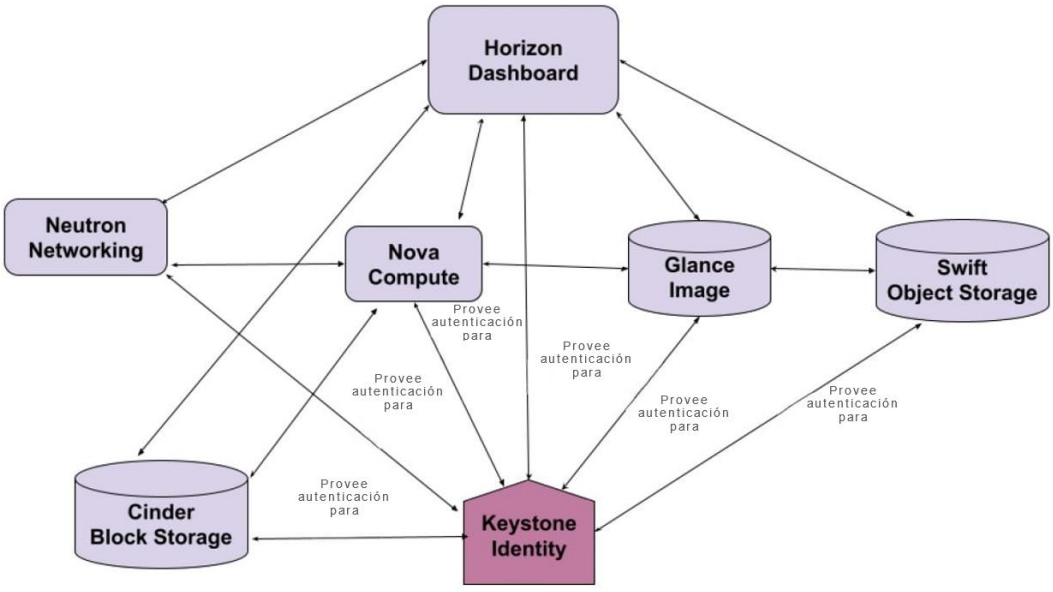


Figura 3.8. Servicio de autenticación de Keystone. Tomado de (Marrich, 2021).

3.5.1.2. Cómputo

En *cloud computing*, el componente de cómputo representa uno de los pilares fundamentales puesto que provee las capacidades de procesamiento virtualizadas bajo demanda. Estas capacidades permiten a los usuarios ejecutar y desplegar aplicaciones, servicios y procesos en servidores virtuales o contenedores, eliminando la necesidad de gestionar hardware físico y ofreciendo escalabilidad y adaptabilidad según las necesidades del momento. A continuación, se describe este componente en el contexto específicos de algunas soluciones de nube populares:

***AWS**:* El componente de cómputo en *AWS* es *Amazon Elastic Compute Cloud (Amazon EC2),* el cual es un servicio web que proporciona capacidad informática en la nube y que permite a los usuarios ejecutar aplicaciones, procesar datos y realizar tareas de cómputo sin tener que preocuparse por la infraestructura subyacente.

***Microsoft Azure**:* Ofrece *Azure Virtual Machine* como su componente de cómputo. Estas máquinas virtuales pueden ser utilizadas para desplegar soluciones Linux y Windows, proporcionando una amplia variedad de tamaños de instancia y opciones configurables para diferentes cargas de trabajo.

***Google Cloud Platform**:* Denomina a su servicio de cómputo como “Compute Engine”. Esta solución permite a los usuarios lanzar máquinas virtuales que se ejecutan en la infraestructura global y altamente segura de Google. Además, ofrece opciones de personalización y auto escalado según las demandas.

OpenNebula: Como solución de nube de código abierto, OpenNebula se centra en ofrecer soluciones simples pero flexibles para la gestión de centros de datos. Su componente de cómputo se encarga de virtualizar recursos, permitiendo a los usuarios ejecutar y administrar máquinas virtuales de manera eficiente.

***OpenStack**: Nova* es el componente del Computo en *OpenStack*. Es la base para servir infraestructura como servicio, admitiendo creación de máquinas virtuales con hipervisores como *KVM, VMWare, Hyper-v* y muchos otros, e incluso LXD.

3.5.1.3. Redes

El componente de Administración de Redes en *cloud computing* es esencial para facilitar la comunicación entre los recursos de una infraestructura en la nube. Permite que las máquinas virtuales y servicios se conecten entre sí y con la red externa, lo que es fundamental para el funcionamiento de aplicaciones y servicios en la nube. A continuación, se describe el componente de redes de los principales proveedores de servicios de nube y alternativas de código abierto para el despliegue y mantenimiento de infraestructura de nube.

AWS: El componente de red de *Amazon Web Services* se basa en servicios como *Amazon VPC* (*Virtual Private Network*) en el que los usuarios pueden crear redes virtuales aisladas, definir subredes, designar ip privadas, etc. También cuenta con componentes como *Route 53* el cual es un servicio de nombre de dominios (*DNS*) lo que permite la construcción de soluciones altamente personalizadas.

Azure: En *Microsoft Azure* el componente de red se denomina *Azure Virtual Network*, el cual permite crear redes virtuales, subredes, configurar grupos de seguridad, entre otras cosas. También puede establecer conexiones *VPN* o el utilizar *ExpressRoute* para conectar la solución de nube con recursos locales.

Google Cloud Platform: El servicio de Google también crear redes privadas virtuales, definir redes y subredes, configurar reglas de *firewall*, también puede conectarse con redes locales con *Cloud Interconnect*. Se puede combinar con *Cloud Load Balancing* para un enfoque con escalabilidad y rendimiento.

OpenNebula: La solución de código abierto *OpenNebula* proporciona funcionalidades de crear redes virtuales, conectarlas a máquinas virtuales, se pueden establecer reglas de *firewall*. La configuración de las redes de *OpenNebula* es más manual y depende de la infraestructura subyacente del usuario.

OpenStack: *Neutron* es el componente que le da a *OpenStack* el servicio de *Networking* o Redes y la administración de ésta para ser usados por las máquinas virtuales. Permite crear redes virtuales, subredes, router, direcciones *IP* flotantes y otros recursos de red para que las máquinas virtuales puedan comunicarse entre sí y con recursos de redes externas a *OpenStack* e incluso a internet.

La arquitectura estándar de *Neutron* se conforma por un nodo controlador, un nodo de red, y un conjunto de servidores de cómputo para correr las máquinas virtuales. A esto se le puede incluir un nodo para el Panel administrativo con interfaz de usuario o "*Dashboard*".

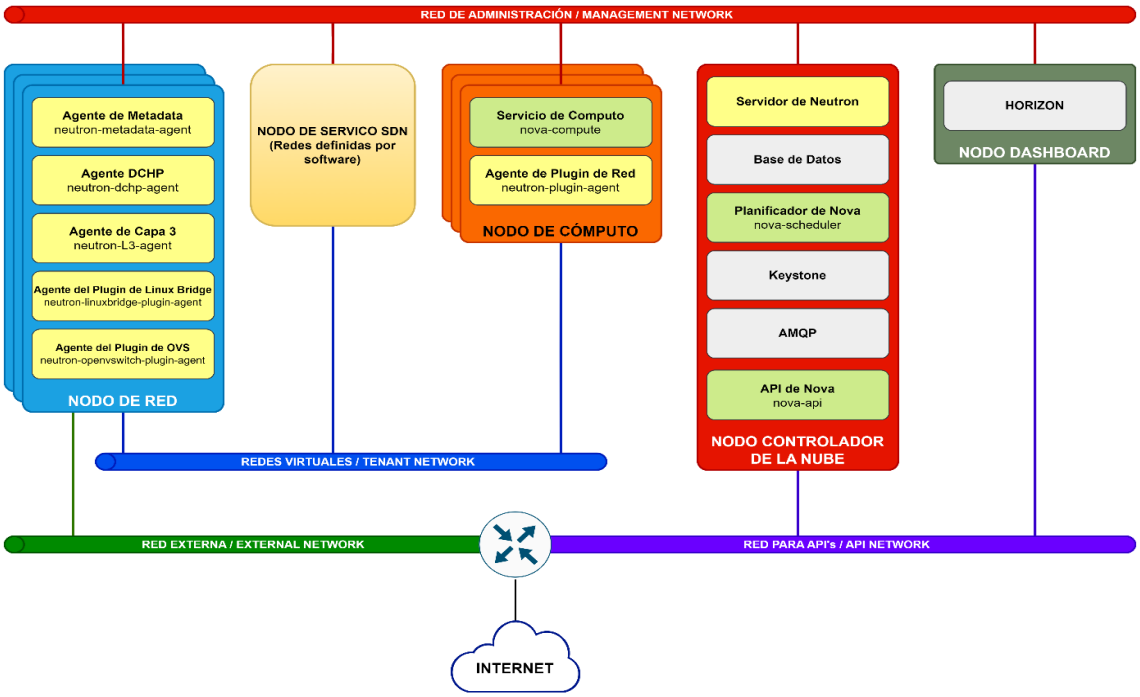


Figura 3.9. Configuración estándar de Neutron en OpenStack. Adaptado de la Documentación Oficial de OpenStack (Networking architecture, 2023) con traducción al español.

3.5.1.4. Almacenamiento en Bloque

El almacenamiento de bloque es un tipo de almacenamiento de datos en la nube que se utiliza para almacenar datos en unidades lógicas conocidas como “bloques”. Cada bloque actúa como un dispositivo de almacenamiento independiente y se asemeja a un disco duro virtual. A diferencia del almacenamiento de objetos, que almacena datos junto con metadatos en objetos individuales, el almacenamiento de bloque almacena datos en bloques discretos con una estructura fija y se accede a ellos a través de una dirección única. Los bloques pueden ser utilizados para crear sistemas de archivos, formatear discos virtuales o proporcionar almacenamiento de datos de nivel inferior para máquinas virtuales y otros servicios.

Los principales proveedores de *cloud computing* cuentan con sus propios componentes de almacenamiento en bloque:

AWS: La solución de almacenamiento de bloque de *Amazon Web Services* es *EBS* (*Elastick Block Storage*) el cual proporciona discos virtuales que pueden ser adjuntado a máquinas virtuales. Estos volúmenes *EBS* son de alta disponibilidad y alto rendimiento, vienen en tipos *SSD* y *HDD* para satisfacer todas las necesidades.

Azure: El componente de almacenamiento de bloque de *Microsoft Azure* se llama *Azure Disk*, los cuales también se pueden adjuntar a las máquinas virtuales. Tienen mucho parecido a los *EBS* de *AWS*, pero setos vienen en discos *Premium SSD*, *Standard SSD* y *Estándar HDD* con opciones de rendimiento y precios variados.

Google Cloud: La solución de la empresa de *Google* para el almacenamiento de bloque se denomina *Persistent Disk*, la cual es muy parecida a la versión de *AWS* o *AzGure*, ya que la idea es la misma, adjuntar volúmenes de disco a las máquinas virtuales y vienen en categorías *Standard, SSD* y *Local SSD*, para ofrecer alto rendimiento y disponibilidad.

OpenNebula: El módulo que permite a *OpenNebula* tener funciones de almacenamiento de bloque se denomina *DataStore* los cuales proporcionan acceso a bloques individuales a las máquinas virtuales como discos adicionales. Los tipos de almacenamiento dependen de la infraestructura de los usuarios.

OpenStack: El servicio de *OpenStack* para el almacenamiento en bloque es *Cinder,* el cual consiste en proveer almacenamiento persistente a una máquina virtual. Es un componente fundamental dentro del ecosistema de *OpenStack* que proporciona servicios de almacenamiento en bloques para entornos de nube.

Su función principal es gestionar el almacenamiento de bloques, que se utiliza principalmente para discos virtuales o volúmenes en máquinas virtuales. Este servicio proporciona una capa de abstracción sobre el almacenamiento subyacente, lo que significa que los usuarios pueden crear volúmenes sin preocuparse por la infraestructura de almacenamiento especifica que se utiliza detrás de escena como por ejemplo *Ceph* o *Gluster*.

Los usuarios pueden crear instantáneas de volúmenes, lo que permite capturar un estado específico de un volumen en un momento dado. Esto es útil para respaldar datos o para crear copias de seguridad de volúmenes de forma eficiente.

3.5.1.5. Almacenamiento de objetos

El almacenamiento de objetos en la nube es una solución diseñada para guardar datos en forma de objetos, a diferencia de los tradicionales sistemas de archivos o bloques. Cada objeto en este tipo de almacenamiento contiene los datos, un identificador único y metadatos. Esta estructura es especialmente útil para grandes cantidades de datos no estructurados, como imágenes, vídeos y backups.

Diferentes proveedores de servicios en la nube han implementado sus propias soluciones de almacenamiento de objetos, cada una con características y ventajas particulares:

AWS: Cuenta con *Amazon S3* el cual es altamente escalable, duradero y confiable. Proporciona una estructura basada en *buckets* para almacenar objetos, con capacidades avanzadas como versionado, eventos Lambda y diferentes clases de almacenamiento según la frecuencia de acceso y la durabilidad requerida.

***Microsoft Azure**:* Presenta *Blob Storage* para el almacenamiento de objetos. Está diseñado para almacenar grandes cantidades de datos no estructurados y ofrece diferentes niveles de acceso*, como acceso frecuente, acceso poco frecuente y almacenamiento en frío.*

Google Cloud Platform (GCP): En *GCP*, el servicio de cómputo se conoce como *Cloud Storage*. Este servicio proporciona almacenamiento de objetos altamente duradero y escalable, con opciones para diferentes clases de almacenamiento basadas en la frecuencia de acceso y el tiempo de almacenamiento, como *Standard*, *Nearline*, *Coldline* y *Archive*.

***OpenNebula**:* Utiliza una solución integrada que facilita la gestión de imágenes y volúmenes en el almacenamiento. Aunque no es un sistema de almacenamiento de objetos puro como las otras soluciones, puede integrarse con sistemas externos para aprovechar estas capacidades.

***OpenStack**: Ofrece Swift* como su solución de almacenamiento de objetos. *Swift* es altamente disponible, distribuido, y eventualmente consistente, diseñado para escalar horizontalmente y ofrecer redundancia.

3.5.1.6. Interfaz de administración y control

***AWS**: AWS Management Console* es una aplicación web, se trata de una plataforma en línea que integra y menciona una amplia variedad de consolas de servicios destinadas a la gestión, administración y control de los recursos que *Amazon Web Services (AWS)* ofrece en cada servicio.

***Microsoft Azure**: Azure Portal* es una interfaz web unificada que ofrece una alternativa a las utilidades de línea de comandos. Con esta herramienta, puede controlar y gestionar de manera efectiva los servicios que *Azure* provee. Puede reunir, administrar y supervisar una amplia gama de recursos, desde simples aplicaciones web hasta despliegues en la nube más complejos.

***Google Cloud**: Google Cloud Console* es una plataforma centralizada para gestionar de manera eficiente sus servicios en *Google Cloud*. Ofrece un conjunto de herramientas que le permiten llevar a cabo diversas acciones, como la configuración de dominios adquiridos a través de la plataforma. Además, esta consola proporciona un entorno unificado que facilita la supervisión, el control y la administración de todos sus recursos en *Google Cloud*.

***OpenNebula**: Sunstone* es la interfaz de usuario web que permite a los administradores y usuarios gestionar y controlar recursos y máquinas virtuales en un entorno OpenNebula. A través de *Sunstone*, puede realizar tareas como creación, gestión y monitorización de máquinas virtuales, la administración de redes, el despliegue de aplicaciones y otras operaciones relacionadas con la gestión de infraestructuras.

***OpenStack**: Horizon* es la implementación estándar del panel de control de *OpenStack*, y sirve como una interfaz web que permite a los usuarios interactuar con los servicios de *OpenStack*, como *Nova* para máquinas virtuales, *Swift* para almacenamiento de objetos, *Keystone* para autenticación y otros servicios. Esencialmente*, Horizon* es la puerta de entrada amigable y basada en navegador que facilita a los usuarios la gestión de recursos y servicios en su nube *OpenStack* sin necesidad de conocimientos técnicos profundos.

3.6. Patrón de Identidad Federada

El patrón de identidad federada es un patrón de diseño de nube que permite a los usuarios autenticarse con múltiples aplicaciones y servicios utilizando un único conjunto de credenciales. Se usa para delegar la responsabilidad de autenticación a un proveedor de identidad externo.

La identidad federada está relacionada con el inicio de sesión único (SSO), en el que el ticket de autenticación único de un usuario, o token, es de confianza en múltiples sistemas de TI o incluso organizaciones. Esto puede simplificar el desarrollo, minimizar el requisito de administración de usuarios y mejorar la experiencia del usuario de la aplicación. (Microsoft, s.f.).

Para explicar cómo funciona el patrón de identidad federada, tomemos un ejemplo en el que un usuario necesita acceder a una aplicación que requiere autenticación.

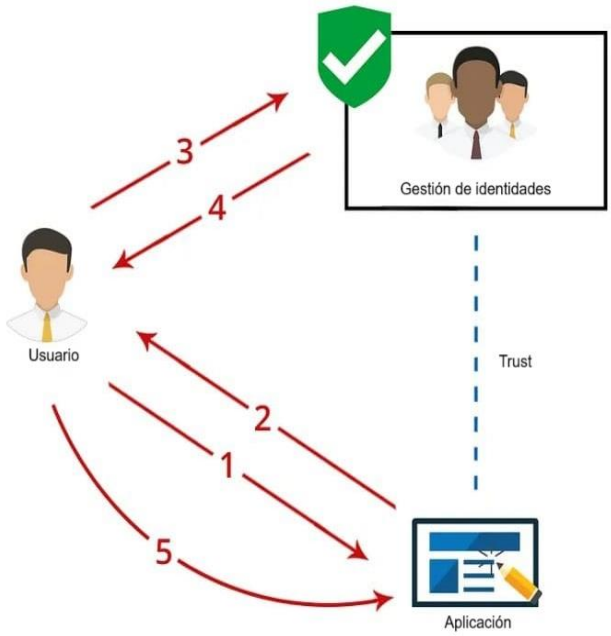


Figura 3.10. Patrón de identidad federada. Tomada de (Kodagoda, 2017) con traducción al español.

El flujo de autenticación es así:

1- El usuario navega a la aplicación segura.
2- Las aplicaciones indican al usuario no autenticado que se autentique en el proveedor de identidades (IdP).
3- El usuario se autentica con el proveedor de identidad (por ejemplo, ingresando el nombre de usuario y la contraseña).
4- El proveedor de identidad proporciona un token de acceso al usuario (se pueden incluir reclamaciones).
5- El usuario vuelve a la aplicación con el token de acceso.
6- La aplicación permite el acceso al contenido de esta de acuerdo con la autorización de ese usuario.

Es muy importante entender que el sistema de gestión de identidades (y el proveedor de identidades también) NO proporciona autorización para el usuario, solo proporciona autenticación. Es responsabilidad de las aplicaciones implementar el mecanismo de autorización utilizando los detalles proporcionados por la administración de identidades o el proveedor de identidades. (Kodagoda, 2017).

3.6.1. Aprovisionamiento de Usuarios

El aprovisionamiento de usuarios es el proceso de creación, actualización y eliminación de cuentas de usuario en múltiples aplicaciones y sistemas. Se trata de un proceso de gestión de identidades y accesos (IAM) que garantiza que las cuentas de usuario se crean, reciben los permisos adecuados, se modifican, se desactivan y se eliminan.

El aprovisionamiento de usuarios se activa cuando se agrega o actualiza nueva información en la base de datos del sistema original y, posteriormente, se administra a lo largo del ciclo de vida del usuario dentro de la organización. El acceso a las aplicaciones y los datos se concede en función de las necesidades de la empresa para ese usuario y se ajusta a medida que los roles y las necesidades empresariales evolucionan y cambian. (What is user provisioning?, 2023).

3.7. Clúster

El termino clúster es un extranjerismo adaptado del inglés el cual significa “grupo” o “agrupación”, y en tecnologías de información se usa para denominar a un grupo de dos o más computadoras o servidores los cuales, ejecutando en paralelo un servicio, buscan cumplir un objetivo común (Nordhoff, 2020).

Estos servidores agrupados actúan como si fueran uno solo, lo cual es transparente para los usuarios u otros servicios que usan esta solución. A cada uno de los servidores que componen el clúster se les suele llamar “nodo”.

El propósito de configurar servidores como un clúster es gestionar grandes cantidades de acciones sobre los datos que manejan, ya sea procesamiento de estos datos, lectura o escritura en grandes cantidades o distribuir la carga que se hacen al servicio que están brindando. Los nodos se conectan mediante una red de alta velocidad para compartir los datos y sincronizarse gracias a una capa llamada “*middleware*” el cual es un software intermedio que es responsable de mostrar el clúster como un único sistema unificado (Malaiya, 2012).

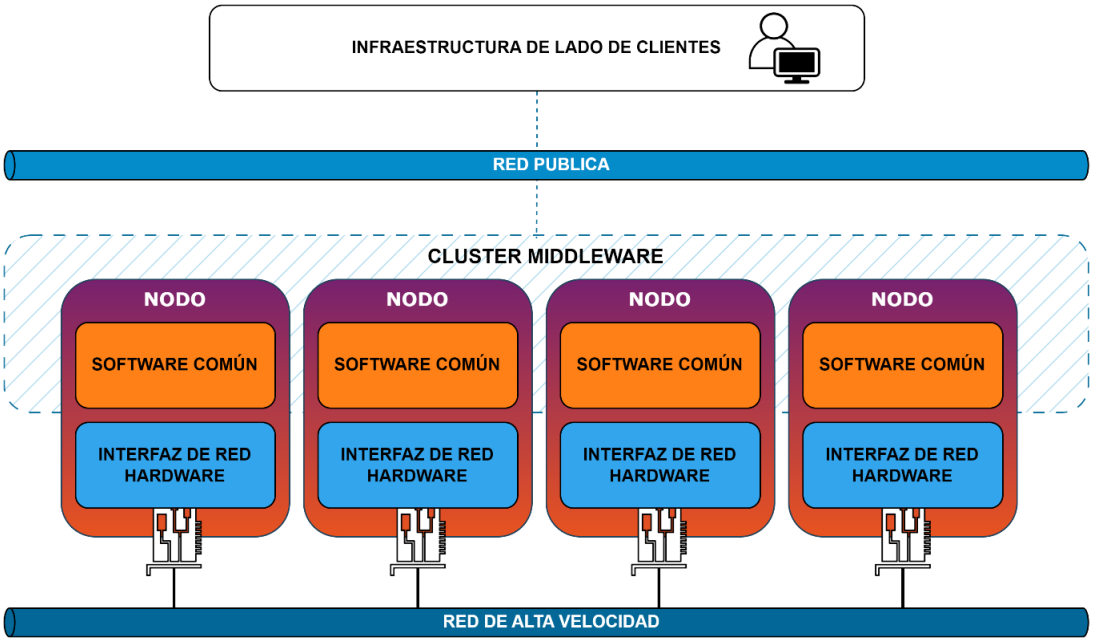


Figura 3.11. Arquitectura típica de un Clúster. Adaptado de (Malaiya, 2012) con traducción al español

3.7.1. Tipos de Clúster.

Clúster de Alta Disponibilidad: Son clúster que son destinados a misiones críticas en los que se requiere que el servicio tenga estado de inactividad mínimo. Se caracterizan por cambiar automáticamente cuando un nodo falla y puede ser eliminado del clúster o reemplazado.

Clúster de Balanceo de Carga: Son clúster donde lo que se persigue es distribuir el tráfico que llega en todos los nodos para optimizar el rendimiento y evitar que se concentre en un nodo la carga de las peticiones.

Clúster de Alto Rendimiento: Son clúster diseñados para dar prestaciones en cuanto a grandes capacidades de cálculo. Estos clústeres tienen como componente más importante, los núcleos del procesamiento.

4 Marco de investigación

4.1. Pregunta de Investigación

¿Es factible utilizar *OpenStack* para construir una infraestructura de nube comunitaria y geo distribuida, aprovechando los recursos internos de cada ubicación, y habilitando la interconexión entre diversas instancias de nube?

4.2. Objetivo General

Evaluar si es factible la construcción de un prototipo de una infraestructura de nube comunitaria basada en *OpenStack* que admita un número ilimitado de regiones con servicios de autenticación federada, persistencia de sesión, base de datos centralizada e interfaz de gestión.

4.3. Objetivos Específicos

Diseñar una arquitectura escalable que se adapte a la adición de nuevas regiones con facilidad y sin comprometer el rendimiento general del sistema.

Implementar un servicio de identificación que garantice la autenticación y autorización de usuarios federados a través de todas las regiones, garantizando la seguridad y el acceso uniforme.

Implementar un mecanismo de persistencia de sesiones que asegure la continuidad de las actividades de los usuarios en las regiones y servicios conectados.

Diseñar y crear una base de datos centralizada que respalde las operaciones y datos de la Nube Comunitaria.

Proveer una interfaz gráfica para la gestión de las múltiples regiones integradas.

Evaluar el prototipo desarrollado para garantizar el correcto funcionamiento, escalabilidad y adaptabilidad a las diferentes demandas y cargas de trabajo.

5 Metodología de la Investigación

5.1. Enfoque de la Investigación

Existe una amplia literatura que proporciona directrices sobre cómo realizar correctamente una investigación, ya sea teórica, práctica o una combinación de ambas. En este trabajo se toma como referencia al autor Piergiorgio Corbetta, quien ha explorado extensamente las metodologías de investigación, destacando la importante interacción entre el objeto o fenómeno estudiado, ya sea físico o social, y el estudio en sí.

Según Corbetta, las técnicas de investigación pueden ser únicamente cualitativas o cuantitativas, dependiendo de las variables que surjan entre el objeto de estudio y el investigador. Dado nuestro caso de estudio y el objetivo final del proyecto, hemos explorado ambos tipos de metodologías, asegurándonos de comprender las diferencias entre ambos enfoques.

5.1.1. Enfoque cualitativo

Inspirada en el paradigma interpretativo, la investigación cualitativa mantiene una relación abierta e interactiva entre teoría e investigación. El investigador cualitativo suele evitar la formulación de teorías previas al trabajo de campo, creyendo que esto podría inhibir su capacidad para comprender la perspectiva del sujeto estudiado y limitar su visión de a priori. La elaboración de la teoría y la investigación empírica ocurren, por lo tanto, de manera simultánea. En este enfoque, se le da menos importancia a la reflexión sobre la literatura existente. (Corbetta, 2007, pág. 41).

La investigación cualitativa busca comprender los fenómenos sociales y humanos desde la perspectiva de los participantes, en lugar de imponer teorías preconcebidas. Los investigadores cualitativos desarrollan teorías a medida que recopilan datos y se sumergen en el campo, priorizando la inmersión en el contexto, la observación participante y la interacción directa con los sujetos de estudio. Esto contrasta con el enfoque cuantitativo, que se basa en una revisión de literatura más estructurada.

5.1.2. Enfoque cuantitativo

La investigación cuantitativa, inspirada por el paradigma neopositivista, sigue una secuencia lógica y estructurada de fases, adoptando un planteamiento deductivo donde la teoría precede a la observación, orientada a la verificación empírica de las teorías previamente formuladas. En este contexto, el análisis sistemático de la literatura existente es crucial, ya que es de este de donde se derivan las hipótesis. (Corbetta, 2007, pág. 41)

El enfoque cuantitativo se apoya en una estructura lógica basada en teorías, utilizando la revisión sistemática de la literatura para formular hipótesis, recopilar datos cuantitativos y emplear análisis estadísticos para verificar o refutar dichas hipótesis. Esta metodología busca proporcionar evidencia empírica sólida para comprender y explicar fenómenos de investigación en diversos campos del conocimiento.

5.2. Diseño de la Investigación

La investigación científica emerge como respuesta a las necesidades cotidianas, requiriendo la definición de un método que permita su realización de manera adecuada, eficiente y eficaz, buscando obtener resultados óptimos para la interpretación de los fenómenos en estudio.

Dado nuestro caso de estudio y los objetivos planteados en este trabajo, se decidió utilizar tanto el enfoque cualitativo como el cuantitativo para desarrollar la investigación.

La implementación de ambos enfoques facilitó la recolección de la información necesaria sobre la implementación y desarrollo de una infraestructura de nube comunitaria multi-región, alineando la investigación con el problema y los objetivos planteados.

El enfoque cualitativo se adoptó para abordar los aspectos contextuales del problema, con el objetivo de entender las necesidades y expectativas de las empresas del sector de cloud computing. También se buscó comprender la percepción de las oportunidades y desafíos asociados con la implementación de una nube comunitaria multi-región. Este enfoque proporcionó una visión más profunda de los aspectos subjetivos, las percepciones y las interacciones humanas vinculadas al problema.

El enfoque cuantitativo se aplicó para abordar los desafíos tecnológicos y empresariales asociados a la computación en la nube, permitiendo la recopilación de datos concretos y el análisis estadístico. A su vez, proporcionó una base sólida para la toma de decisiones informadas.

5.2.1. Población y Muestro

Dada la naturaleza dual de nuestro enfoque metodológico, que comprende tanto aspectos cualitativos como cuantitativos, y considerando las particularidades de nuestro caso de estudio, resultó imperativo seleccionar una muestra representativa que nos permitiera obtener resultados confiables y aplicables, por lo que optamos por Next-Latam, una empresa con vasta experiencia en el ámbito de la computación en la nube y la provisión de servicios de infraestructura cloud. De esta empresa, seleccionamos al CEO para realizar una entrevista en profundidad, buscando obtener una perspectiva experta y detallada sobre la implementación y desarrollo de infraestructuras de nube comunitaria multi-región.

Por otro lado, para abordar el aspecto cuantitativo de la investigación, nos enfocamos en individuos con conocimientos y experiencia práctica en *cloud computing*, específicamente aquellos que desempeñan funciones de TI en roles como administradores de centros de cómputo, administradores de infraestructura *cloud* y desarrolladores de aplicaciones especializados en despliegues en nubes públicas. Para este segmento, establecimos una muestra de 20 personas, seleccionadas a través de un muestreo aleatorio simple. Este grupo nos ayudará a evaluar la viabilidad de la implementación y desarrollo del prototipo de infraestructura de nube comunitaria multi-región desde una perspectiva más amplia y cuantitativa.

5.2.2. Instrumentos y Técnicas de recopilación de datos.

Dado que esta investigación tiene aspectos cualitativos y cuantitativos, es fundamental destacar la necesidad de herramientas adecuadas para la recolección de datos acordes a cada enfoque. En este sentido, se han evaluado y seleccionado los siguientes instrumentos para este propósito.

5.2.2.1. Entrevista semiestructurada.

Con el objetivo de obtener datos basados en la experiencia acumulada en el tema de *cloud computing* y en el área de Tecnologías de la Información (TI), se ha decidido implementar la técnica de la entrevista semiestructurada para la recopilación de datos en este estudio. Esta se llevará a cabo mediante una plataforma digital de videoconferencias.

La entrevista semiestructurada, cuyos detalles se pueden consultar en el Anexo 1 fue dirigida al *CEO* de *Next-Latam*. Se le ha elegido por su profundo conocimiento en la temática relacionada con la investigación, su extensa experiencia en el área de TI, y el éxito alcanzado con Next-Latam y su expansión territorial. El propósito de esta entrevista es explorar aspectos fundamentales relativos a la provisión de servicios y el uso de la computación en la nube en el ámbito de TI.

5.2.2.2. Encuesta.

Se diseñaron encuestas, cuyos detalles se encuentran en el Anexo 2, dirigidas a profesionales del área de TI con conocimientos en computación en la nube.

El propósito principal de estas encuestas es recolectar datos e información cuantitativa sobre temas específicos tales como infraestructura en la nube, tipos de nube, seguridad en la nube, *Openstack*, entre otros.

Esto permitirá comprender las opiniones y necesidades de los profesionales en TI respecto al tema en investigación, contribuyendo así al desarrollo de un prototipo de alta calidad capaz de generar un impacto positivo en las organizaciones y empresas donde se implemente.

5.2.3. Resultados

5.2.3.1. Análisis de datos cualitativos.

El análisis cualitativo se ha realizado con base en el enfoque hermenéutico de Hans-Georg Gadamer.

El análisis hermenéutico, según Hans-Georg Gadamer, es una práctica interpretativa que busca comprender textos, eventos y expresiones artísticas dentro de un marco de referencia histórico y cultural. Este enfoque se centra en la interacción dinámica entre el intérprete y el objeto de estudio, subrayando que el entendimiento va más allá de descifrar una intención original; es un diálogo vivo que se enriquece con cada encuentro interpretativo. Gadamer sostiene que la comprensión es siempre una fusión de horizontes, que incluye tanto las perspectivas históricas del texto o del fenómeno como las del intérprete. Esta fusión se refiere a la manera en que nuestras comprensiones previas y el nuevo entendimiento emergen y se informan mutuamente en el acto de la interpretación (VLADUTESCU, 2018).

La “fusión de horizontes” es un proceso que reconoce que cualquier interpretación se construye a partir de un entendimiento básico o prejuicio previo. Gadamer argumenta que este proceso de interpretación es esencial en la búsqueda del significado de la palabra escrita, y es en esta interacción donde el texto cobra vida para el intérprete, un proceso que es más significativo que desenterrar la “verdad” que el autor pudo haber intentado transmitir (Regan, 2012).

Gadamer articula también el "círculo hermenéutico" para ilustrar como los intérpretes se mueven entre la comprensión previa y las nuevas perspectivas que emergen durante la interpretación, destacando la importancia de estar conscientes de nuestras comprensiones previa y como estas informan nuestra comprensión previa continua. De esta manera, el análisis hermenéutico se convierte en un proceso reflexivo que desafía la búsqueda de un significado fijo y enfatiza la necesidad de un entendimiento que se adapta y cambia con el tiempo (Regan, 2012).

Así, el análisis hermenéutico de Gadamer se presenta no solo como una técnica de interpretación sino como una filosofía que reconoce y valora nuestras comprensiones previas como esenciales para el dialogo significativo con el texto y con los otros. Con ello se evidencia que el significado es siempre una construcción compartida y evolutiva.

Por lo que, aplicar el enfoque hermenéutico de Gadamer al análisis de la entrevista con el CEO de Next-Latam, Ing. Alexis Rojas, implica un proceso interpretativo que considera el contexto histórico, las comprensiones previas tanto del entrevistador como del entrevistado y el texto mismo.

He aquí un análisis hermenéutico detallado:

1. **Comprensión previa y conciencia histórica efectiva**
    - La entrevista fue abordada con una comprensión previa del impacto significativo de la computación en la nube en la infraestructura contemporánea. Esta comprensión previa informó la línea de interrogatorio y fue esencial para situar las respuestas del CEO dentro de un marco de transformación digital emergente. Asimismo, se reconoció la manera en que la experiencia del CEO y su visión estratégica para la empresa dan forma a sus respuestas. Esto muestra la interconexión de sus conocimientos y su recorrido personal con los objetivos y los planes a largo plazo de Next-Latam.
2. **Contenido de la entrevista**
    - Rol y beneficios de la nube: La descripción de Ing. Alexis Rojas sobre los beneficios de la nube refleja una actitud progresista hacia la adopción de la nube, equilibrando el escepticismo tradicional con las tendencias tecnológicas actuales.
    - Uso de soluciones en la nube: La asociación de Next-Latam con Bolt Cloud y la decisión en contra de utilizar proveedores de nube convencionales como AWS, Azure o Google Cloud pueden indicar una estrategia adaptada a necesidades empresariales o regionales específicas.
    - Decisiones de infraestructura: El cambio de servidores físicos a infraestructura en la nube sugiere un movimiento estratégico influenciado por consideraciones de coste y escalabilidad pertinentes para el crecimiento de Next-Latam.

3. **Círculo hermenéutico**

- Al entrelazar las partes específicas de la entrevista con el entendimiento general del papel de la computación en la nube, el análisis revela una comprensión iterativa y más profunda de la estrategia corporativa de Next-Latam.
- La percepción de Ing. Alexis Rojas sobre las barreras culturales en El Salvador frente a la sofisticación del mercado en Panamá sugiere que los factores no técnicos desempeñan un papel sustancial. De igual forma, matiza los desafíos de seguridad, respecto a que el servicio de nube adecuado puede ofrecer una seguridad superior a la de los servidores locales tradicionales. Finalmente, la idea de que el coste es un factor fundamental en la adopción de los servicios en nube se afina con el ejemplo concreto de la transición de Next-Latam, que abandona los servidores físicos por cuestiones de sostenibilidad. Esto refleja una tendencia más amplia de la industria en la que las consideraciones de coste son críticas, pero se evalúan frente a otras ventajas como la escalabilidad y el rendimiento.

4. **Fusión de horizontes**

- La integración del conocimiento histórico del CEO con las percepciones actuales sobre la computación en la nube proporciona una comprensión más rica de los retos y oportunidades para Next-Latam en el contexto actual. Esta fusión de horizontes se evidencia en el reconocimiento de las barreras culturales y la sofisticación del mercado en diferentes regiones.

5. **Aplicación y compromiso dialógico**
    - Las conclusiones del análisis sugieren cómo las prácticas de Next-Latam pueden reflejar y responder patrones más amplios de adopción de la tecnología y desafíos de seguridad en la industria de software como servicio. El análisis invita a un diálogo continuado sobre la estrategia organizacional y la evolución del mercado tecnológico.
6. **Creación de significado en la aplicación**
    - Las respuestas del CEO sobre las ventajas prácticas de la computación en la nube son interpretadas como un reflejo de la eficiencia operativa y ventaja competitiva. La transición de Next-Latam al adoptar la infraestructura basada en la nube es examinada como un ejemplo concreto del cambio tecnológico en acción.
7. **Reflexión sobre los prejuicios**
    - La discusión sobre las percepciones culturales en El Salvador y Panamá desafía las suposiciones previas y enriquece el análisis con una dimensión sociocultural que es crucial para comprender la adopción y la implementación de nuevas tecnologías.
8. **El papel del lenguaje**
    - El uso de terminología específica por parte del CEO proporciona una ventana a su conocimiento estratégico y operativo, resaltando la importancia de un lenguaje preciso para la comprensión integral del rol de la computación en la nube en la empresa.

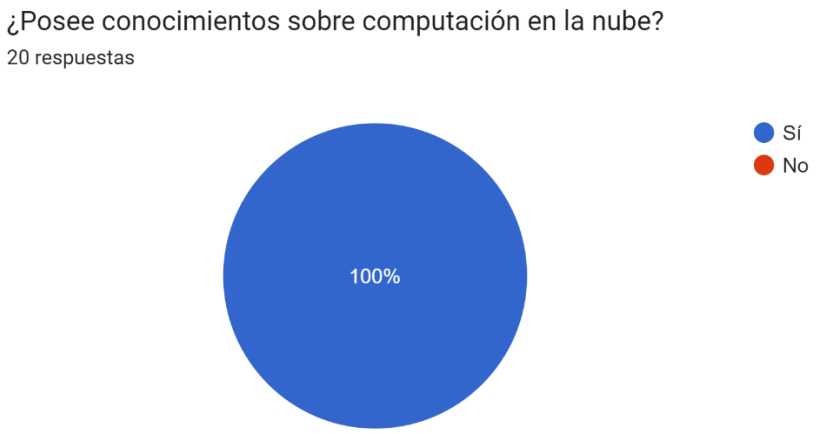
Conclusiones del análisis hermenéutico

- Las decisiones estratégicas de Next-Latam de adoptar e integrar servicios en la nube reflejan una alineación con las tendencias tecnológicas globales y las demandas del mercado regional. La entrevista con el CEO de Next-Latam, refuerza el valor práctico y estratégico de la computación en nube para escalar las operaciones y permitir un modelo de software como servicio para la empresa.
- El análisis puso de relieve las barreras culturales a la adopción de tecnología en mercados específicos, como El Salvador, que contrastan con la sofisticación del mercado panameño. Esto subraya la necesidad de un enfoque de mercado diferenciado y la importancia de la adaptación cultural a medida que las empresas del sector tecnológico se expanden por diversas regiones.
- Los beneficios económicos y las consideraciones de seguridad son fundamentales en la transición de la empresa de los servidores físicos tradicionales a una infraestructura basada en la nube. Las ideas del CEO sobre las ventajas financieras y operativas de los servicios en la nube, como el ahorro de costes, la escalabilidad y las medidas de seguridad preconfiguradas, sugieren que los factores económicos y de seguridad son impulsores cruciales para la adopción de la nube.
- La visión del CEO sobre el futuro de la computación en nube coincide con la noción de que adoptar la tecnología en nube no es opcional, sino un imperativo estratégico en la era de la transformación digital. Esto refleja la comprensión más amplia de que la tecnología en la nube es una fuerza transformadora que las empresas deben aprovechar para seguir siendo competitivas e innovadoras.

5.2.3.2. Análisis de datos cuantitativos

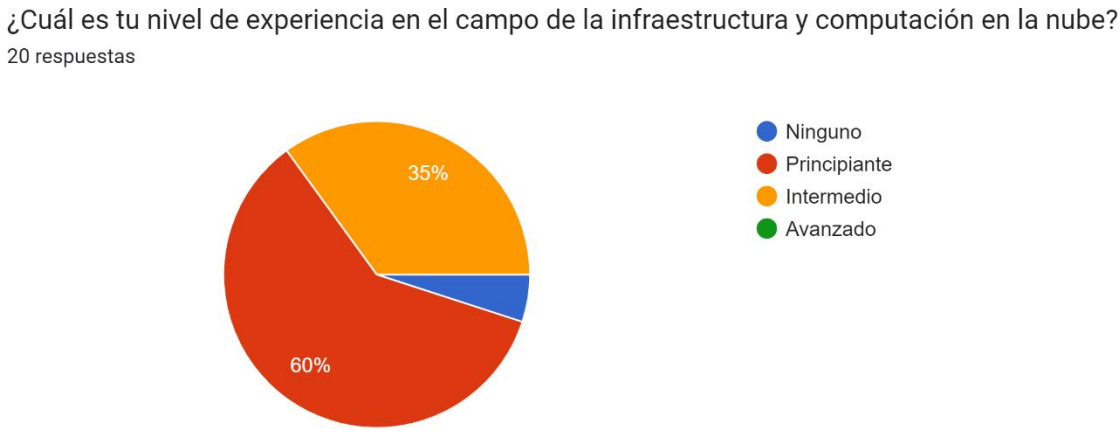
A continuación, se presentan los resultados de la encuesta (ver Anexo 2) realizada en el mes de octubre de 2023, dirigida a 20 profesionales del área TI que laboran desempeñando funciones con el desarrollo tecnológico y a su vez con temas de infraestructura y computación en la nube.

Pregunta 1



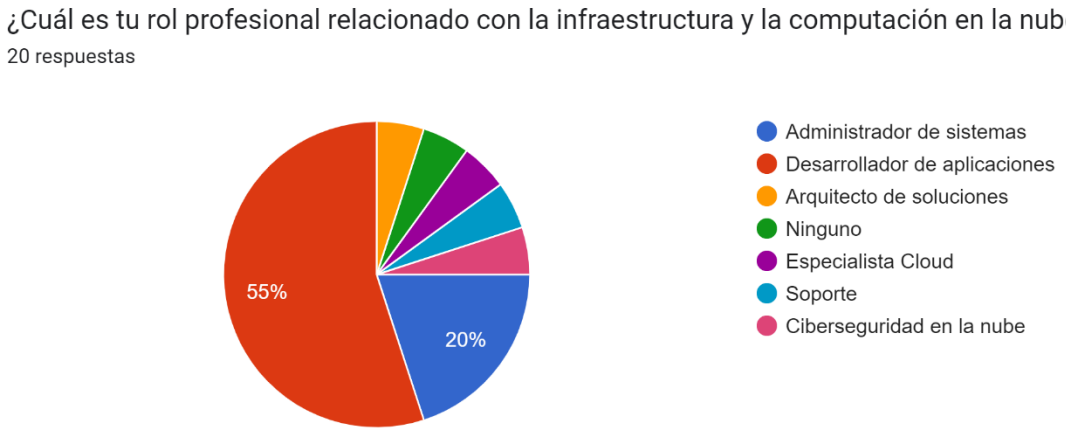
El 100% de los encuestados afirmó poseer conocimientos sobre computación en la nube. Esto sugiere que la muestra de encuestados está bien informada y familiarizada con este tema específico de la tecnología.

Pregunta 2



El 60% de los encuestados se considera principiante en el campo de la infraestructura y computación en la nube. El 35% de los encuestados se considera intermedio y el 5% de los encuestados indicó no tener experiencia en este campo.

Pregunta 3

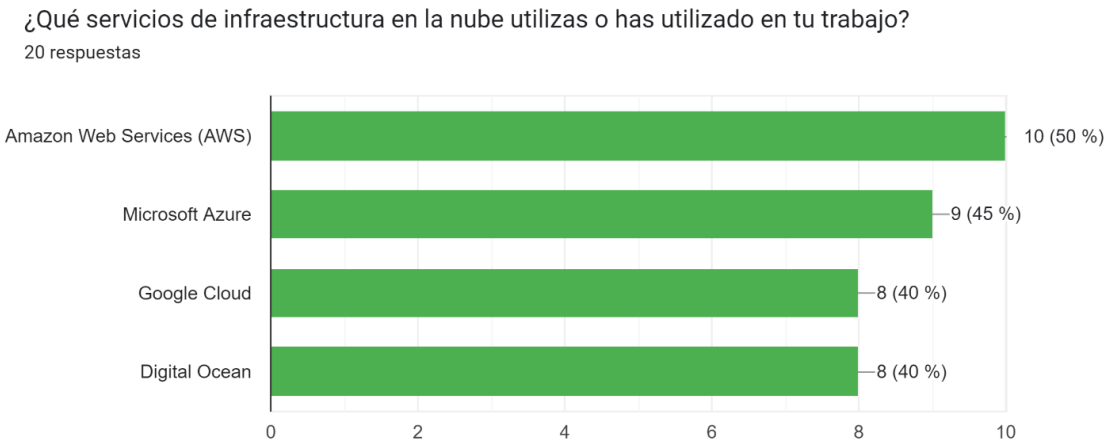


El 55% de los encuestados tiene un rol como desarrollador de aplicaciones en el ámbito de la infraestructura y la computación en la nube. El 20% de los encuestados desempeña el rol de administrador de sistemas.

El 5% de los encuestados se identificó como arquitecto de soluciones en este contexto. Otro 5% indicó no tener ninguno de los roles proporcionados. El 5% trabaja como especialista en la nube.

Otro 5% está en el área de soporte técnico. El 5% restante se especializa en ciberseguridad en la nube. Esto sugiere una diversidad de roles profesionales relacionados con la infraestructura y la computación en la nube entre los encuestados, con una mayoría que se desempeña como desarrolladores de aplicaciones. También es relevante notar que existe una variedad de roles especializados en este campo.

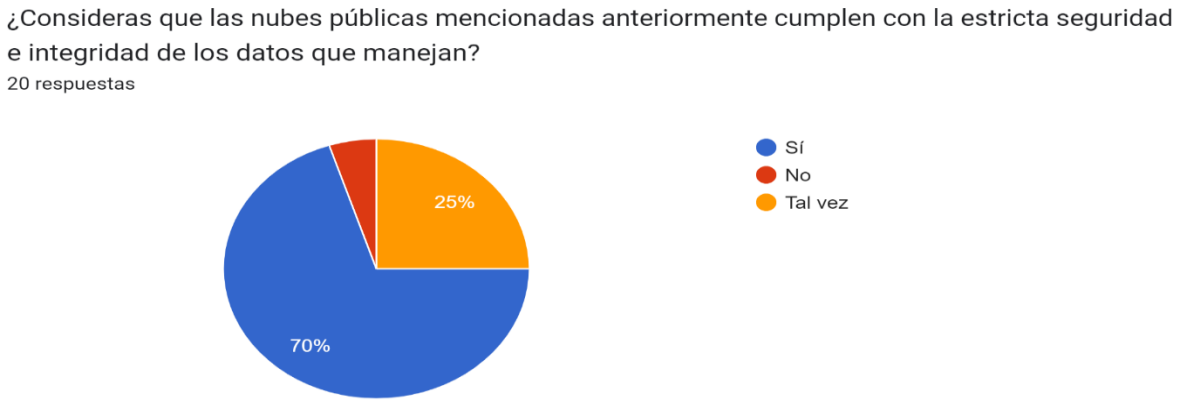
Pregunta 4



El 50% de los encuestados utiliza o ha utilizado servicios de *AWS (Amazon Web Services)* en su trabajo relacionado con la infraestructura en la nube. El 45% de los encuestados utiliza o ha utilizado servicios de *Microsoft Azure*. El 40% de los encuestados utiliza o ha utilizado servicios de *Google Cloud*. Otro 40% utiliza o ha utilizado servicios de *Digital Ocean*.

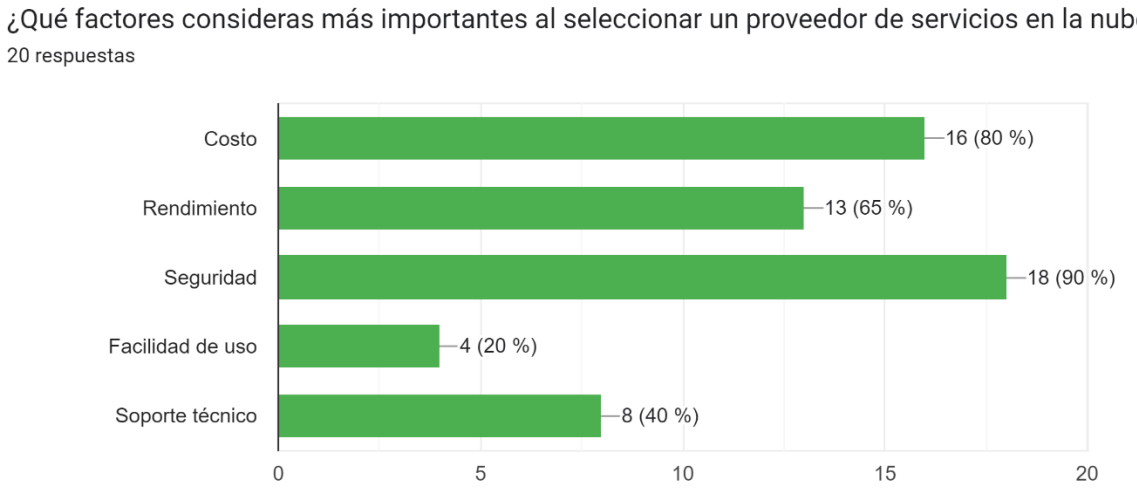
Esto indica una distribución diversa en el uso de servicios de infraestructura en la nube, con una presencia significativa de *AWS*, seguido de cerca por *Microsoft Azure* y *Google Cloud. Digital Ocean* también es utilizado por un porcentaje considerable de encuestados en sus trabajos relacionados con la infraestructura en la nube.

Pregunta 5



El 70% de los encuestados considera que las nubes públicas mencionadas cumplen con la estricta seguridad e integridad de los datos. El 25% de los encuestados respondió “tal vez,” lo que indica cierta incertidumbre o dudas en cuanto a la seguridad e integridad de los datos en estas plataformas. El 5% de los encuestados no considera que estas nubes públicas cumplan con los estándares de seguridad e integridad de los datos. La mayoría de los encuestados confían en la seguridad e integridad de los datos en las nubes públicas mencionadas, pero hay un grupo minoritario que tiene dudas o preocupaciones al respecto.

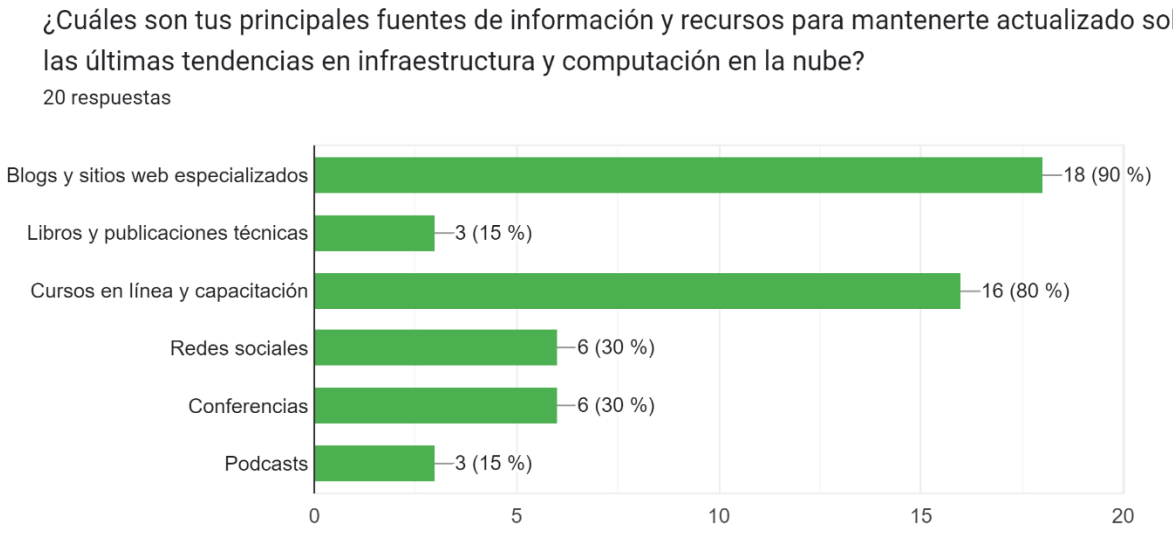
Pregunta 6



El 90% de los encuestados considera que la seguridad es el factor más importante al seleccionar un proveedor de servicios en la nube. El 80% de los encuestados considera el costo como un factor esencial en la selección de un proveedor de servicios en la nube, lo que sugiere que la economía y la eficiencia financiera son consideraciones significativas. El 65% de los encuestados valora el rendimiento de los servicios de la nube como un factor importante. El 40% de los encuestados considera el soporte técnico relevante en su elección de proveedor de servicios en la nube.

Solo el 20% de los encuestados menciona la facilidad de uso como un factor importante, lo que indica que, aunque relevante, es menos prioritario que otros factores. Esto destaca que la seguridad es una principal preocupación al elegir un proveedor de servicios en la nube, seguida de cerca por el costo y el rendimiento.

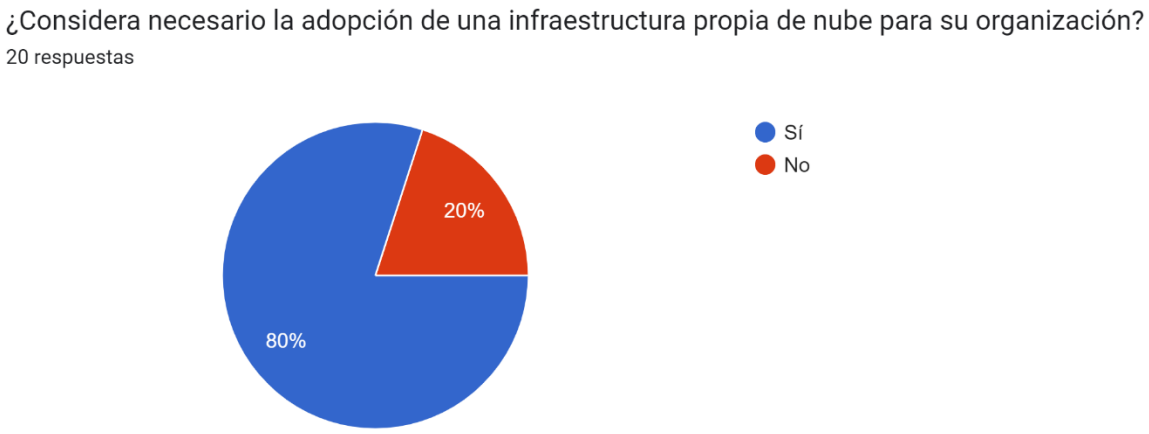
Pregunta 7



El 90% de los encuestados utiliza blogs y sitios web especializados como su principal fuente de información para mantenerse actualizado sobre las últimas tendencias en infraestructura y computación en la nube. El 80% de los encuestados recurre a cursos en línea y capacitación para mantenerse actualizados en este campo, lo que indica que la formación en línea es altamente valorada. El 30% de los encuestados utiliza redes sociales y conferencias como fuentes de información, aunque en menor medida que los blogs y cursos en línea. El 15% de los encuestados menciona libros y publicaciones técnicas como sus fuentes principales de información.

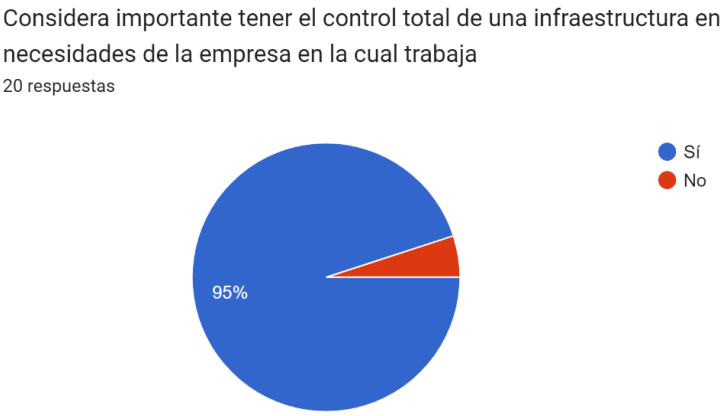
Otro 15% escucha podcasts para mantenerse actualizado. La mayoría de los encuestados confían en recursos en línea, como blogs y cursos, para mantenerse al tanto de las últimas tendencias en infraestructura y computación en la nube. Las redes sociales y las conferencias también son utilizadas por un segmento importante.

Pregunta 8



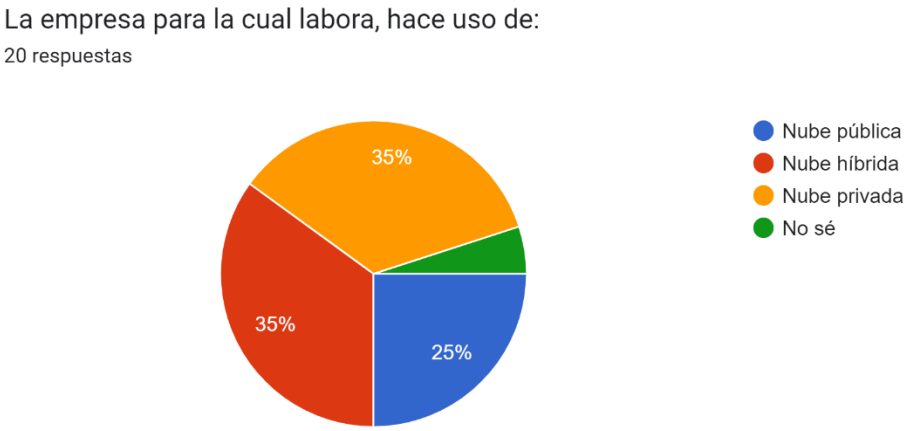
El 80% de los encuestados considera necesario adoptar una infraestructura propia de nube para su organización. El 20% de los encuestados no considera necesaria esta adopción. Esto sugiere que la gran mayoría de los encuestados ven la adopción de una infraestructura propia de nube como una necesidad para sus organizaciones, lo que refleja la importancia creciente de la computación en la nube en el entorno empresarial.

Pregunta 9



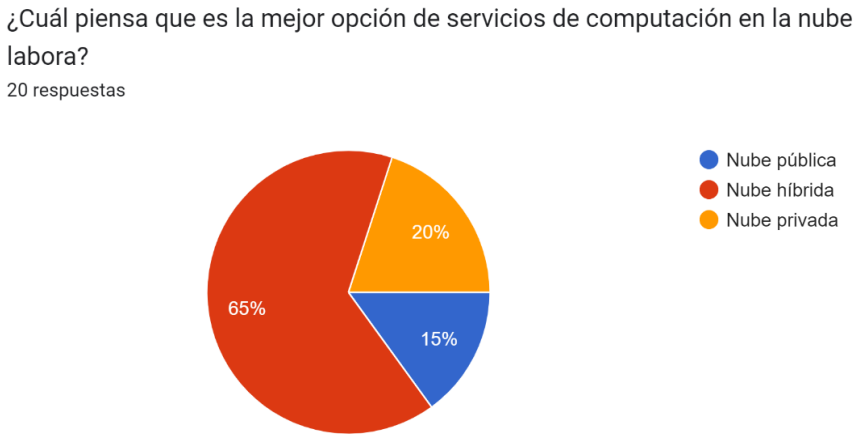
El 95% de los encuestados considera importante tener el control total de una infraestructura en la nube en cuanto a las necesidades de la empresa en la cual trabaja. Esto indica una fuerte preferencia por el control y la personalización de la infraestructura en la nube para satisfacer las necesidades específicas de la empresa. El 5% de los encuestados no considera importante tener el control total de la infraestructura en la nube para satisfacer las necesidades de la empresa en la cual trabaja. Esto resalta la alta prioridad que se da al control y la adaptabilidad de la infraestructura en la nube para alinearla con las necesidades empresariales.

Pregunta 10



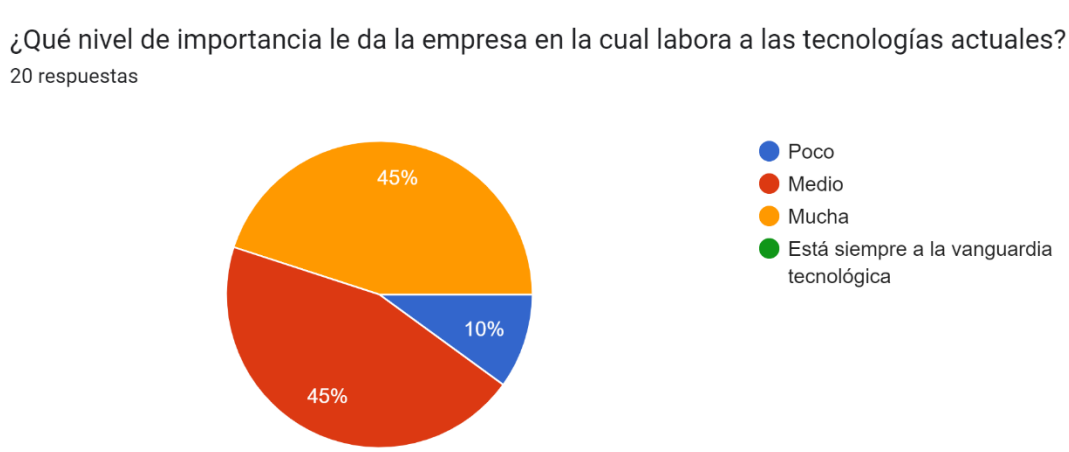
El 35% de los encuestados menciona que la empresa para la cual trabajan hace uso de una infraestructura de nube híbrida, lo que implica la combinación de servicios de nube pública y privada. Otro 35% de los encuestados señala que la empresa utiliza una infraestructura de nube privada, lo que implica que la infraestructura de nube se mantiene en una red privada y controlada por la empresa. El 25% de los encuestados indica que la empresa utiliza una infraestructura de nube pública, que se proporciona y se gestiona por proveedores externos. Un 5% de los encuestados no está seguro o no sabe qué tipo de infraestructura en la nube utiliza su empresa.

Pregunta 11



El 65% de los encuestados considera que la mejor opción para la empresa en la que trabajan es una infraestructura de nube híbrida, que combina aspectos de la nube pública y privada. El 20% de los encuestados prefiere una infraestructura de nube privada para su empresa. El 15% de los encuestados opta por una infraestructura de nube pública como la mejor opción. Esto indica que la mayoría de los encuestados cree que una nube híbrida es la mejor opción para su empresa, seguida de cerca por la nube privada. La elección depende de las necesidades específicas de la empresa y los factores individuales.

Pregunta 12



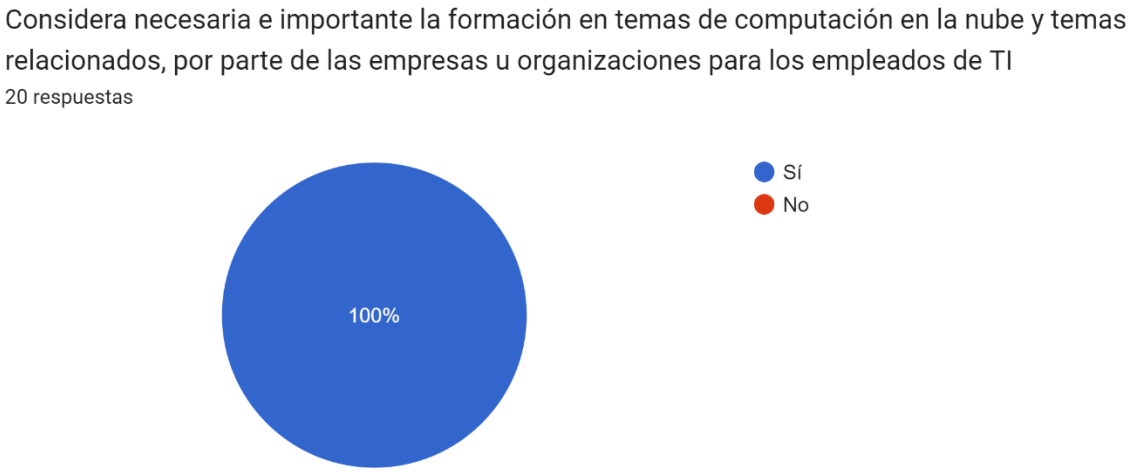
El 45% de los encuestados considera que la empresa para la cual laboran les da una importancia media a las tecnologías actuales. Otro 45% de los encuestados cree que su empresa da una alta importancia a las tecnologías actuales. El 10% de los encuestados indica que la empresa les da poca importancia a las tecnologías actuales.

Esto muestra que una proporción significativa de encuestados percibe que sus empresas valoran mucho las tecnologías actuales, mientras que otro grupo igualmente grande considera que la importancia es de nivel medio. Solo un pequeño porcentaje cree que la empresa da poca importancia a las tecnologías actuales.

Pregunta 13

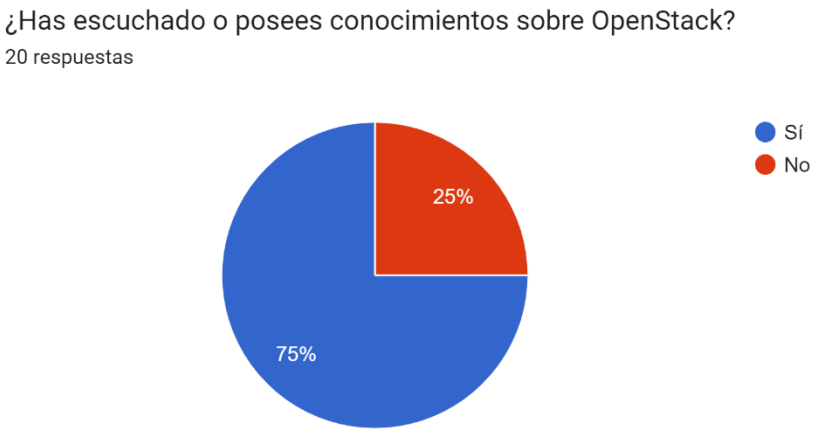
Según los porcentajes proporcionados por los encuestados, el 80% de ellos considera que la nube ya es predominante en la actualidad, mientras que el 20% cree que será predominante en el futuro. La computación en la nube ya ha alcanzado un papel predominante en la tecnología y la industria actual. La nube se ha convertido en una parte integral de la infraestructura tecnológica en muchas organizaciones y seguirá desempeñando un papel fundamental en el futuro.

Pregunta 14



El 100% de los encuestados está de acuerdo en que es necesaria e importante que las empresas u organizaciones proporcionen formación en temas de computación en la nube y temas relacionados para sus empleados de TI. Este resultado indica un consenso total entre los encuestados en la importancia de la formación en la nube y tecnologías relacionadas para los profesionales de tecnología de la información.

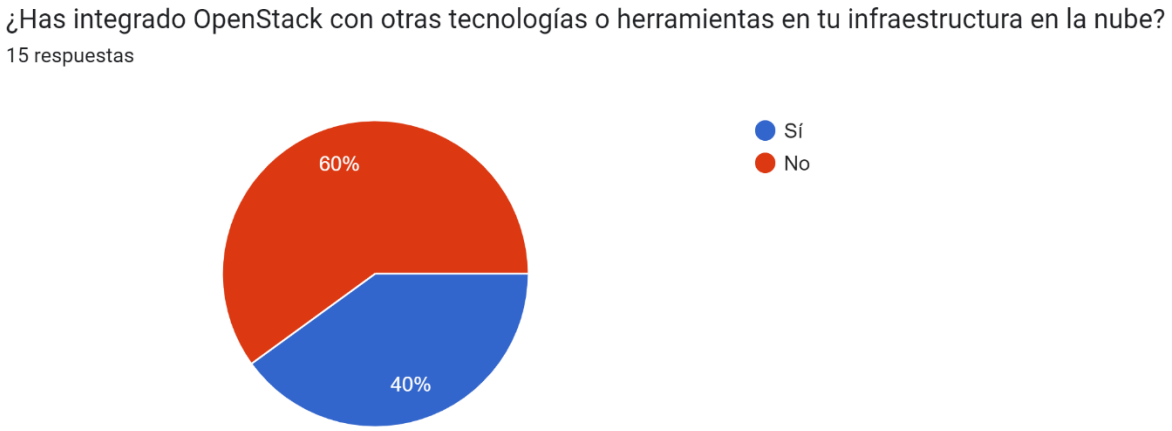
Pregunta 15



El 75% de los encuestados ha escuchado o posee conocimientos sobre OpenStack. El 25% de los encuestados no ha escuchado ni posee conocimientos sobre OpenStack. La mayoría de los encuestados está familiarizada con *OpenStack*, una plataforma de código abierto para la gestión de la infraestructura en la nube.

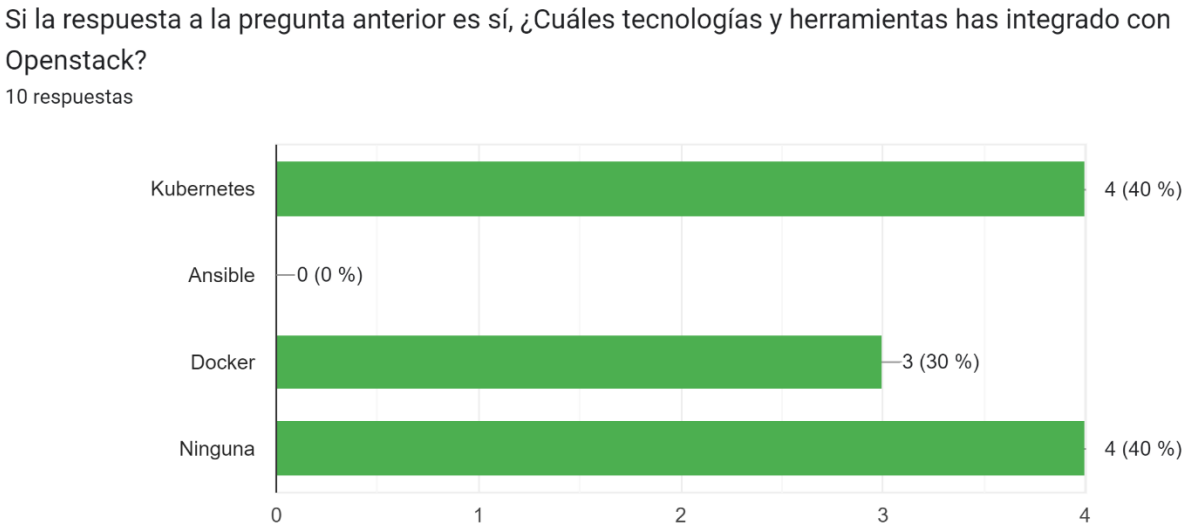
Las siguientes preguntas son presentadas únicamente a los que respondieron sí en la pregunta número 15.

Pregunta 16



El 60% de los encuestados ha integrado *OpenStack* con otras tecnologías o herramientas en su infraestructura en la nube. El 40% de los encuestados no ha integrado *OpenStack* con otras tecnologías en su infraestructura en la nube.

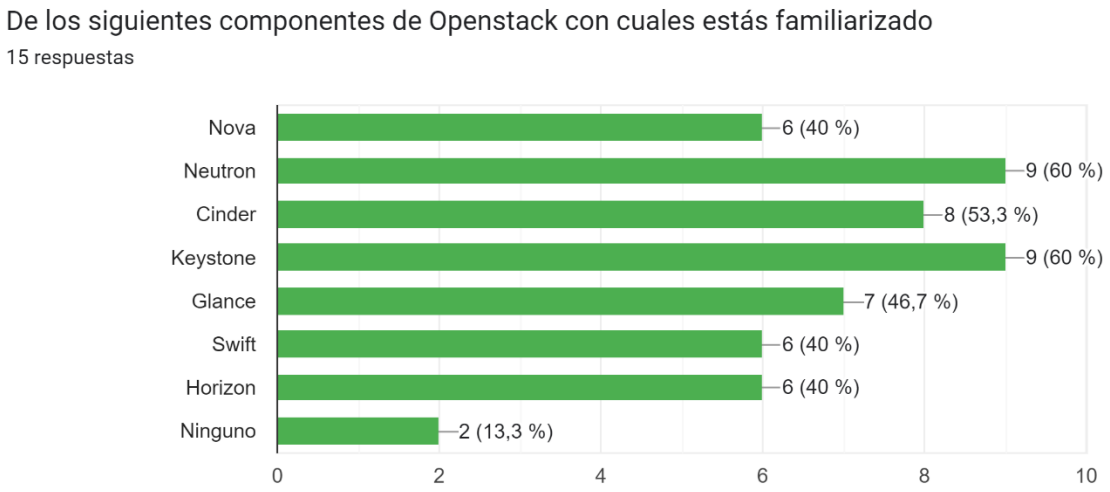
Pregunta 17



El 40% de los encuestados ha integrado Kubernetes con *OpenStack*. El 30% de los encuestados ha integrado Docker con *OpenStack*. El 40% de los encuestados no ha integrado ninguna de las tecnologías mencionadas (*Kubernetes o Docker*) con *OpenStack*.

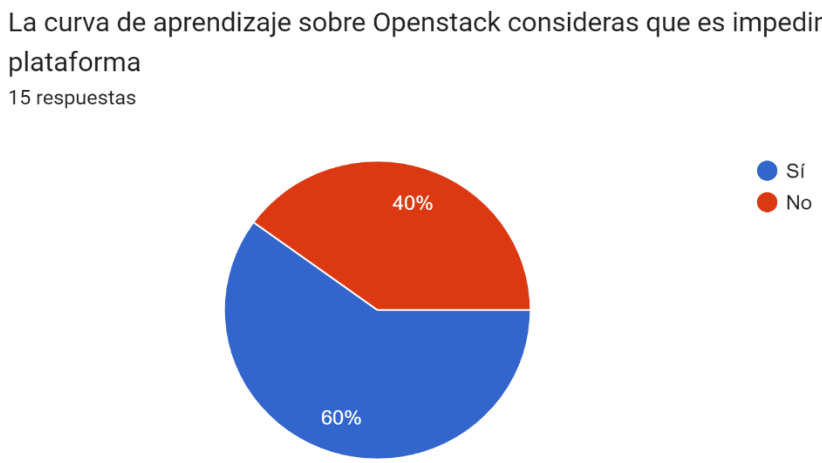
Esto muestra que una proporción significativa de los encuestados ha realizado integraciones tanto con *Kubernetes* como con *Docker* en sus entornos *OpenStack*, pero también hay un grupo que no ha integrado ninguna de estas tecnologías específicas con *OpenStack*.

Pregunta 18



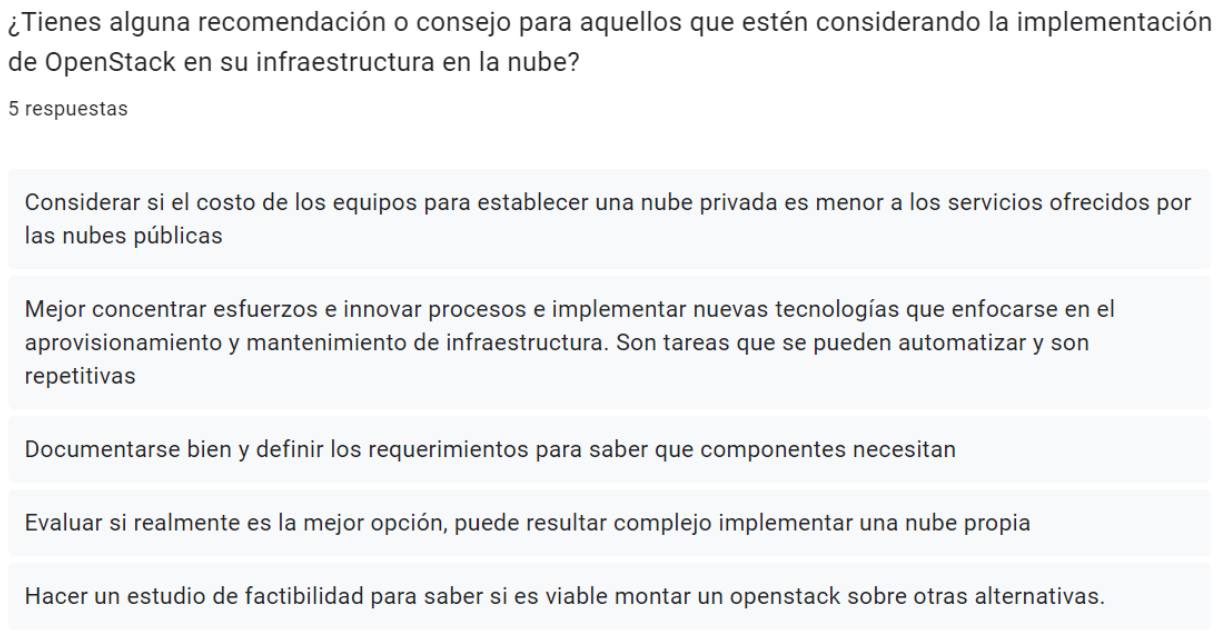
El 60% de los encuestados está familiarizado con los componentes de OpenStack, como *Neutron y Keystone*. El 53.3% de los encuestados está familiarizado con *Cinder*. El 46.7% de los encuestados está familiarizado con *Glance*. El 40% de los encuestados está familiarizado con *Nova, Swift y Horizon*. El 13.3% de los encuestados no está familiarizado con ninguno de los componentes mencionados. Esto refleja una variedad en los niveles de familiaridad con los diferentes componentes de *OpenStack*, con *Neutron y Keystone* siendo los más ampliamente conocidos.

Pregunta 19



Según los porcentajes proporcionados, el 60% de los encuestados considera que la curva de aprendizaje en *OpenStack* es un impedimento para la utilización de esta plataforma, mientras que el 40% no lo ve como un impedimento. Esto sugiere que una mayoría de los encuestados percibe la curva de aprendizaje en *OpenStack* como un desafío o barrera para su adopción, mientras que un grupo significativo no lo considera un impedimento importante.

Pregunta 20



Con base en los resultados de la pregunta 20, para una mayor comprensión de los profesionales de TI con experiencia en el área de computación en la nube y entender los patrones que emergen, clasificamos las respuestas en los siguientes temas: viabilidad económica, complejidad técnica, automatización de procesos, formación en el área y evaluación de alternativas.

Viabilidad Económica:

Se destaca la importancia de evaluar si los costos de establecer una nube privada con OpenStack son menores que los servicios ofrecidos por las nubes públicas. La viabilidad económica es una preocupación central para los profesionales encuestados.

Complejidad Técnica:

Se subraya la necesidad de una cuidadosa configuración e implementación para garantizar resultados óptimos. La complejidad técnica es reconocida como un desafío, pero fundamental para el éxito.

Automatización de Procesos:

Recomendación de concentrarse en innovar procesos y utilizar la automatización para tareas repetitivas. La eficiencia a través de la automatización es un enfoque sugerido para liberar recursos.

Formación y Documentación:

Se aconseja documentarse bien y definir claramente los requisitos. La formación y la documentación son esenciales para la implementación exitosa.

Evaluación de Alternativas:

Se destaca la complejidad de implementar una nube propia y se sugiere un estudio de factibilidad. La evaluación cuidadosa de OpenStack frente a otras alternativas es recomendada.

La viabilidad económica y la complejidad técnica son consideraciones clave para los profesionales al implementar OpenStack. La automatización, la formación y la evaluación de alternativas también se destacan como aspectos importantes para el éxito en este proceso.

5.2.4. Conclusión de los resultados y de la investigación

Los resultados cualitativos y cuantitativos de la presente investigación indican que la computación y la infraestructura en la nube, incluida la utilización de *OpenStack*, se ha convertido en un componente esencial en la estrategia tecnológica actual y futura de muchas organizaciones, ofreciendo ventajas en términos de eficiencia, costos y escalabilidad. La decisión de adopción de la nube depende de factores específicos de cada empresa, como las necesidades, los recursos y la evaluación de costos. Además, la formación en tecnologías de nube se considera esencial para los profesionales de TI.

6 Prototipo

6.1. Requerimientos

Como se describe en el contexto de este trabajo, se ha creado una infraestructura de nube comunitaria basada en *OpenStack* cuya característica principal es la admisión de múltiples regiones. Siendo esta (infraestructura) el nodo central para el alojamiento de las múltiples regiones, las cuales consumirán los servicios que se provean.

Los principales requerimientos para el desarrollo de este prototipo basado en OpenStack fueron:

- Brindar el servicio de identidad a todas las regiones que forman parte de la nube comunitaria para la autenticación y autorización, gestionándolo a través de un servicio de federación de usuarios.
- Un servicio de federación de usuarios para tener un solo repositorio de usuarios que sirva para todas las regiones.
- Proveer almacenamiento de caché de sesiones dentro de la plataforma, y el servicio de cola de mensajes para coordinar las operaciones e información de estado entre los servicios.
- Proporcionar un servicio de base de datos centralizado.
- Propiciar una interfaz para la gestión y administración de la infraestructura y las regiones alojadas.

En otras palabras, servir como una región maestra que brinda los servicios mediante API a las regiones.

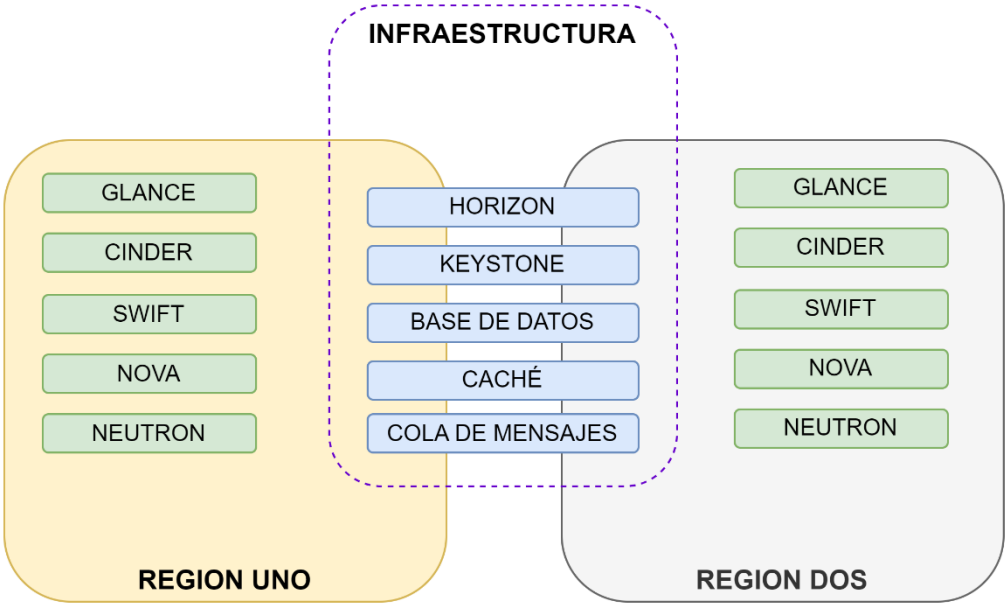


Figura 6.1 Conceptualización de la Infraestructura de la Nube Comunitaria

La Infraestructura de los servicios funciona como una “Región 0” donde no se provee servicios típicos de *OpenStack,* sino que sirve a las Regiones que sí proveen servicios como computo, almacenamiento en bloque, almacenamiento de objetos, entre otro.

Se profundizará sobre los recursos utilizados para satisfacer estos requerimientos en el apartado 6.3 de este mismo capitulo.

6.2. Arquitectura

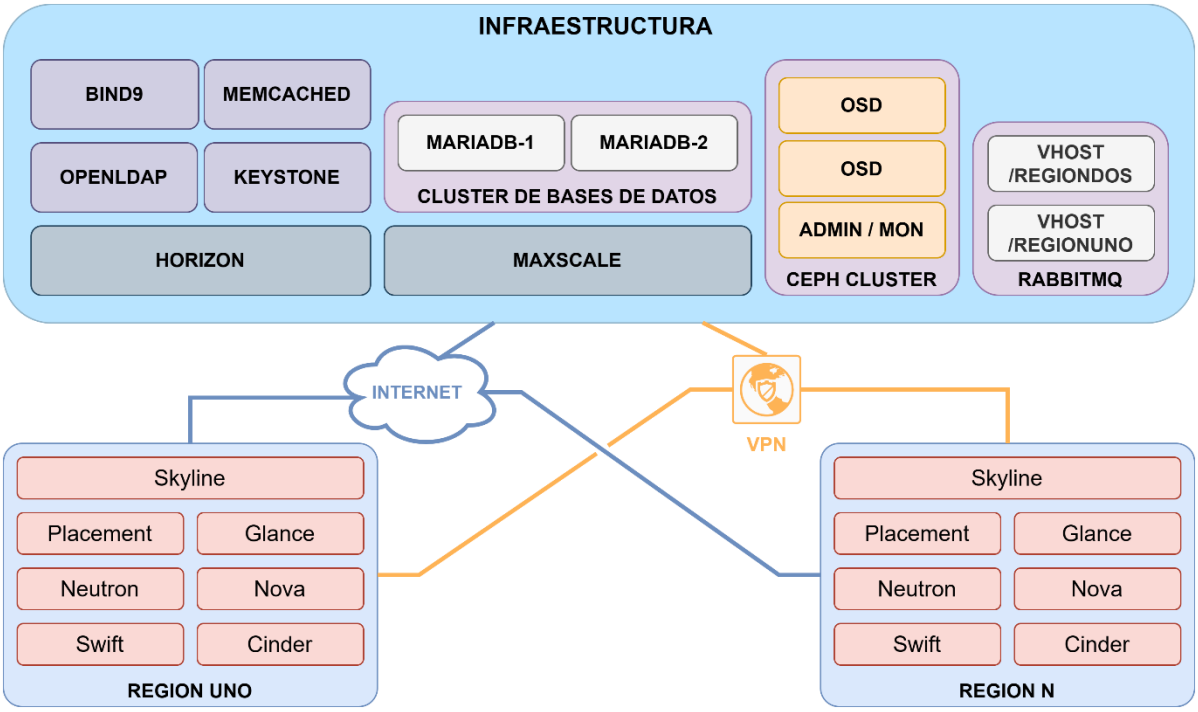


Figura 6.2 Arquitectura de la Infraestructura que brinda servicios a las regiones.

La región infraestructura brinda los servicios necesarios para las regiones, a través de distintas *API's*.

Servicio de caché: La infraestructura ofrece servicios de cache mediante *Memcached*, permitiendo a las regiones de la nube comunitaria almacenar y recuperar datos de manera eficiente, mejorando el servicio de las aplicaciones.

Servicio de cola de mensaje: Además proporciona mediante *RabbitMQ* el servicio de cola de mensajes lo que facilita la comunicación asíncrona entre los diversos componentes permitiendo el procesamiento de tareas en segundo plano y gestión de eventos de forma eficiente.

Para evitar posibles retrasos en los mensajes y garantizar la separación de las comunicaciones de las regiones se ha configurado un *vhost* dedicado para cada región garantizando una cola de mensajes independiente.

Base de datos centralizado: Se cuenta con un clúster de *MariaDB* que sirve de base de datos centralizado para toda la nube comunitaria, permitiendo disponibilidad e integridad de los datos a todas las regiones.

Servicio de Almacenamiento Distribuido: Se cuenta con un clúster de *Ceph* para brindar almacenamiento con alta disponibilidad y redundancia, para brindar seguridad en los datos.

Panel de administración: Para la gestión de los recursos de las regiones conectados se hace uso del componente de *Horizon*, proporcionando una interfaz intuitiva y centralizada. La autenticación y autorización se gestiona mediante Keystone que se comunica con un servicio ***LDAP*** donde se almacenan los usuarios federados de las regiones incluyendo usuarios administrativos.

Servicio de Nombre de Dominio: Se ha configurado un servidor DNS con Bind9 para resolver nombres de dominio para toda la infraestructura y para las regiones así evitar configurar los componentes usando direcciones IP.

Red Privada Virtual: Para que exista la comunicación privada entre la infraestructura y las regiones. A través de esta red las regiones consumen los servicios servidos por la infraestructura y a la vez, la infraestructura administre los recursos de las regiones.

6.3. Configuración del Entorno

A continuación, se detalla el hardware utilizado en base a los requerimientos del prototipo, se describirá a nivel de máquinas virtuales y sus componentes.

Nodo controlador: Contiene el servicio de identidad de *OpenStack*, se encarga de la autenticación y autorización distribuida de múltiples usuarios. Al igual que la interfaz para la gestión y administración de la infraestructura. Para un correcto y optimo funcionamiento de este nodo se requiere el siguiente hardware:

Tabla 1
Requerimientos de nodo controlador que conforma la infraestructura

Nodo	CPU (Núcleos)	RAM (GB)	Almacenamiento	OS
Controlador	4	8	200 GB	Ubuntu

Nota. Requerimientos necesarios en nodo controlador de CPU, RAM, almacenamiento y sistema operativo para la infraestructura de una nube comunitaria basada en OpenStack. Autoría propia.

Nodo identidad: Contiene el protocolo de autenticación para la gestión de información como usuarios, características de usuarios, permisos de usuarios, entre otros; con la finalidad de permitir la autenticación de los recursos y de las regiones que pertenecen a la infraestructura. Para su optimo funcionamiento se requiere lo siguiente:

Tabla 2
Requerimientos de nodo identidad que conforma la infraestructura

Nodo	CPU (Núcleos)	RAM (GB)	Almacenamiento	OS
Identidad	4	8	50 GB	Ubuntu

Nota. Requerimientos necesarios en nodo identidad de CPU, RAM, almacenamiento y sistema operativo para la infraestructura de una nube comunitaria basada en OpenStack. Autoría propia.

Nodo servicios: Se encarga de mejorar el rendimiento por medio del almacenamiento temporal de datos en memoria RAM utilizando el servicio de OpenStack Memcached, contiene a su vez el servicio de comunicación e intercambio de datos entre distintos componentes de OpenStack facilitando la comunicación asíncrona de estos y contiene el clúster de base de datos que se encargarán del almacenamiento de los datos de la infraestructura y las regiones. Sus requerimientos son los siguientes:

Tabla 3
Requerimientos de nodo servicios que conforma la infraestructura

Nodo	CPU (Núcleos)	RAM (GB)	Almacenamiento	OS
Servicios	4	8	150 GB	Ubuntu

Nota. Requerimientos necesarios en núcleo nodo de CPU, RAM, almacenamiento y sistema operativo para la infraestructura de una nube comunitaria basada en OpenStack. Autoría propia.

Nodo DNS: Su finalidad principal es la comunicación de instancias y recursos de toda la infraestructura por medio de la resolución de nombres de dominio (nombres en lugar de direcciones IP).

Tabla 4
Requerimientos de nodo DNS que conforma la infraestructura

Nodo	CPU (Núcleos)	RAM (GB)	Almacenamiento	OS
DNS	2	4	25 GB	Ubuntu

Nota. Requerimientos necesarios en nodo DNS de CPU, RAM, almacenamiento y sistema operativo para la infraestructura de una nube comunitaria basada en OpenStack. Autoría propia.

Nodo Maxscale: Contiene el servicio de balanceo de carga que va hacia un clúster de servidores de base de datos. También se le ha habilitado un Panel para administrar y ver estadísticas sobre el tráfico y configuración del clúster.

Tabla 5
Requerimientos de nodo balanceador que conforma la infraestructura

Nodo	CPU (Núcleos)	RAM (GB)	Almacenamiento	OS
Maxscale	2	2	50 GB	Ubuntu

Nota. Requerimientos necesarios en nodo balanceador de carga de base de datos, de CPU, RAM, almacenamiento y sistema operativo para la infraestructura de una nube comunitaria basada en OpenStack. Autoría propia.

Nodo de Base de Datos: Contienen el servidor de base de datos *MySQL* que se brinda a toda la nube comunitaria, configurados como clúster.

Tabla 6
Requerimientos de nodos base de datos que conforman la infraestructura

Nodo	CPU (Núcleos)	RAM (GB)	Almacenamiento	OS
Maria-1	2	2	50 GB	Ubuntu
Maria-2	2	2	50 GB	Ubuntu
Maria-3	2	2	50 GB	Ubuntu

Nota. Requerimientos necesarios en nodos base de datos, de CPU, RAM, almacenamiento y sistema operativo para la infraestructura de una nube comunitaria basada en OpenStack. Autoría propia.

Nodo de Ceph: Contienen el servicio de almacenamiento distribuido ceph. Se divide en nodos administradores y nodos *OSD* que guardan los datos.

Tabla 7
Requerimientos de los nodos de ceph que conforma la infraestructura

Nodo	CPU (Núcleos)	RAM (GB)	Almacenamiento	OS
Ceph-Admin	2	4	50 GB	Ubuntu
Ceph-Mon	2	4	50 GB	Ubuntu
Ceph-OSD-1	2	2	100 GB + 100 GB	Ubuntu
Ceph-OSD-2	2	2	100 GB + 100 GB	Ubuntu
Ceph-OSD-3	2	2	100 GB + 100 GB	Ubuntu

Nota. Requerimientos necesarios en los nodos del clúster de ceph, de CPU, RAM, almacenamiento y sistema operativo para la infraestructura de una nube comunitaria basada en OpenStack. Autoría propia.

6.4. Topología de Red

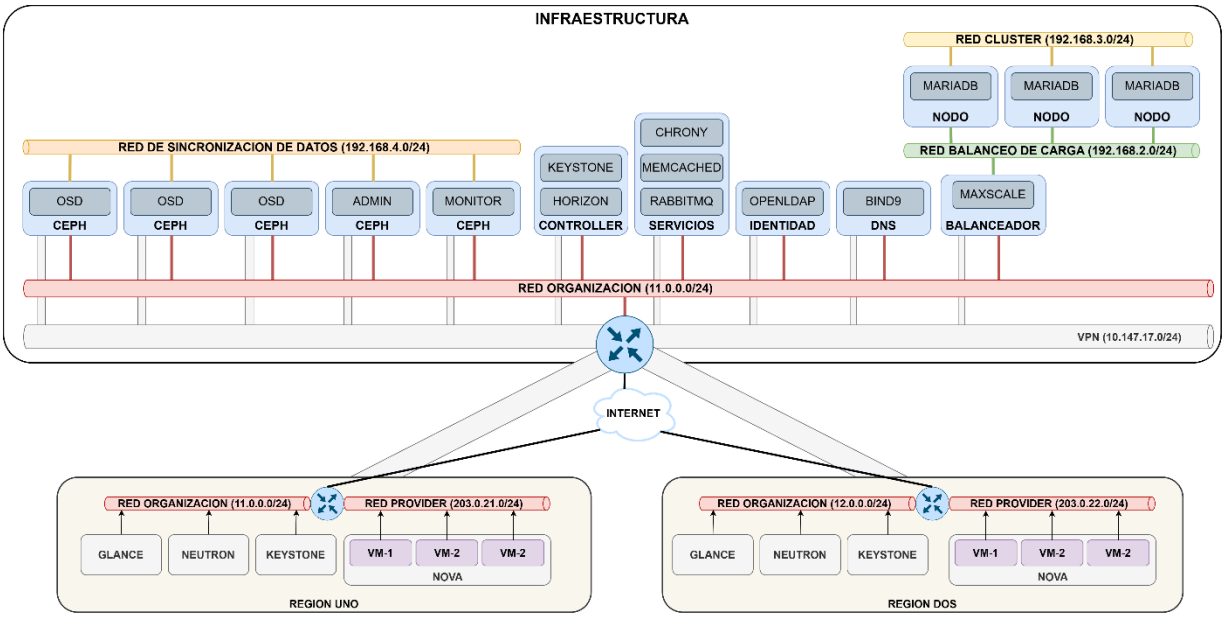


Figura 6.3 Topología de redes para la infraestructura de nube comunitaria

La topología de la infraestructura se conforma de la red principal de la organización o entidad de la infraestructura que es la red expuesta a internet para ser accesible a los servicios públicos.

El clúster galera de *Mariadb* se compone de una red para el manejo de la carga de trabajo que conecta el balanceador de carga con *Maxscale* a los nodos de base de datos. Los nodos sincronizan la información a través de una red privada de alta velocidad para la redundancia de los datos.

Una red muy importante para el funcionamiento de la infraestructura es una red privada virtual o *VPN* que conecta de manera segura y privada las regiones que conforman la nube comunitaria con la infraestructura, para que estas alcancen las *API's* que proveen los servicios.

6.5. Construcción

Ver Anexo 5

6.6. Pruebas

6.6.1. Interfaz

Se ha instalado la VPN en la computadora de prueba para consumir el servicio DNS, así en el navegador colocar el nombre de dominio. La *url* del panel es *http://main.venus-icc115.net/horizon* y redirige a la pantalla de inicio de sesión. Las credenciales son: usuario -> admin, contraseña -> icc115, dominio -> default.

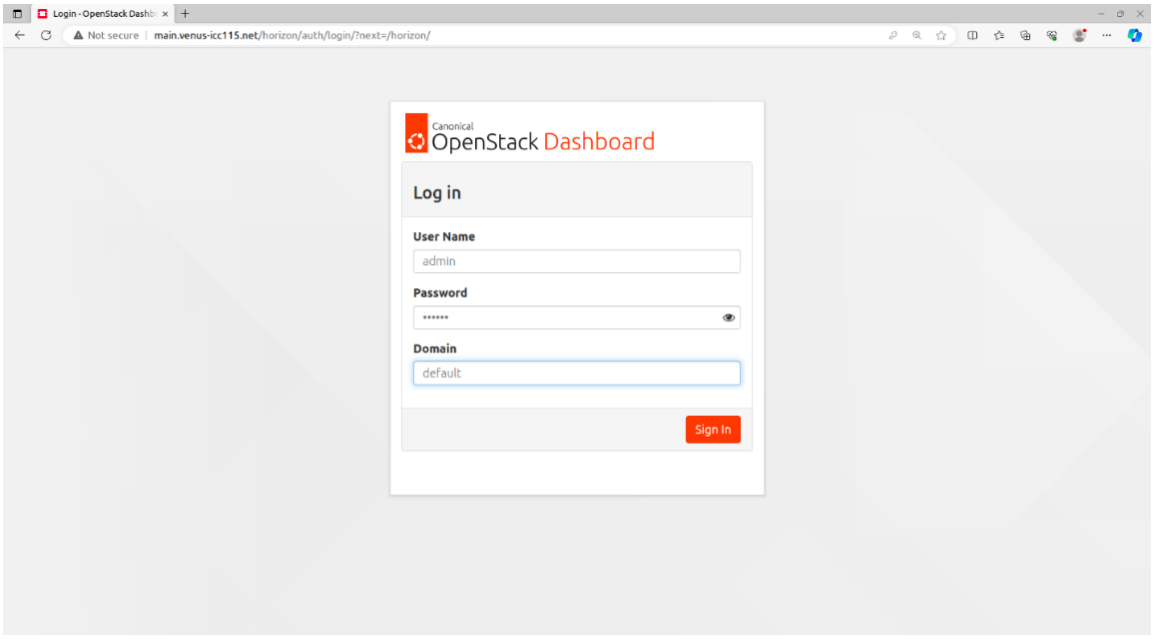


Figura 6.4 Pantalla de inicio de sesión de Horizon en la Infraestructura

Ingresando al panel como un usuario administrador, si es que se tienen los permisos, se puede cambiar de Región desde la barra superior. También se puede cambiar entre los distintos proyectos que pertenezcan al dominio.

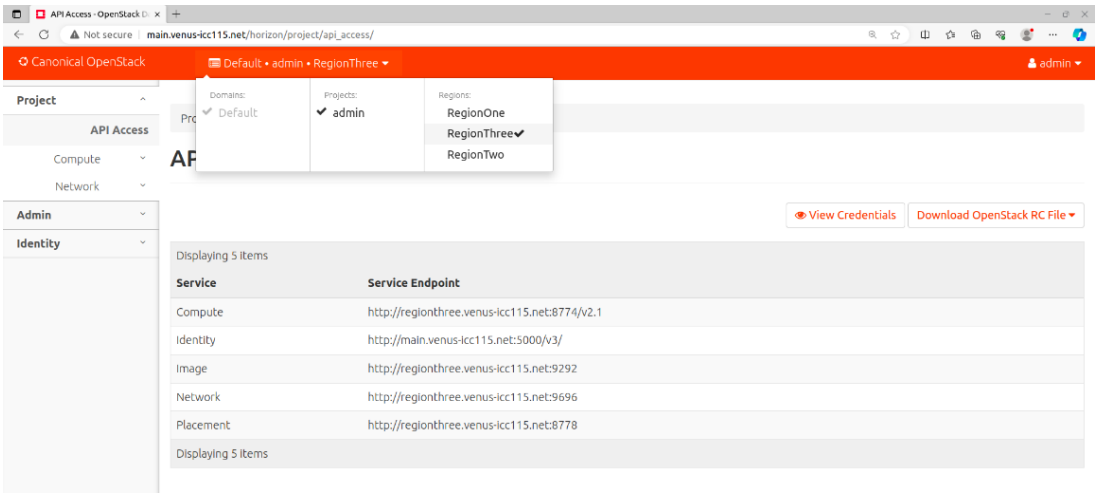


Figura 6.5 Panel de acceso a distintas regiones

El panel cuenta con una vista grafica de la configuración de redes que está configurada en la región seleccionada, la cual presenta las redes, router e instancias de máquina virtual.

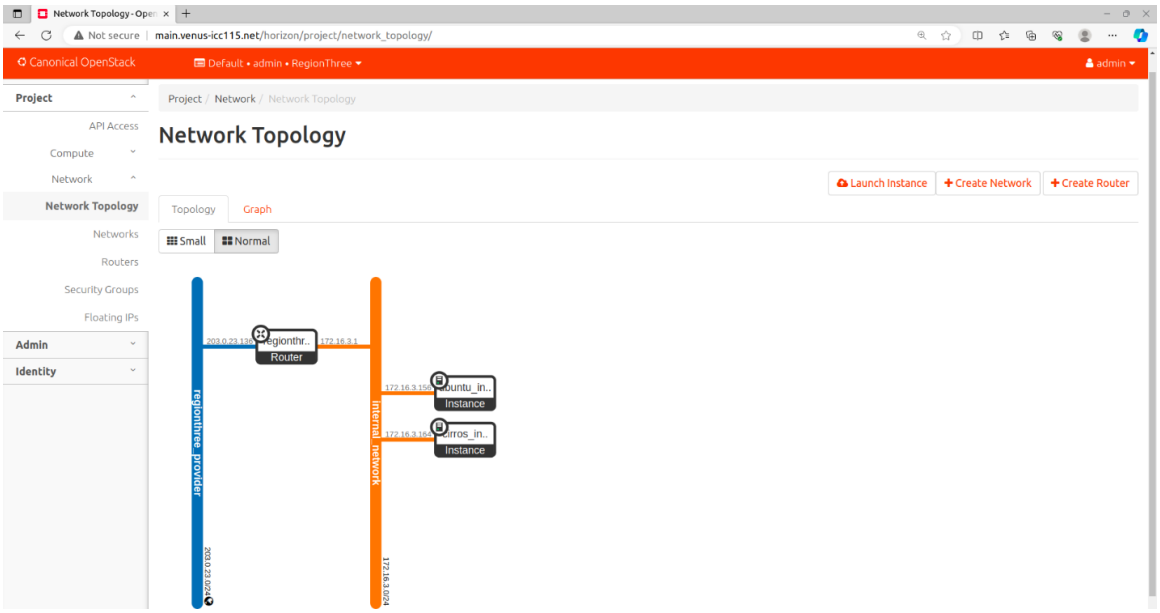


Figura 6.6 Topología de red en una región

Dentro de la sección de administración, se puede obtener información de los agentes y servicios de computación y red que están configurados. En la imagen siguiente se puede ver que está activo el servicio de conductor, planificación en el nodo *controller* y en los nodos compute (*compute-1* y *compute-2*) el servicio de *nova-compute*.

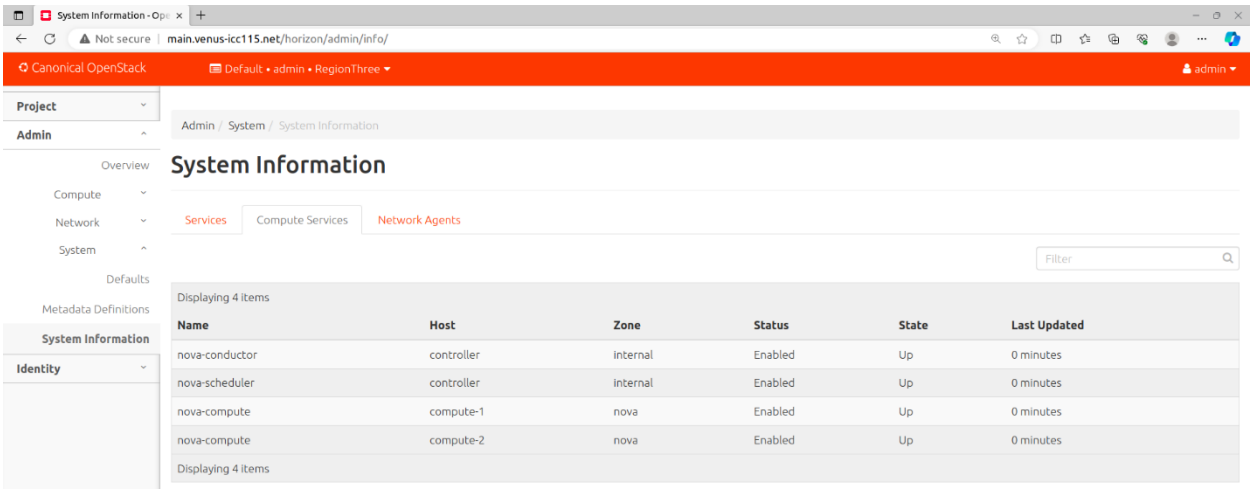


Figura 6.7 Información de servicios de Cómputo de una región

El panel cuenta con una vista a todas las máquinas virtuales creadas en el proyecto, dando información a primera vista del nombre, información de red, estado, zona de disponibilidad, entre otros. La imagen siguiente muestra que está corriendo una máquina virtual llamada "*ubuntu_instance*", la cual tiene una IP flotante para acceder desde la red externa mediante la dirección 203.0.23.152.

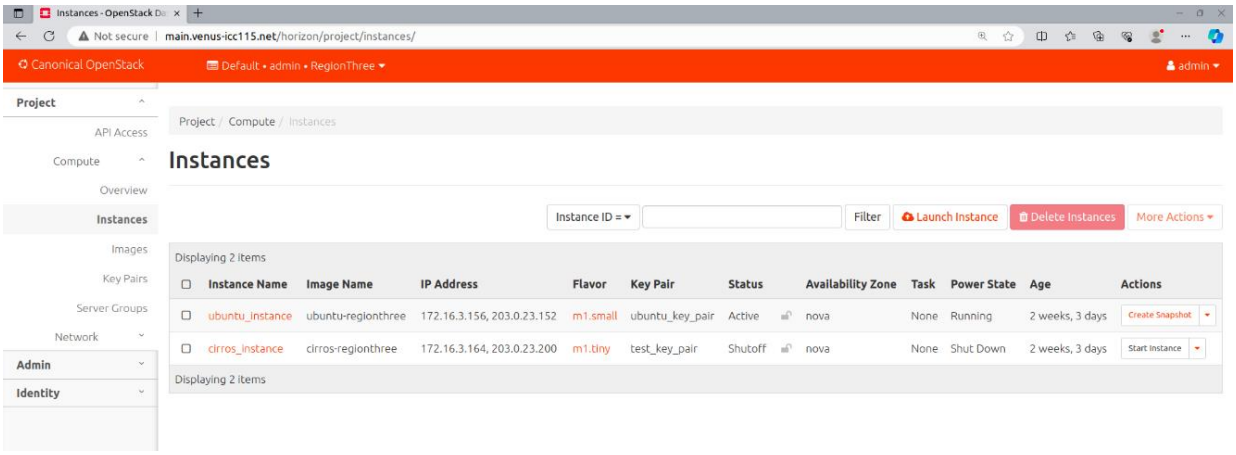


Figura 6.8 Listado de instancias

Al ingresar a esa dirección *IP* desde el navegador, se puede ver que se ha configurado en esa máquina virtual una página web con el gestor de contenido *Drupal CMS* con sitio de pruebas

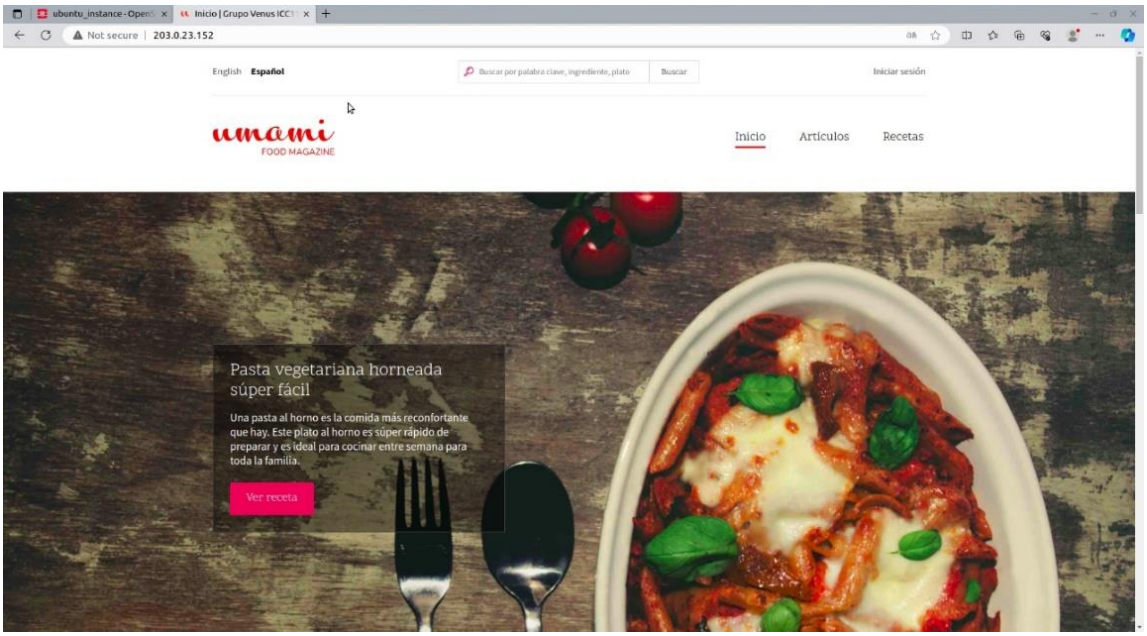


Figura 6.9 Drupal CMS corriendo en instancia

Desde el panel se puede acceder a la máquina virtual mediante un servicio de VNC para observar y realizar acciones en la máquina virtual en una terminal. La máquina cuenta con acceso a internet para descargar paquetes.

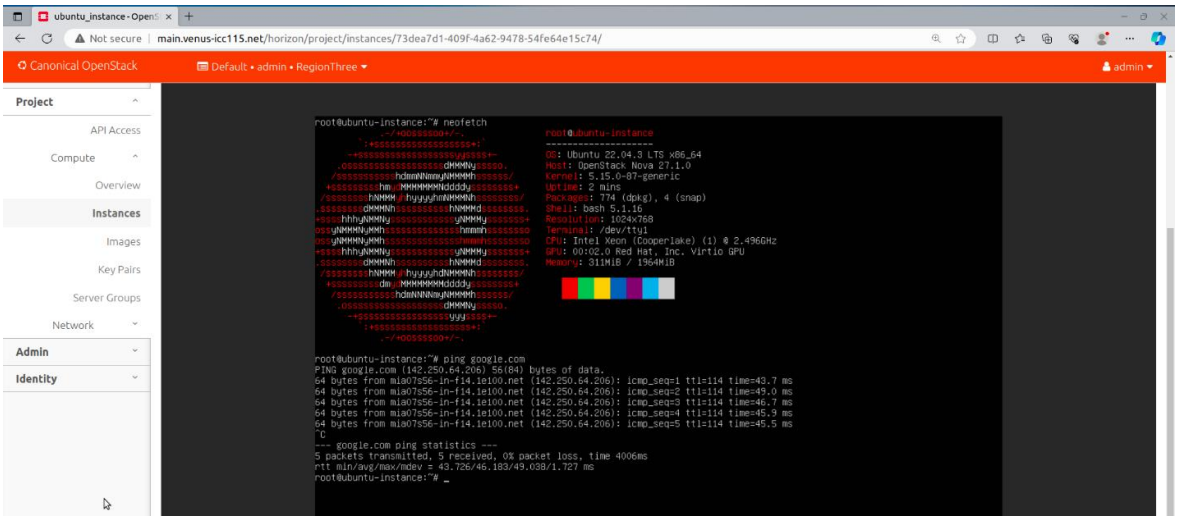


Figura 6.10 Acceso a terminal de instancia desde Horizon

6.6.2. Base de datos

El servicio de base de datos se visualiza desde el panel de *Maxscale*. La primera vista es una lista de los nodos de base de datos que se tiene monitorizados mediante un servicio de *galera-monitor*.

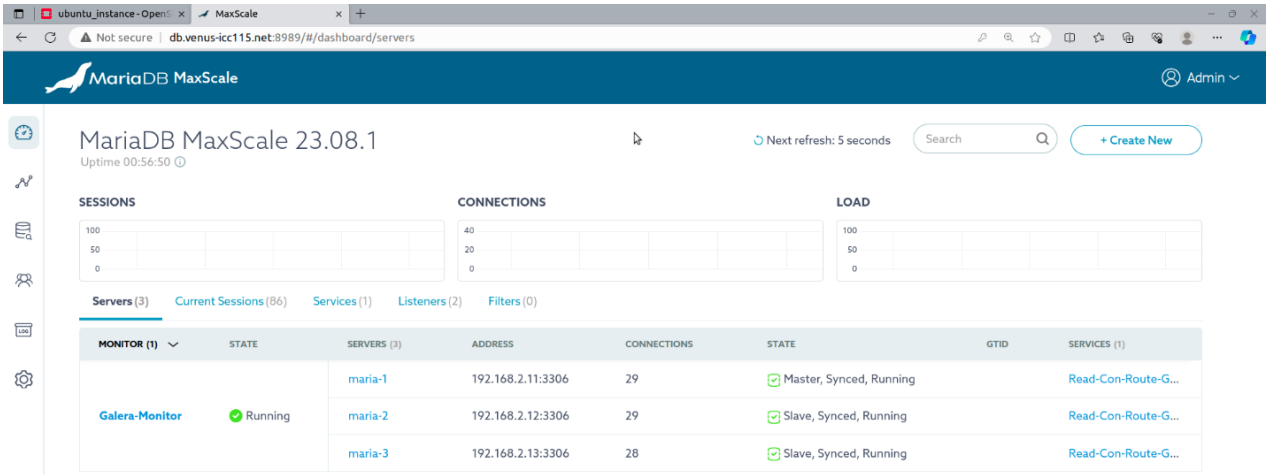


Figura 6.11 Interfaz de Administración de Maxscale

Para tener una vista más clara del servicio completo de base de datos configurado, Maxscale cuenta con un diagrama jerárquico de los nodos y servicios. El servicio balanceador escucha por dos interfaces, por la interfaz de la red de la organización 13.0.0.15 y por la interfaz de la VPN 10.147.17.36. Balancea la carga a los 3 nodos de base de datos con *MariaDB* y estos tiene un servicio monitor para sincronizar los datos.

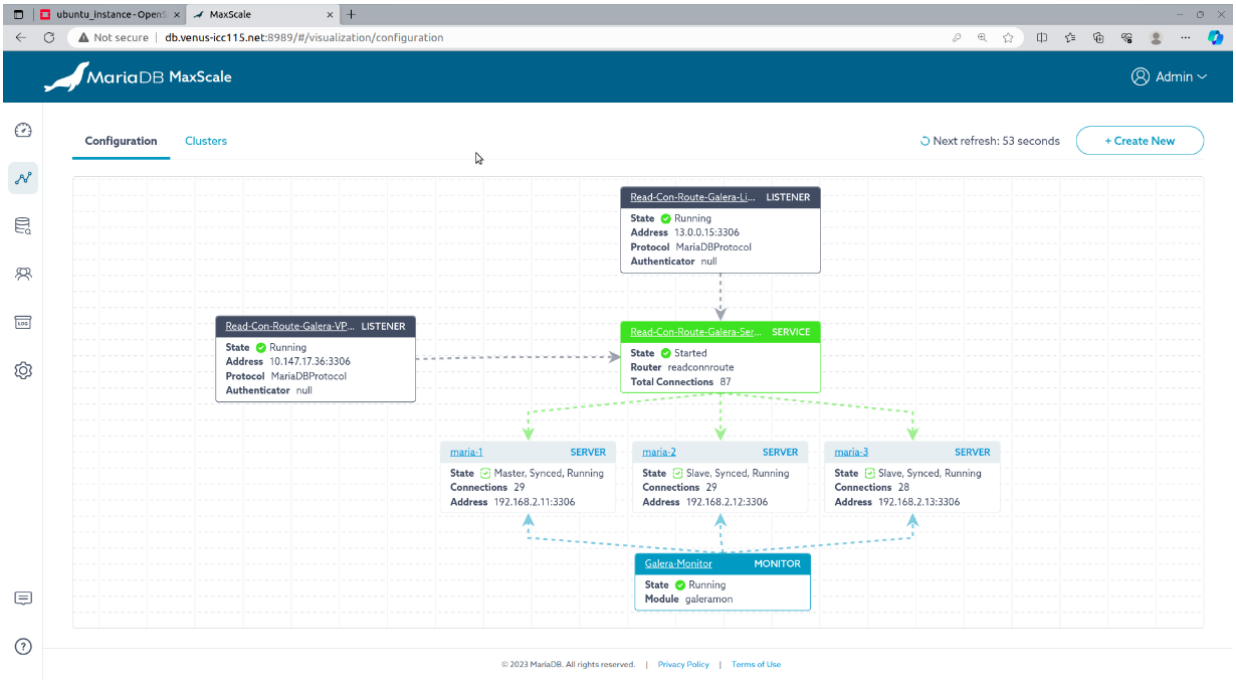


Figura 6.12 Diagrama del Clúster de Base de Datos

6.6.3. Cola de Mensajes

Se ha activado un panel de monitoreo en el servicio de RabbitMQ en el cual se inicia sesión con el usuario que se ha habilitado como administrador.

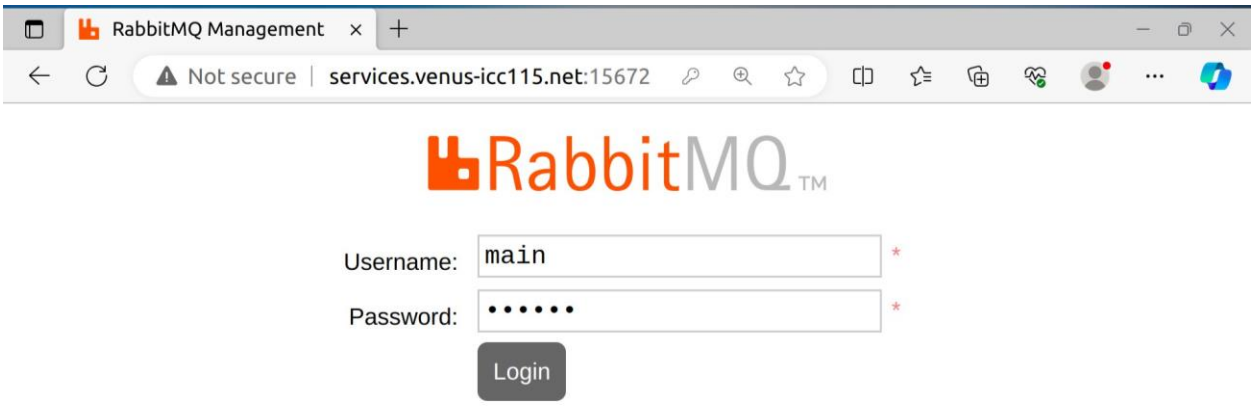


Figura 6.13 Panel de Administración de RabbitMQ

El panel cuenta con una vista general del tráfico hacia el servicio de colas mediante gráficos dinámicos.

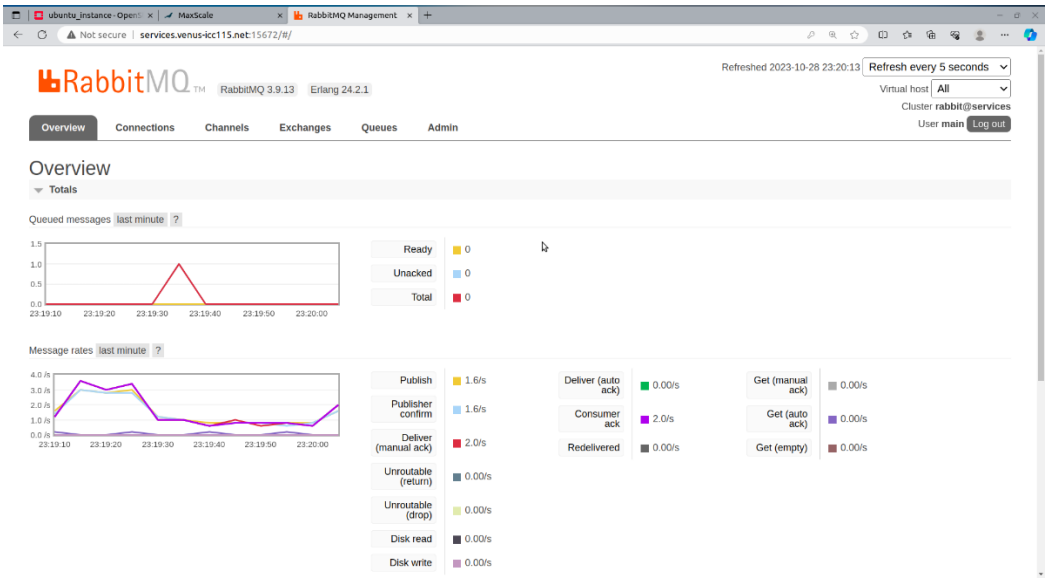


Figura 6.14 Panel de Administración de RabbitMQ

El panel muestra información sobre los puertos de los servicios adicionales configurados y activados para el servidor.

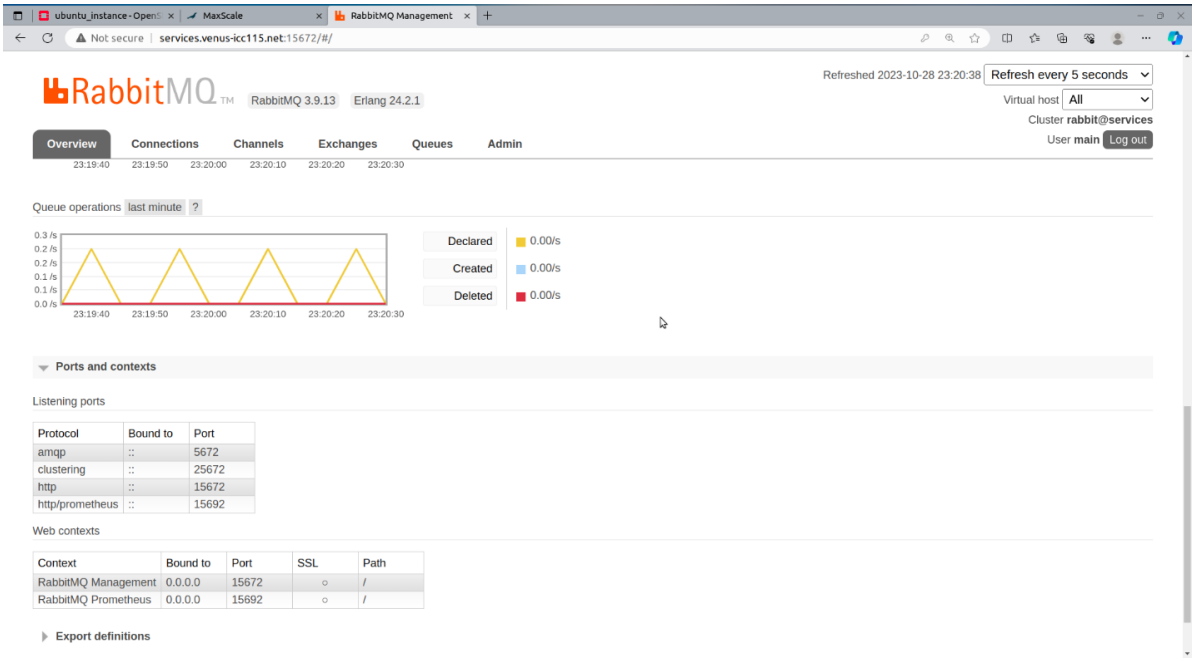


Figura 6.15 Información de conexiones a servicio de mensajes

6.6.4. LDAP

Para administrar el servicio LDAP se ha instalado el componente *LAM* (*LDAP Account Manager*) el cual es una interfaz gráfica más intuitiva al uso de la terminal.

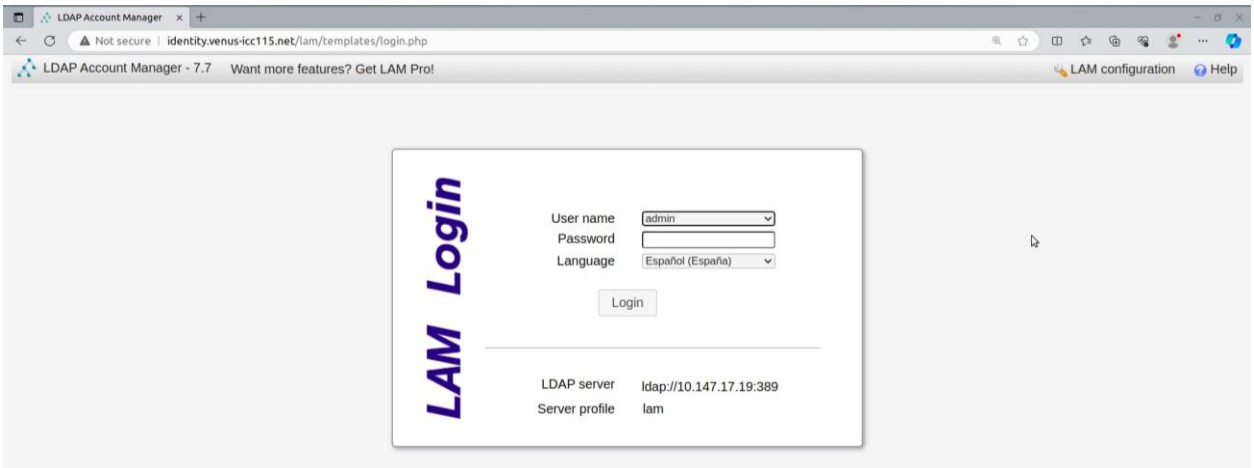


Figura 6.16 Pantalla de inicio de sesión LAM

Se puede ver exitosamente la lista de usuarios almacenados en el LDAP con sus atributos, así como acciones rápidas de administración de cada uno.

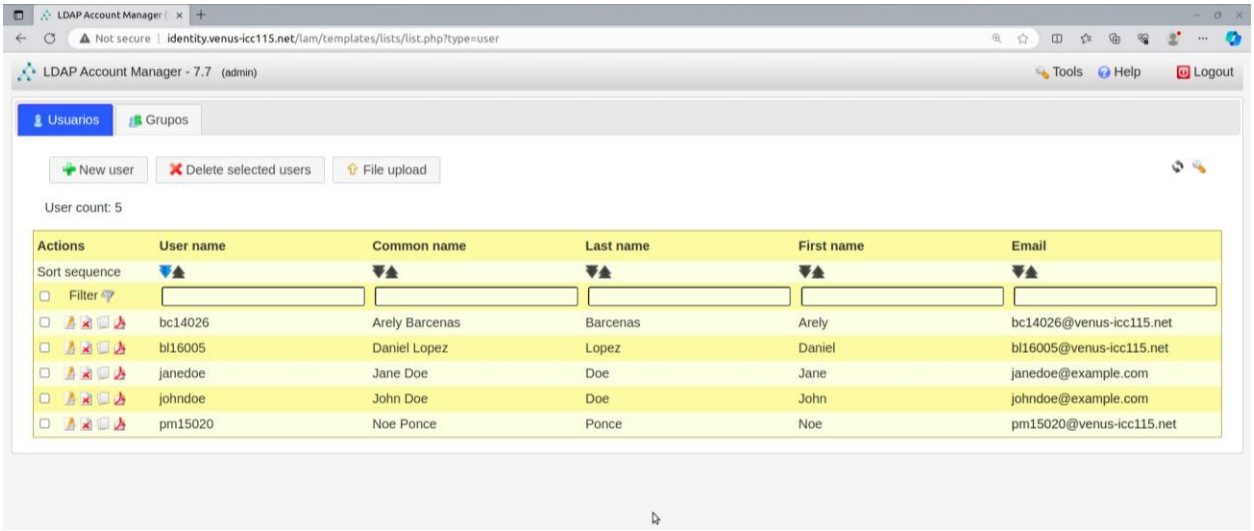


Figura 6.17 Usuarios almacenados en el LDAP

6.6.5. Memcached

En *Datadog*, en el *dashboard* de Memcached se encuentra información general del servicio, así como gráficos dinámicos para la monitorización de los recursos usados.

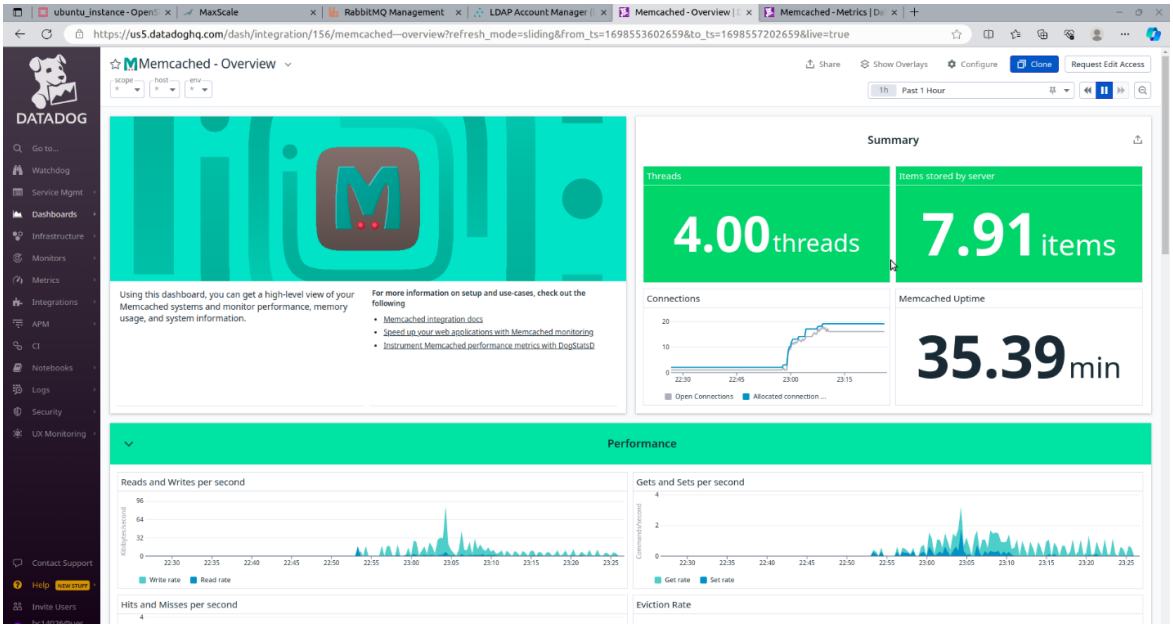


Figura 6.18 Panel de monitoreo del servicio Memcached en Datadog

El *dashboard* de Memcached cuenta con muchas métricas que miden el rendimiento en vivo del servicio y conexiones de caché usado por las regiones y la infraestructura.

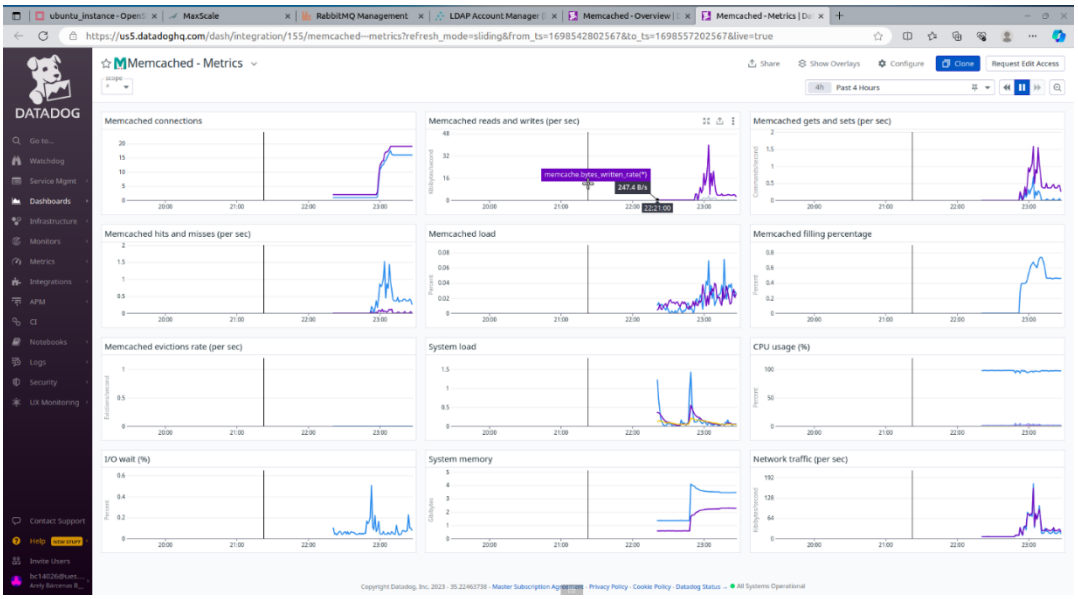


Figura 6.19 Vista de métricas de Memcached en Datadog

7 Caso de aplicación del prototipo

Las empresas de tecnología que buscan expandir sus operaciones a nivel de sucursales, ya sea internacionalmente o dentro de un mismo país, pueden enfrentarse a desafíos significativos en la gestión de su infraestructura de TI. Una infraestructura como la diseñada y construida en este trabajo de grado ofrece una solución escalable y flexible para apoyar operaciones regionales, mejorar la continuidad del negocio y ofrecer servicios digitales eficientes.

La sede principal de la empresa tendría configurado un centro de datos con OpenStack, integrando servicios fundamentales como base de datos y almacenamiento distribuido. Estos servicios estarán disponibles para la nueva sucursal a través de una conexión VPN segura, garantizando la cohesión y la seguridad de la información entre todas las operaciones regionales. Esta configuración funcionaría como infraestructura central.

En la infraestructura se implementaría un servidor con Keystone en el cual se gestionará la autenticación, gestión de usuarios y el acceso de estos a los sistemas de la empresa mediante la conexión a un directorio LDAP. Esto proporcionará un control centralizado de usuarios, facilitando la gestión uniforme a través de la sede principal y la nueva sucursal.

El almacenamiento de todos los archivos, objetos y bloques de la nueva sucursal y la actual sede, se alojarían en un clúster de Ceph, el cual brinda alta disponibilidad y alta redundancia para mayor seguridad y respaldo de la información.

Con la base de datos centralizada, mediante balanceadores de carga *maxscale,* garantizaría alta disponibilidad en cuanto al manejo y procesamiento de datos e información.

Para acelerar la entrega de contenido dinámico se haría uso de un servidor de caché con Memcached, mejorando la velocidad de carga de contenido, al mismo tiempo para aplicaciones interactivas y transaccionales se haría uso del servicio de RabbitMQ, para procesar mensajes en tiempo real de manera eficiente.

La nueva sucursal, al ser implementada sobre openstack aprovecharía cada uno de los servicios anteriormente mencionados, y centraría sus recursos únicamente para otorgar el servicio a los clientes que consumen los recursos de dicha sucursal, mejorando la conectividad y la eficiencia al entregar los servicios de tecnología ofrecidos.

Los recursos necesarios para implementar a producción el prototipo visto en este documento, dependen del tamaño que se piense brindar como infraestructura. En los siguientes capítulos se hace un estudio de factibilidad de lo que podría ser una implementación con cantidad de recursos más reales, y no únicamente un prototipo como el que se ha diseñado y construido para este trabajo de grado.

8 Factibilidad

La evaluación de factibilidad es un paso crucial en la toma de decisiones estratégicas para la implementación de una infraestructura en la nube. En este contexto, se ha desarrollado un prototipo de infraestructura en la nube utilizando *OpenStack* como plataforma. Sin embargo, antes de comprometer recursos significativos, es esencial determinar si esta solución de nube es la más adecuada en comparación con otras opciones líderes en el mercado, como *AWS, Azure y Google Cloud.*

Es fundamental analizar en detalle los componentes específicos de *OpenStack* para una infraestructura de nube comunitaria en un ambiente de producción como se detallará más adelante. Además, debemos comparar estos componentes con las alternativas disponibles en *AWS, Azure y Google Cloud* para determinar cuál plataforma ofrece la mejor combinación de características y beneficios para implementar dicha infraestructura.

Tabla 8
Principales componentes de nube

Componente	OpenStack	AWS	Google Cloud	Azure
Identidad	Keystone	IAM	Identity and Access Management	Azure Active Directory
Computo	Nova	EC2	Compute Engine	Virtual Machines
Redes	Neutron	VPC	VPC	Virtual Network
Almacenamiento en bloque	Cinder	EBS	Persistent Disk	Managed Disks
Almacenamiento de objetos	Swift	S3	Cloud Storage	Almacenamiento Blob
Base de datos	Trove	Amazon RDS	Azure SQL DB	Cloud SQL
Interfaz	Horizon	AWS Console	Azure Portal	Google Cloud Console

Nota. Nombres que tienen los componentes de la nube en las soluciones de nube OpenStack, AWS, Google Cloud y Microsoft Azure.

A continuación, se describe las fortalezas de cada uno de los componentes en las soluciones de nube.

Tabla 9
Comparación de Componentes de Nube

Componente	OpenStack	AWS	Azure	Google Cloud
Identidad	Personalización y control sobre la gestión de identidades y acceso, ideal para organizaciones con requisitos específicos.	Flexibilidad en la administración de permisos y usuarios, con opciones de acceso a una amplia gama de servicios de AWS.	Integración sólida con servicios y aplicaciones de Azure, facilitando la administración de identidades en un entorno Microsoft.	Controles detallados sobre el acceso a recursos y servicios en Google Cloud, con facilidad de gestión.
Computo	Escalabilidad y orquestación de recursos de cómputo en un entorno de nube de código abierto, adecuado para organizaciones que desean un alto grado de control.	Amplia gama de instancias virtuales, opciones de escalabilidad y servicios de orquestación en AWS, respaldada por una sólida infraestructura global.	Máquinas virtuales flexibles y escalables en la nube de Azure, con integración profunda en el ecosistema de Microsoft.	Máquinas virtuales altamente personalizables y escalables, con una infraestructura de alto rendimiento.
Redes	Control total sobre la configuración de redes virtuales y componentes, ideal para organizaciones que necesitan personalización.	AWS VPC ofrece aislamiento y control sobre la conectividad de redes virtuales, respaldado por la amplia presencia global de AWS.	Redes virtuales flexibles y aisladas en Azure, con integración en el ecosistema de Microsoft y opciones avanzadas de seguridad.	Redes virtuales personalizables y escalables con opciones de alta disponibilidad.
Almacenamiento en bloque	Almacenamiento persistente en bloque en un entorno de código abierto, adecuado para organizaciones que buscan una solución de costo efectivo.	EBS ofrece almacenamiento en bloque de alto rendimiento con opciones de instantáneas y escalabilidad en AWS.	Almacenamiento en bloque persistente y escalable con opciones avanzadas en Azure.	Persistent Disk proporciona almacenamiento en bloque fiable y escalable en Google Cloud.
Almacenamiento de objetos	Almacenamiento de objetos de código abierto adecuado para organizaciones que requieren	AWS S3 ofrece un servicio altamente escalable y altamente duradero para almacenamiento	Azure Blob Storage es compatible con datos no estructurados a gran escala y se	Google Cloud Storage proporciona un almacenamiento de objetos altamente

	flexibilidad y control sobre sus datos no estructurados.	de objetos, respaldado por una amplia gama de funciones.	integra con servicios de análisis y aplicaciones.	confiable y escalable para una variedad de aplicaciones.
Base de datos	Trove simplifica la gestión de bases de datos SQL y NoSQL en un entorno de código abierto, adecuado para organizaciones que desean una solución sencilla.	AWS RDS ofrece bases de datos gestionadas altamente disponibles y escalables, con soporte para varias bases de datos relacionales.	Azure Database proporciona una amplia gama de servicios de bases de datos gestionadas con opciones de escalabilidad.	Cloud SQL es un servicio de bases de datos gestionadas para aplicaciones web y móviles en Google Cloud.
Interfaz	Horizon es el panel de control web en un entorno de código abierto, adecuado para organizaciones que buscan flexibilidad y personalización en la administración de recursos.	AWS Management Console proporciona una interfaz web intuitiva y unificada para administrar una amplia gama de servicios y recursos de AWS.	Azure Portal es la interfaz de administración basada en web que ofrece una experiencia unificada para gestionar servicios y recursos en Azure.	Google Cloud Console proporciona una interfaz web amigable para administrar recursos en Google Cloud, con una amplia gama de servicios y aplicaciones integrados.

Nota. Descripción de los componentes de las diferentes alternativas de nube.

8.1. Factibilidad Económica

La implementación de una infraestructura de nube comunitaria con *OpenStack* representa una decisión estratégica para muchas organizaciones. La infraestructura de nube comunitaria ofrece ventajas significativas en términos de flexibilidad, control y colaboración, pero también es fundamental considerar la factibilidad económica antes de tomar esta decisión. En este contexto, la comparación de costos con proveedores de nube pública líderes como *Azure, Google Cloud y AWS* es esencial para evaluar adecuadamente la inversión.

Los precios de los servicios de computación en la nube como *Amazon Web Services (AWS), Microsoft Azure y Google Cloud* varían en base a factores como:

- Tipo de almacenamiento
- Ubicación geográfica del servicio
- Tipo de instancia

Para la comparación de precios de los proveedores mencionados anteriormente se ha utilizado una herramienta en internet llamada *Cloudorado*, la cual calcula el precio de servidores en la nube con base en especificaciones como cantidad de memoria RAM, almacenamiento, núcleos de CPU y sistema operativo. A la vez muestra un listado de diferentes proveedores de servicios de nube con distintos precios de cada uno.

Tomando en cuenta lo anterior, para la infraestructura de nube comunitaria en un ambiente de producción con un rendimiento óptimo, se tienen los siguientes requerimientos mostrados en la siguiente tabla:

Tabla 10
Requerimientos de nodos que conforman la infraestructura

Nodo	CPU (Núcleos)	RAM (GB)	Almacenamiento	OS
Controlador	8	32	256 GB	Ubuntu
Identidad	8	16	256 GB	Ubuntu
Colas de peticiones	8	16	256 GB	Ubuntu
Maxscale	8	32	256 GB	Ubuntu
MariaDB-1	8	32	256 GB	Ubuntu
MariaDB-2	8	32	256 GB	Ubuntu
MariaDB-3	8	32	256 GB	Ubuntu
Ceph-Admin	8	32	500 GB	Ubuntu
Ceph-Mon	8	32	500 GB	Ubuntu
Ceph-OSD-1	8	32	1024 GB	Ubuntu
Ceph-OSD-2	8	32	1024 GB	Ubuntu
Ceph-OSD-3	8	32	1024 GB	Ubuntu

Nota. Requerimientos necesarios en núcleos de CPU, RAM, almacenamiento y sistema operativo para la infraestructura de una nube comunitaria basada en OpenStack. Autoría propia.

A continuación, se presentarán los costos detallados con base en los elementos mencionados anteriormente para cada uno de los nodos de la infraestructura:

Tabla 11
Precios por mes y año para nodo Controlador

Proveedor	Precio/mes	Precio/año
Google Cloud	$ 213.00	$ 2,556.00
Microsoft Azure	$ 276.00	$ 3,312.00
Amazon WS	$ 268.00	$ 3,216.00

Nota. Especifica los precios por mes y año de los proveedores de servidores de nube escogidos.

Tabla 12
Precios por mes y año para nodo Identidad

Proveedor	Precio/mes	Precio/año
Google Cloud	$ 178.00	$ 2,136.00
Microsoft Azure	$ 245.00	$ 2,940.00
Amazon WS	$ 265.00	$ 3,180.00

Nota. Especifica los precios por mes y año de los proveedores de servidores de nube escogidos

Tabla 13
Precios por mes y año para nodo de Cola de peticiones

Proveedor	Precio/mes	Precio/año
Google Cloud	$ 178.00	$ 2,136.00
Microsoft Azure	$ 245.00	$ 2,940.00
Amazon WS	$ 265.00	$ 3,180.00

Nota. Precios por mes y año de los proveedores de servidores de nube escogidos.

Tabla 14
Precios por mes y año para nodo de balanceador de carga Maxscale

Proveedor	Precio/mes	Precio/año
Google Cloud	$ 213.00	$ 2,556.00
Microsoft Azure	$ 276.00	$ 3,312.00
Amazon WS	$ 268.00	$ 3,216.00

Nota. Precios por mes y año de los proveedores de servidores de nube escogidos.

Tabla 15
Precios por mes y año para nodo de Base de datos MariaDB-1

Proveedor	Precio/mes	Precio/año
Google Cloud	$ 213.00	$ 2,556.00
Microsoft Azure	$ 276.00	$ 3,312.00
Amazon WS	$ 268.00	$ 3,216.00

Nota. Precios por mes y año de los proveedores de servidores de nube escogidos.

Tabla 16
Precios por mes y año para nodo de Base de datos MariaDB-2

Proveedor	Precio/mes	Precio/año
Google Cloud	$ 213.00	$ 2,556.00
Microsoft Azure	$ 276.00	$ 3,312.00
Amazon WS	$ 268.00	$ 3,216.00

Nota. Precios por mes y año de los proveedores de servidores de nube escogidos.

Tabla 17
Precios por mes y año para nodo de Base de datos MariaDB-3

Proveedor	Precio/mes	Precio/año
Google Cloud	$ 213.00	$ 2,556.00
Microsoft Azure	$ 276.00	$ 3,312.00
Amazon WS	$ 268.00	$ 3,216.00

Nota. Precios por mes y año de los proveedores de servidores de nube escogidos.

Tabla 18
Precios por mes y año para nodo de almacenamiento Ceph-Admin

Proveedor	Precio/mes	Precio/año
Google Cloud	$ 225.00	$ 2,700.00
Microsoft Azure	$ 290.00	$ 3,480.00
Amazon WS	$ 298.00	$ 3,576.00

Nota. Precios por mes y año de los proveedores de servidores de nube escogidos.

Tabla 19
Precios por mes y año para nodo de almacenamiento Ceph-Mon

Proveedor	Precio/mes	Precio/año
Google Cloud	$ 225.00	$ 2,700.00
Microsoft Azure	$ 290.00	$ 3,480.00
Amazon WS	$ 298.00	$ 3,576.00

Nota. Precios por mes y año de los proveedores de servidores de nube escogidos.

Tabla 20
Precios por mes y año para nodo de almacenamiento Ceph-OSD-1

Proveedor	Precio/mes	Precio/año
Google Cloud	$ 245.00	$ 2,940.00
Microsoft Azure	$ 312.00	$ 3,432.00
Amazon WS	$ 348.00	$ 3,828.00

Nota. Precios por mes y año de los proveedores de servidores de nube escogidos.

Tabla 21
Precios por mes y año para nodo de almacenamiento Ceph-OSD-2

Proveedor	Precio/mes	Precio/año
Google Cloud	$ 245.00	$ 2,940.00
Microsoft Azure	$ 312.00	$ 3,432.00
Amazon WS	$ 348.00	$ 3,828.00

Nota. Precios por mes y año de los proveedores de servidores de nube escogidos.

Tabla 22
Precios por mes y año para nodo de almacenamiento Ceph-OSD-3

Proveedor	Precio/mes	Precio/año
Google Cloud	$ 245.00	$ 2,940.00
Microsoft Azure	$ 312.00	$ 3,432.00
Amazon WS	$ 348.00	$ 3,828.00

Nota. Precios por mes y año de los proveedores de servidores de nube escogidos.

Consolidando todos los nodos a utilizar, los precios por mes y año de las instancias requeridas para la infraestructura, se muestran a continuación en la tabla 23:

Tabla 23
Consolidado de precios por mes y año para todos los nodos

Proveedor	Precio/mes	Precio/año
Google Cloud	$ 2,606.00	$ 31,272.00
Microsoft Azure	$ 3,386.00	$ 40,632.00
Amazon WS	$ 3,510.00	$ 42,120.00

Nota. Consolidado de precios por mes y año para las instancias requeridas para el funcionamiento óptimo de la infraestructura.

Con base en lo anterior se puede observar que el proveedor de servicios de computación en la nube *Google Cloud*, presenta la alternativa de menor costo para llevar a cabo una comparación al valor presente con la inversión inicial en la implementación de la nube comunitaria.

Para la implementación de una infraestructura en un ambiente de producción en base a los requerimientos de cada instancia que la conforman se ha consultado el precio del hardware necesario, el cual se detalla a continuación:

Tabla 24
Información de hardware para prototipo de infraestructura

Nodo	Modelo	Procesador	RAM (GB)	Alm. (GB)	Precio
Controlador	PowerEdge R740	Intel Xeon Silver 4214R	32	480	$ 3,743.42
Identidad	PowerEdge R740	Intel Xeon Bronce 3206R	16	480	$ 2,953.07
Colas de peticiones	PowerEdge R740	Intel Xeon Bronce 3206R	16	480	$ 2,953.07
Maxscale	PowerEdge R740	Intel Xeon Silver 4214R	32	480	$ 3,743.42
MariaDB-1	PowerEdge R740	Intel Xeon Bronce 3206R	32	480	$ 3,318.00
MariaDB-2	PowerEdge R740	Intel Xeon Bronce 3206R	32	480	$ 3,318.00
MariaDB-3	PowerEdge R740	Intel Xeon Bronce 3206R	32	480	$ 3,318.00
Ceph-Admin	PowerEdge R740	Intel Xeon Silver 4214R	32	480	$ 3,743.42
Ceph-Mon	PowerEdge R740	Intel Xeon Silver 4214R	32	480	$ 3,743.42
Ceph-OSD-1	PowerEdge R740	Intel Xeon Bronce 3206R	32	1024	$ 3,451.00
Ceph-OSD-2	PowerEdge R740	Intel Xeon Bronce 3206R	32	1024	$ 3,451.00
Ceph-OSD-3	PowerEdge R740	Intel Xeon Bronce 3206R	32	1024	$ 3,451.00
				Total	**$ 41,186.82**

Nota. Información sobre el hardware requerido para la implementación a producción del prototipo de infraestructura.

Aparte del equipo de hardware mencionado en la tabla 24, se incorpora un *Switch Ethernet* con un precio de $1,700.00. El costo de implementación se estima de $2,500.00 y el costo de mantenimiento se estima de $500.00 y se realizará dos veces por año. Dando un total de **$46,386.82**.

Con base en el costo de implementación de la infraestructura con *OpenStack* se muestra la tabla de comparaciones para visualizar los gastos que estos incurren:

Proveedor	Año 1	Año 2	Año 3	Año 4	Año 5
Google Cloud	$31,272	$62,544	$93,816	$125,088	$156,360
Microsoft Azure	$40,632	$81,264	$121,896	$162,528	$203,160
Amazon Web Services	$42,120	$84,240	$126,360	$168,480	$210,600
OpenStack	$46,386.82	$47,386.82	$48,386.82	$49,386.82	$50,386.82

En la tabla anterior evidenciamos la diferencia de costos sobre la implementación de una infraestructura de nube de *OpenStack* con las nubes publicas más reconocidas en la actualidad. Se puede observar que en el primer año los costos de implementación de con OpenStack es mucho mayor a las alternativas de nube comparadas, sin embargo, a partir del segundo año, *OpenStack* presenta costos menores.

Con base en lo anterior, para un proyecto a largo plazo la implementación de una infraestructura de nube con *OpenStack* sería más rentable en comparación a otras soluciones de nube como lo son *AWS, Microsoft Azure y Google Cloud.*

8.2. Factibilidad Técnica

Para poder analizar la factibilidad técnica debemos analizar 3 puntos destacados:

Adaptabilidad: Gracias a que se está usando *OpenStack*, la cual es *software* libre y gratuito, el prototipo puede correr en diferentes configuraciones de *hardware* dependiendo de la capacidad de cada organización, que va desde estar montado sobre máquinas virtuales o montado en equipo físicos que van desde una configuración mínima hasta *hardware* de alto rendimiento y avanzado.

Otro aspecto de la adaptabilidad es que admite módulos externos a los desarrollados por la comunidad de *OpenStack*, ya sea por otras empresas de *software* como por la misma organización que lo desea implementar.

Integración: La infraestructura tiene la capacidad de trabajar con diferentes tecnologías y aplicaciones de otras organizaciones o de la misma organización pero que se organiza distinta y sobre diferentes tecnologías, ya que se basa en *APIs* para la comunicación y gestión de los recursos. Se pueden consumir los recursos de la infraestructura y de las regiones administradas por esta, a través de los diferentes *endpoints* y protocolos servidos, como *HTTP*, *HTPS*, *AMQP*, *iSCSI*, *NTP*, *WebSockets,* entre otros.

La integración con servicios de empresas que ofrecen soluciones de nube es otro punto fuerte de *OpenStack*, ya que se muchos basan en protocolos abiertos, lo que permite a la infraestructura pueda conectarse a esos servicios y consumirlos, por ejemplo, *S3* de *AWS* y otros.

Por último, el proveer servicio de Identidad Federada, sobre plataforma de código libre, sea muy fácil integrarse con nuevos nodos de *OpenStack* y que estos consuman el servicio se usuarios.

Escalabilidad: El prototipo tiene la capacidad de empezar en entornos pequeños como para desarrollo, pero puede crecer y adaptarse a la medida de cada una de las necesidades que tengan las organizaciones que pudieran usarlo.

Puede ser escalado tanto horizontalmente, ya sea de cada servicio agregando más instancias de nodos de manera virtual o servidores físicos, como escalar verticalmente, agregando nuevo *hardware* o más capacidad de almacenamiento, computo o carga de trabajo o manejo de usuarios.

También en términos de Regiones, puede escalar para servir a gran cantidad de regiones que se puedan agregar a la nube comunitaria, ya que ese es el fin de este tipo de nube.

9 Conclusiones

La realización de este prototipo demostró que es posible construir una infraestructura de nube capaz de adaptarse a la incorporación de nuevas regiones con facilidad sin comprometer el rendimiento de dicha infraestructura ni de las regiones agregadas a la nube comunitaria.

La infraestructura garantiza la integridad de acceso y autorización a los usuarios mediante un sistema federado transversal en el entorno de la nube comunitaria y los servicios que brinda.

El prototipo permite un mecanismo de persistencia de sesiones exitoso el cual asegura la experiencia de usuario ininterrumpida en todas las regiones y servicios.

Tener una base de datos centralizado ha demostrado un alto nivel de eficiencia operativa gestionando de manera efectiva grandes volúmenes de datos de las múltiples regiones de la nube comunitaria.

La interfaz gráfica de *OpenStack* proporciona una gestión efectiva e intuitiva de las regiones integradas, la infraestructura y sus componentes y recursos.

El prototipo demostró un rendimiento optimo, escalabilidad y adaptabilidad lo que confirma su robustez y fiabilidad ante diversas demandas y cargas de trabajo.

La evaluación de factibilidad económica demostró que la implementación de OpenStack como una Infraestructura de Nube Comunitaria es costo-eficiente a mediano y largo plazo.

10 Recomendaciones

Continuar con la optimización y mejora de la infraestructura para asegurar que su capacidad de adaptarse a la incorporación de nuevas regiones se mantenga eficiente mediante la adopción de tecnologías emergentes o metodologías de mejora continua, realizando supervisión y mantenimiento de los equipos.

Realizar revisiones de seguridad periódicos y actualizar los protocolos conforme las mejores prácticas y estándares de la industria.

Reforzar mecanismos de autenticación y autorización, garantizando mantenerse con las mejores prácticas de seguridad, como 2FA.

11 Referencias

Bigelow., S. J., Neenan, S., & Casey, K. (s.f.). *What is public cloud? Everything you need to know*. Recuperado el 31 de agosto de 2023, de techtarget: https://www.techtarget.com/searchcloudcomputing/definition/public-cloud

Carrero, L. (23 de febrero de 2023). *Stacksale*. Recuperado el 13 de agosto de 2023, de https://www.stackscale.com/es/blog/openstack/

CLOUDFLARE. (s.f.). *Glossary: What is Platform-as-a-Service (PaaS)?* Recuperado el 1 de 9 de 2023, de CLOUDFLARE: https://www.cloudflare.com/learning/serverless/glossary/platform-as-a-service-paas/

Computación en la nube. (5 de septiembre de 2023). Recuperado el 7 de septiembre de 2023, de Wikipedia: https://es.wikipedia.org/wiki/Computaci%C3%B3n_en_la_nube

Desde Linux. (8 de junio de 2023). Recuperado el 13 de agosto de 2023, de https://blog.desdelinux.net/virtualizacion-tecnologias-disponibles-2019/

Energy. (17 de febrero de 2015). *Qué es la virtualización*. Recuperado el 30 de agosto de 2023, de Evaluando Software: https://www.evaluandosoftware.com/que-es-la-virtualizacion/

Estrin, E. (13 de junio de 2020). *Usando la nube para construir una arquitectura de varias regiones*. Recuperado el 31 de agosto de 2023, de SpanishCloud: https://www.spainclouds.com/blog/usando-la-nube-para-construir-una-arquitectura-de-varias-regiones

Farina, M. (1 de marzo de 2021). *An Overview of Virtual Machines*. Recuperado el 13 de agosto de 2023, de Medium: https://medium.com/reverse-engineering-for-dummies/an-overview-of-virtual-machines-d409fc89ad8f

IONOS. (1 de enero de 2023). *IONOS Digital Guide*. Recuperado el 30 de agosto de 2023, de Virtualización de servidores: definición y funcionalidad: https://www.ionos.mx/digitalguide/servidores/know-how/virtualizacion-de-servidores/

Kodagoda, K. (10 de abril de 2017). *Design Patterns for Cloud: Federated Identity Pattern*. Recuperado el 3 de septiembre de 2023, de Medium: https://kasunkodagoda.medium.com/design-patterns-for-cloud-federated-identity-pattern-a8e0b2a24dcb

Kumar, R., Gupta, N., Charu, S., Jain, K., & Jangir, S. K. (mayo de 2014). Open Source Solution for Cloud Computing Platform Using OpenStack., (pág. 5). Jaipur. doi:10.13140/2.1.1695.9043

Log in to the dashboard. (9 de Agosto de 2019). Recuperado el 7 de septiembre de 2023, de Openstack: https://docs.openstack.org/horizon/latest/user/log-in.html

Malaiya, R. (16 de junio de 2012). *Cloud Computing Seminar*. Recuperado el 13 de agosto de 2023, de https://cloudcomputingseminar.wordpress.com/2012/06/16/unit-iii-cluster-computing/

Marinos, A., & Briscoe, G. (2009). Community Cloud Computing. En A. Marinos, & G. Briscoe, *Cloud Computing* (págs. 472-484). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-10665-1_43

Marrich, A. (16 de septiembre de 2021). *How OpenStack's Keystone handles authentication and authorization*. Recuperado el 3 de septiembre de 2023, de RedHat: https://www.redhat.com/sysadmin/keystone-identity-openstack

Medrano, A. (7 de marzo de 2023). *Cloud computing / computación en la nube. Conceptos, servicios, arquitecturas, plataformas e infraestructuras*. Recuperado el 14 de septiembre de 2023, de Inteligencia Artificial y Big Data: https://aitor-medrano.github.io/iabd2223/cloud/01cloud.html

Microsoft. (s.f.). *Federated Identity pattern*. Recuperado el 4 de septiembre de 2023, de Microsoft: https://learn.microsoft.com/en-us/azure/architecture/patterns/federated-identity

Microsoft. (s.f.). *What is a public cloud?* Recuperado el 31 de agosto de 2023, de Azure: https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-a-public-cloud

Networking architecture. (27 de Julio de 2023). Recuperado el 13 de agosto de 2023, de OpenStack: https://docs.openstack.org/security-guide/networking/architecture.html

Nordhoff, A. (13 de Julio de 2020). *Capital One*. Recuperado el 13 de agosto de 2023, de https://www.capitalone.com/tech/cloud/what-is-a-cluster/

Nova System Architecture. (25 de septiembre de 2022). Recuperado el 13 de agosto de 2023, de OpenStack: https://docs.openstack.org/nova/2023.1/admin/architecture.html

Obasuyi, G. C., & Sari, A. (17 de Julio de 2015). *Security Challenges of Virtualization Hypervisors in Virtualized Hardware Environment.* Recuperado el 13 de agosto de 2023, de https://www.scirp.org/pdf/IJCNS_2015071715070809.pdf

OpenMetal. (26 de julio de 2023). *glossary: what-is-openstack-keystone*. Recuperado el 2 de septiembre de 2023, de openmetal: https://openmetal.io/docs/glossary/what-is-openstack-keystone/

OpenStack. (04 de 04 de 2013). *ReleaseNotes/Grizzly*. Obtenido de https://wiki.openstack.org/wiki/ReleaseNotes/Grizzly?_ga

openstack. (25 de julio de 2023). *OpenStack Documentation: Domains*. Recuperado el 15 de septiembre de 2023, de openstack: https://docs.openstack.org/security-guide/identity/domains.html

OpenStack Docs. (03 de junio de 2020). Recuperado el 30 de agosto de 2023, de Enterprise requirements: https://docs.openstack.org/arch-design/arch-requirements/arch-requirements-enterprise.html

RedHat. (3 de septiembre de 2023). *Chapter 7. Configuring instance scheduling and placement*. Obtenido de RedHat Customer Portal: https://access.redhat.com/documentation/es-

es/red_hat_openstack_platform/16.1/html/configuring_the_compute_service_for_instance_creation/assembly_configuring-instance-scheduling-and-placement_scheduling-and-placement

RedHat. (5 de septiembre de 2023). *Nube privada: ¿qué es y cómo funciona?* Recuperado el 6 de septiembre de 2023, de RedHat: https://www.redhat.com/es/topics/cloud-computing/what-is-private-cloud

RedHat. (s.f.). *Chapter 3. Region and Zones*. Recuperado el 30 de agosto de 2023, de RedHat Costumer Support: https://access.redhat.com/documentation/es-es/reference_architectures/2017/html/deploying_cloudforms_at_scale/regions_and_zones

Rodríguez, H. (24 de marzo de 2021). *Crehana*. Recuperado el 13 de agosto de 2023, de https://www.crehana.com/blog/transformacion-digital/que-es-openstack/

Role based access control (RBAC). (7 de agosto de 2023). Recuperado el 14 de septiembre de 2023, de Digital Guide IONOS: https://www.ionos.es/digitalguide/servidores/seguridad/que-es-el-role-based-access-control-rbac/

Role-based access control - definition & overview. (s.f.). Recuperado el 14 de septiembre de 2023, de Sumo Logic: https://www.sumologic.com/glossary/role-based-access-control/

Shrivastwa, A., Sarat, S., Jackson, K., Bunch, C., Sigler, E., & Campbell, T. (2016). *OpenStack: Building a Cloud Enviroment.* Birmingham, UK: Packt Publishing Ltd. Recuperado el 13 de Agosto de 2023

Vanecek, S. (9 de marzo de 2017). *Blog: Domains in OpenStack.* Recuperado el 16 de septiembre de 2023, de CLOUD&HEAT: https://www.cloudandheat.com/en/domains-in-openstack/

Vennam, S. (s.f.). *¿Qué es la computación en la nube?* Recuperado el 15 de agosto de 2023, de IBM: https://www.ibm.com/es-es/topics/cloud-computing

vmware. (s.f.). *¿Qué es una nube pública?* Recuperado el 31 de agosto de 2023, de vmware: https://www.vmware.com/topics/glossary/content/public-cloud.html

What is user provisioning? (13 de marzo de 2023). Recuperado el 5 de septiembre de 2023, de SailPiont: https://www.sailpoint.com/identity-library/what-is-user-account-provisioning/

Zettler, K. (1 de 9 de 2023). *Microservicios: Plataforma como servicio*. Obtenido de ATTLASIAN: https://www.atlassian.com/es/microservices/cloud-computing/platform-as-a-service

Zhang, M. (10 de abril de 2023). *Data Centers: Cloud Regions and Availability Zones: Explained*. Recuperado el 15 de septiembre de 2023, de Dgtl Infra: https://dgtlinfra.com/cloud-regions-availability-zones/

12 Anexos

- Anexo 1: Entrevista semiestructurada
- Anexo 2: Encuesta
- Anexo 3: Transcripción de entrevista
- Anexo 4: Documento HLD
- Anexo 5: Documento LLD

Anexo 1: Entrevista semiestructurada

Universidad de El Salvador

Facultad de Ingeniería y Arquitectura

Escuela de Ingeniería de Sistemas Informáticos

Trabajo final del Curso de Especialización en Infraestructura en la Nube

PROTOTIPO DE INFRAESTRUCTURA DE NUBE COMUNITARIA MULTI-REGIÓN ORIENTADA A PROPORCIONAR SERVICIOS FUNDAMENTALES

Guía de entrevista

Dirigida al *CEO de Next-Latam*

Objetivo:

Obtener información relevante sobre aspectos generales y puntuales de las tecnologías de computación en la nube, así como el impacto positivo o negativo de estas tecnologías, su implementación y aspectos esenciales desde la experiencia del uso y provisión de infraestructura Cloud.

Preguntas:

1. ¿Cómo describiría el papel actual de la nube en *Next-Latam*?
2. ¿Utiliza soluciones de Nube pública como *AWS, Azure, Google Cloud*?
3. ¿Ha considerado *NEXT LATAM* la posibilidad de migrar o expandir a una infraestructura de nube privada o híbrida?
4. ¿Ha pensado en tener sus propios servidores físicos?
5. ¿Cómo ve usted el futuro de la computación en la nube en el contexto de la transformación digital?
6. ¿Existen planes para expandir o mejorar la infraestructura y servicios basados en la nube?
7. ¿Qué desafíos y oportunidades ha presentado la adopción de la nube?
8. ¿Qué importancia tiene la nube en la estrategia global de la empresa?

Anexo 2: Encuesta

1. ¿Posee conocimientos sobre computación en la nube?
    - Si
    - No
2. ¿Cuál es tu nivel de experiencia en el campo de la infraestructura y computación en la nube?
    - Ninguno
    - Principiante
    - Intermedio
    - Avanzado
3. ¿Cuál es tu rol profesional relacionado con la infraestructura y la computación en la nube?
    - Administrador de sistemas
    - Desarrollador de aplicaciones
    - Arquitecto de soluciones
    - Otro
4. ¿Qué servicios de infraestructura en la nube utilizas o has utilizado en tu trabajo?
    - Amazon Web Services (AWS)
    - Microsoft Azure
    - Google Cloud
    - Digital Ocean
5. ¿Consideras que las nubes públicas mencionadas anteriormente cumplen con la estricta seguridad e integridad de los datos que manejan?
    - Si
    - No

6. ¿Qué factores consideras más importantes al seleccionar un proveedor de servicios en la nube?

- Costo
- Rendimiento
- Seguridad
- Facilidad de uso
- Soporte técnico

7. ¿Cuáles son tus principales fuentes de información y recursos para mantenerte actualizado sobre las últimas tendencias en infraestructura y computación en la nube?

- Blogs y sitios web especializados
- Libros y publicaciones técnicas
- Cursos en línea y capacitación
- Redes sociales
- Conferencias
- Podcasts

8. ¿Considera necesario la adopción de una infraestructura propia de nube para su organización?

- Si
- No

9. Considera importante tener el control total de una infraestructura en la nube en cuanto a las necesidades de la empresa en la cual trabaja

- Si
- No

10. La empresa para la cual labora, hace uso de:

- Nube Pública
- Nube Híbrida
- Nube privada
- No sé

11. ¿Cuál piensa que es la mejor opción de servicios de computación en la nube para la empresa que labora?

- Nube Pública
- Nube Híbrida
- Nube privada

12. ¿Qué nivel de importancia le da la empresa en la cual labora a las tecnologías actuales?

- Poco
- Medio
- Mucha
- Está siempre a la vanguardia tecnológica

13. Según su criterio, la nube será predominante en el futuro o ya lo es hoy en día

- Será predominante en el futuro
- Ya es predominante hoy en día
- Tal vez

14. Considera necesaria e importante la formación en temas de computación en la nube y temas relacionados, por parte de las empresas u organizaciones para los empleados de TI

- Si
- No

15. ¿Has escuchado o posees conocimientos sobre *OpenStack*?

- Si
- No

16. ¿Has integrado *OpenStack* con otras tecnologías o herramientas en tu infraestructura en la nube?

- Si
- No

17. Si la respuesta a la pregunta anterior es sí, ¿Cuáles tecnologías y herramientas has integrado con *Openstack*?

- Kubernetes
- Ansible
- Docker
- Ninguna de las anteriores

18. De los siguientes componentes de *Openstack* con cuales estás familiarizado

- Nova
- Neutron
- Cinder
- Keystone
- Glance
- Swift
- Horizon
- Ninguno

19. La curva de aprendizaje sobre *Openstack* consideras que es impedimento para utilización de esta plataforma

- Si
- No

20. ¿Tienes alguna recomendación o consejo para aquellos que estén considerando la implementación de *OpenStack* en su infraestructura en la nube?

Anexo 3: Transcripción de Entrevista

Entrevista a CEO de *Next-Latam* S.A. de C.V
Entrevista a: Ing. Alexis Rojas
Realizada: 19 de octubre de 2023
Hora de inicio: 16:24
Duración: 38:20 minutos
Lugar: Sala de videoconferencias Google Meet

Daniel: Buenas tardes, dando inicio con esta entrevista, ¿Usted como *CEO de Next-Latam* cómo describiría el papel actual de los beneficios y bondades de la nube en Next-Latam?
Alexis: *Next-Latam* principalmente es una empresa o sea digamos que tiene operaciones tanto en El Salvador como ya en Panamá donde el rubro principal es el desarrollo de software, estamos enfocados y orientados principalmente a software como servicio. Con el tema de infraestructura en el cual de hecho le proveemos infraestructura a lo que son ya como cinco o seis empresas, básicamente infraestructura en la nube y en este sentido pues creo que ha sido bastante interesante los beneficios que ha traído la nube digamos al mercado local creo que todavía hay una barrera digamos cultural o sea lo cual es un poco fuerte porque la gente siempre sigue creyendo de que si lo tengo allí físicamente el equipo pues garantizo de que la seguridad de la información. La nube nos ha demostrado pues que tienen mejores costos, mejor performance, mejor escalabilidad, mejor seguridad, etcétera.
Daniel: Bastante interesante, los servicios que brinda como infraestructura en la nube, ¿Utiliza soluciones que ya existen como *AWS, Google Cloud, Azure* o algún otro?

Alexis: Fíjate que nosotros principalmente ocupamos Bolt Cloud, digamos nos ha funcionado súper bien además de eso, logramos hacer una alianza con ellos en España con la representación de España y nos da costos mucho más baratos. En ese sentido este ha sido súper estable o sea nos han resuelto con un soporte mucho más rápido.

Daniel: ¿Ha considerado para *Next-Latam* la posibilidad de migrar o expandir una infraestructura de nube o ha pensado tener sus propios servidores o los tuvo en su momento?

Alexis: Fíjate que los tuvimos o sea digamos inicialmente ya pero como digo o sea en realidad tener nuestros propios servidores nuestros, servidores de forma local o sea no era sostenible o sea en realidad los costos de infraestructuras son demasiado altos y el beneficio pues en realidad era poco versus la nube, por eso fue que nos migramos directamente a la nube e hicimos ese traspaso, fuimos probando distintos servicios.

Daniel: En cuanto a la seguridad del servicio que usted utiliza para la computación en la nube, ¿Qué podría mencionar que le ha gustado más de ese tema de seguridad o porque eligió *Bolt* en comparación de los demás en cuanto a seguridad?

Alexis: Primero lo elegimos por varios factores uno escalabilidad, segundo este lo que tiene que ver con precios que es un factor super fuerte, tercero por lo que es la infraestructura de servicios que ofrece ya que es bastante robusta y cuarto pues obviamente por la seguridad o sea digamos la ventaja de Bolt es de que ya nosotros nos da por ejemplo un firewall que ya tiene como las reglas generales y básicas preconfiguradas, la cual pues simplemente tengo que meter los equipos que tenga, digamos o los servicios que tengan ejecución los meto al firewall y a ese ya tiene como una capa de seguridad preconfigurada y no tengo que hacer nada o sea simplemente lo asigno, por el otro lado también lo que es la asignación de lo que son los DNS él ocupa digamos un proxy que se basa como básicamente como un principio como de Cloudflare o sea digamos este para filtrar muchas peticiones.

Daniel: ¿Cómo ve usted el futuro de la computación en la nube en el contexto de la transformación digital?

Alexis: Es algo que se va a dar tarde o temprano o sea digamos va a llegar con fuerza y es algo de lo que en uno no se puede escapar definitivamente, o sea, esta es una ola y entre yo pues más tiempo me tarde en subirla más atrasado estaré. Este sentido es un tema cultural que poco a poco tiene que irse impulsando sobre todo para los nuevos informáticos ya no es un temor. Como digo es tema cultural, poco a poco se va a ir dando, es completamente necesario abarata costos, nos permite tener implementaciones rápidas, nos permite tener mucho más esquema de seguridad este y pues y nos facilita el trabajo.

Daniel: Ya para finalizar, en cuanto a *Next-Latam* nos comentaba que ya se expandió a Panamá, ¿hay planes para seguir expandiendo y a la vez mejorar los servicios de infraestructura y todo lo que esté basado en la nube para *Next-Latam*?

Alexis: Sí definitivamente o sea digamos hemos estado trabajando en el mercado salvadoreño ya por cuatro años, ya este año abrimos ya una oficina en Panamá y vamos a comenzar operaciones ahí en Panamá este es un mercado que nos ha gustado bastante porque son un mercado mucho más sofisticado, digamos así donde ir a facilitar u ofrecer por ejemplo esquemas de infraestructura en la nube ya lo entienden, ya lo comprenden, eso es su día a día. Ellos ya lo dan por sentado y de hecho si les llegara con una infraestructura local, para ellos es así como ¿por qué me estás ofreciendo esto? no tiene mucho sentido. Y por eso se ha escogido Panamá para establecer una oficina de Next-Latam ahí.

Daniel: Muchas gracias por su tiempo y por atender a nuestra entrevista.

Alexis: Estamos para servir, gracias por tomarme en cuenta.

UNIVERSIDAD DE EL SALVADOR

FACULTAD DE INGENIERIA Y ARQUITECTURA

DOCUMENTO DE DISEÑO DE ALTO NIVEL

HIGH LEVEL DESIGN DOCUMENT

PROTOTIPO DE INFRAESTRUCTURA DE NUBE COMUNITARIA MULTI-REGION ORIENTADA A PROPORCIONAR SERVICIOS FUNDAMENTALES

NOVIEMBRE 2023

VERSIÓN 3.0 12/NOV/2023

Contenido

Introducción

Este documento presenta un prototipo de infraestructura de nube que sirve como región central administrativa de otras regiones que se encuentran geo distribuidas las cuales forman una nube comunitaria en la cual se comparten recursos y al mismo tiempo hacen uso de los servicios bridados por la infraestructura.

El prototipo se basa en la tecnología de OpenStack que es una combinación de herramientas de código abierto, altamente conocido en el mundo del cloud computing. Sin embargo, esta implementación se centra únicamente en los componentes de Keystone para la autenticación de los usuarios y la puesta de los puntos de acceso, el otro componente es el panel de administración de OpenStack llamado Horizon.

La implementación que se presenta en esta solución no solo ofrece servicios fundamentales para el funcionamiento de las regiones, sino que también ofrece la capacidad de almacenamiento con Ceph, para el manejo en bloque, almacenamiento de objetos, almacenamiento de archivos, etc.

Luego de presentar el detalle en alto nivel en este documento, se profundizará en los detalles técnicos de la solución presentada, así como la guía para implementar el prototipo siguiendo los pasos que llevaron a consecución de la infraestructura.

Objetivos

- Definir la visión global de la infraestructura, destacando su importancia y los beneficios que aporta a las organizaciones.
- Describir los componentes y los servicios clave. Explicar cómo se integran para poder proporcionar una solución coherente.
- Explorar la arquitectura del prototipo de alto nivel, describiendo e identificando sus componentes. También se incluye en este apartado una visión de la topología de red utilizada.

Audiencia Objetivo

Este documento está dirigido a audiencia ejecutiva y técnica en las que se pueden incluir:

Ejecutivos de Organizaciones que necesiten comprender la visión estratégica y beneficios tanto logísticos como comerciales de una infraestructura de nube multirregión.

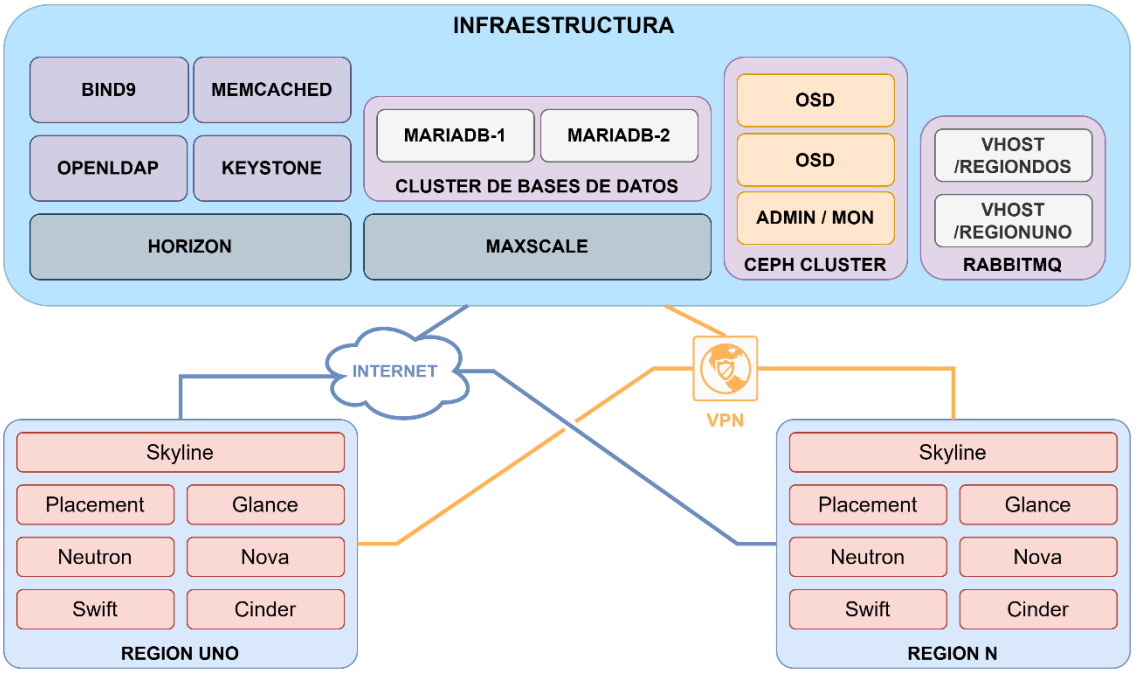
Arquitectos de soluciones que deseen estudiar los puntos clave de la infraestructura y dar su opinión en cuando decisiones de diseño.

Gerentes de proyectos involucrados en planificación e implementación de infraestructuras de nube que deseen comprender los alcances y objetivos del proyecto.

Equipos de TI y Operación responsables de la administración y mantenimiento de infraestructuras de nube tanto hibridas, como públicas o privadas.

Es importante destacar que este documento se enfoca en brindar una visión general de a un alto nivel, posteriormente se presenta un documento de bajo nivel que profundiza más detalles técnicos.

Arquitectura del Prototipo



Componentes de la Infraestructura

La infraestructura se compone por diversos servicios que serán consumidos por las múltiples regiones que se conecten a ésta.

Horizon: Es el panel de control de OpenStack basado en la web y escrito en Python que permite la administración de las regiones conectadas a la infraestructura.

Keystone: Es el servicio de identidad que usa tanto Horizon como los demás servicios de las regiones para la autenticación y autorización dentro de la nube.

OpenLDAP: Es un servicio de implementación de LDAP el cual proporciona servicio de directorio de usuarios, grupos y otros recursos. Es usado por Keystone para la autenticación de usuarios.

Bind9: Es el servicio DNS para toda la nube, el cual permite traducir nombres de dominios amigables para las personas a direcciones IP y viceversa, para un mejor manejo de las conexiones dentro de la nube.

Memcached: Es un sistema de almacenamiento en memoria caché de objetos de alto rendimiento, usado por las regiones para almacenar datos efímeros como tokens.

RabbitMQ: Es el servicio de cola de mensajes que facilita coordinar las operaciones y la información de estado entre los servicios de openstack. Se configura vhost o entidades virtuales de colas para separar los mensajes entre las regiones.

Clúster de Bases de Datos: Es un sistema de bases de datos que trabajan juntas y se sincronizan los datos para garantizar alta disponibilidad. Estos servidores corren el servidor de bases de datos mariadb y es usado por la infraestructura para guardar la información de los componentes, pero también para ofrecer el servicio de bases de datos a las regiones conectadas a la nube.

Maxscale: Actúa como un proxy de base de datos o balanceador de la carga que va hacia el clúster de base de datos. Es que punto donde se comunican los componentes de la infraestructura o las regiones.

Clúster de Ceph: Es un sistema de almacenamiento distribuido de bloques, objetos y/o archivos de alta disponibilidad y redundancia de estos. Los nodos ADMIN (Administrador) y MON (Monitoreo) son los encargados de administrar el clúster, mientras que los nodos OSD son los que tienen los Daemon de almacenamiento de datos y son los responsables de almacenar los datos y replicarlos entre ellos.

Internet: Es la red WAN (Wide Area Network) que permite comunicación a todo el mundo de manera pública. Por esta red se accede a los servicios públicos como el panel o permite conexiones VCN, entre otros.

VPN: Virtual Private Network o Red Privada Virtual es una red que va sobre internet pero que permite comunicación de forma privada y segura entre los nodos que tengan la conexión en esa red.

En resumen, la Infraestructura contiene componentes clave que respaldan y permiten a regiones geo distribuidas compartir y escalar los recursos.

Componentes de las Regiones

Para las regiones de prueba en esta infraestructura, se han configurado los servicios básicos de un despliegue de OpenStack.

Horizon: Al igual que la infraestructura, una región puede tener su propio panel de control de OpenStack. Para el caso de estudio, en las regiones se estaría apuntando al OpenStack de la infraestructura para la autenticación.

Placement: Es el responsable de hacer seguimiento de inventarios y usos de los proveedores de recursos, como nodo informático, grupo de almacenamiento compartido, entre otros.

Glance: Es el servicio de imágenes de maquias virtuales, hace el registro, descubrimiento y recuperación de estas máquinas. Puede conectarse al servicio de almacenamiento de Ceph para almacenar las imágenes en un pool de bloques u objetos.

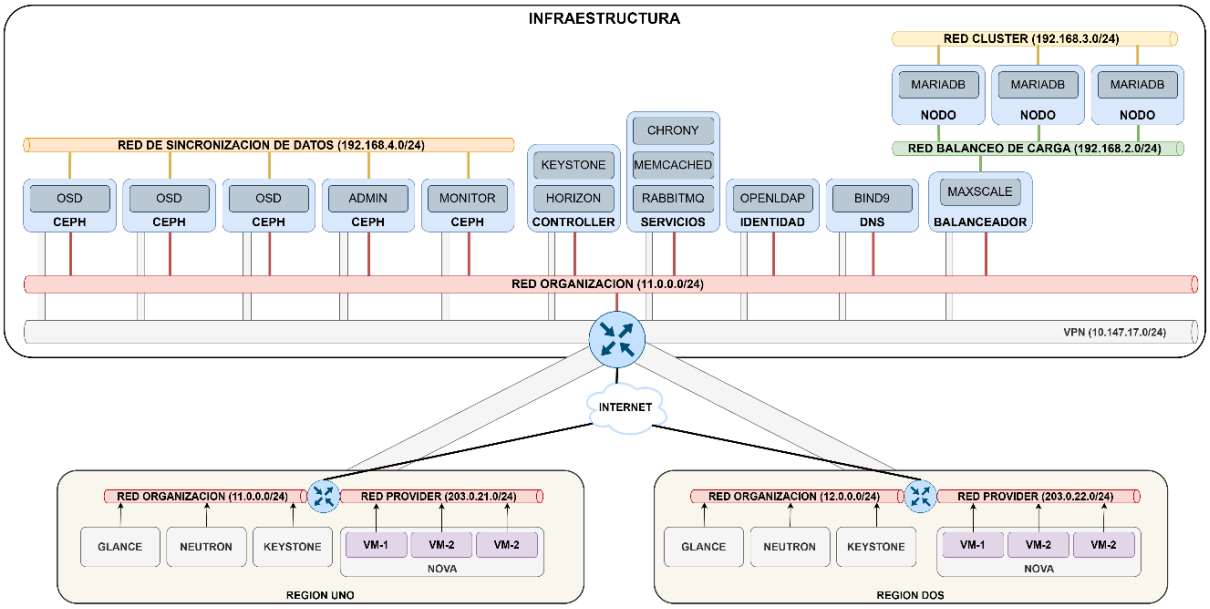
Neutron: Es el componente encargado de gestionar y orquestar redes SDN y ofrecer el NaaS (redes como servicio) para entornos informáticos virtuales.

Nova: Se encarga de aprovisionar y gestionar recursos informáticos de cómputo, como máquinas virtuales, bare-metal o contenedores. Puede usar el componente de Cinder para almacenar de forma más optima las unidades de discos virtuales como bloques.

Cinder: Es un servicio de almacenamiento en bloques, el cual administra la creación y montaje de volúmenes de almacenamiento persistentes. Provee una API que permite solicitar y consumir los volúmenes sin la necesidad de saber la localización de estos, por eso es posible conectarlo con el clúster de almacenamiento de Ceph de la infraestructura.

Swift: Es el sistema de almacenamiento de objetos en openstack, el cual permite la creación, montaje y gestión de volúmenes de almacenamiento persistente. Es un componente que permite almacenar de forma escalable y optimizada datos no estructurados. También puede consumir el servicio de Ceph para usar un pool de almacenamiento de objetos.

Topología de Red de la Infraestructura.



La topología de red de la infraestructura está dividida en múltiples redes y segmentos que agrupan componentes específicos.

Red de Sincronización de Datos: Es la red por la cual se comunican todos los nodos del clúster de Ceph. Es la red por la cual se sincronizan los datos en todos los OSD para la redundancia de los datos. Los nodos de administración y monitoreo son a los cuales se acceden los datos mediante la red de la organización y también a través de la VPN para que las regiones consuman el servicio de almacenamiento.

Red Clúster de Base de Datos: Mediante esta red privada los nodos de mariadb que estén configurados en el clúster de galera sincronizan los registros mientras que reciben peticiones de lectura y escritura simultanea desde el nodo de maxscale.

Red de Balanceo de Carga: Es la red privada específica para la comunicación entre el balanceador de carga Maxscale y los nodos en Clúster de Mariadb.

Red de la Organización: Es la red que tiene salida a internet y también que permite acceder a algunos servicios públicos, como el panel de administración de OpenStack. Comunica todos los servidores de la infraestructura para una comunicación rápida. Mediante esta red, se consumen los servicios entre servidores de la infraestructura.

VPN: Es la conexión privada y segura que une a las regiones con la infraestructura, para el consumo de los servicios fundamentales y la comunicación y administración de los recursos por parte del servidor main a las regiones.

Descripción de caso

El caso se basa en que existe una infraestructura de nube que brinda servicios fundamentales a otras regiones que están geográficamente distribuidas.

Estas regiones, así como la propia infraestructura necesitan servicios fundamentales como almacenamiento en cache, servicios de base de datos, servicio de colas de mensaje para que los componentes de OpenStack organicen los recursos y procesos.

También se tiene un servicio de almacenamiento distribuido, el cual provee solución a la necesidad de redundancia y alta disponibilidad de la información. Ceph brinda almacenamiento de objetos para los componentes de Swift de las regiones, almacenamiento en bloque para los componentes de Cinder que a su vez es usado por el componente de Nova para organizar los volúmenes de disco virtuales para las máquinas virtuales.

Los usuarios de los servicios de las regiones se autentican usando el componente de Keystone en el nodo principal de la infraestructura (Main). La información de los usuarios esta almacenada en un servidor LDAP (Lightweight Directory Access Protocol o protocolo ligero de acceso a directorios), lo cual permite que se los usuarios formen parte de un sistema de usuarios federados, los cuales usan las mismas credenciales en todas las regiones y servicios disponibles en la nube comunitaria.

Para darle una capa más de seguridad de la información en la comunicación entre las regiones y la infraestructura, se ha configurado un VPN (Red Privada Virtual) la que permite conexión directa con los nodos que proveen los servicios.

Todos los servicios, exceptuando el panel, se exponen mediante endpoint que responden a diferentes puertos y protocolos. Ejemplo de esto son las comunicaciones que viajan sobre TCP como la comunicación HTTP y HTTPS para los paneles de administración de Horizon, Ceph, Maxscale, Rabbit, o las conexiones a la base de datos sobre Maxscale por el puerto 3306. También una combinación de TCP y UDP como el puerto 53 por el DNS.

Las regiones cuentan con una red externa expuesta a internet para darle salida exclusiva a las instancias de máquina virtual de Nova, sin comprometer la seguridad de la red de la organización de cada Región.

UNIVERSIDAD DE EL SALVADOR

FACULTAD DE INGENIERIA Y ARQUITECTURA

DOCUMENTO DE DISEÑO DE BAJO NIVEL

LOW LEVEL DESIGN DOCUMENT

PROTOTIPO DE INFRAESTRUCTURA DE NUBE COMUNITARIA MULTI-REGION ORIENTADA A PROPORCIONAR SERVICIOS FUNDAMENTALES

NOVIEMBRE 2023

VERSION 3.0 12/NOV/2023

Tabla de Contenido

Introducción

El presente documento es una continuación del documento de Diseño de Alto Nivel (*HLD*) previamente presentado. El *HLD* estableció la visión y los componentes clave del prototipo de infraestructura de nube comunitaria geo-distribuida. En este *LLD*, nos adentramos en los detalles técnicos de la solución proporcionando una guía para la implementación del prototipo, siguiendo los pasos que llevaron a su concepción, mostrando las configuraciones necesarias y pruebas de funcionamiento.

Objetivos

- Explicar los pasos que se deben seguir para la construcción del prototipo planteado en el documento *HLD*.
- Demostrar las configuraciones de en archivos y configuraciones internas de cada componente de la infraestructura.

Audiencia Objetivo

Equipos de TI y Operación responsables de la administración y mantenimiento de infraestructuras de nube tanto hibridas, como públicas o privadas.

Entusiastas de la tecnología de computación en la nube interesados en implementar el prototipo presentado.

Configuraciones previas

Instalación de Virt-Manager

La instalación se hace sobre sistema operativo *Ubuntu* 22.04. Se recomienda actualizar todos los paquetes.

```
estudiante@laptop:~$ sudo apt update && sudo apt upgrade -y
```

Se comprueba que el terminal donde se va a instalar *virt-manager* y *kvm* sea compatible.

```
estudiante@laptop:~$ egrep -c '(vmx|svm)' /proc/cpuinfo
16
```

Si la salida es un numero distinto a 0, el terminal es compatible, de lo contrario es incompatible y no se recomienda seguir con la instalación.

Se debe instalar los paquetes de máquina virtual *KVM.*

```
estudiante@laptop:~$ sudo apt install libvirt-clients libvirt-daemon-system libvirt-daemon \
virtinst bridge-utils qemu qemu-kvm
```

Se instala la interfaz gráfica para manejar las máquinas virtuales.

```
estudiante@laptop:~$ sudo apt install virt-manager
```

Se habilita el servicio *libvirt.*

```
estudiante@laptop:~$ sudo systemctl enable libvirtd --now
estudiante@laptop:~$ sudo systemctl start libvirtd
```

Se verifica el estado del servicio.

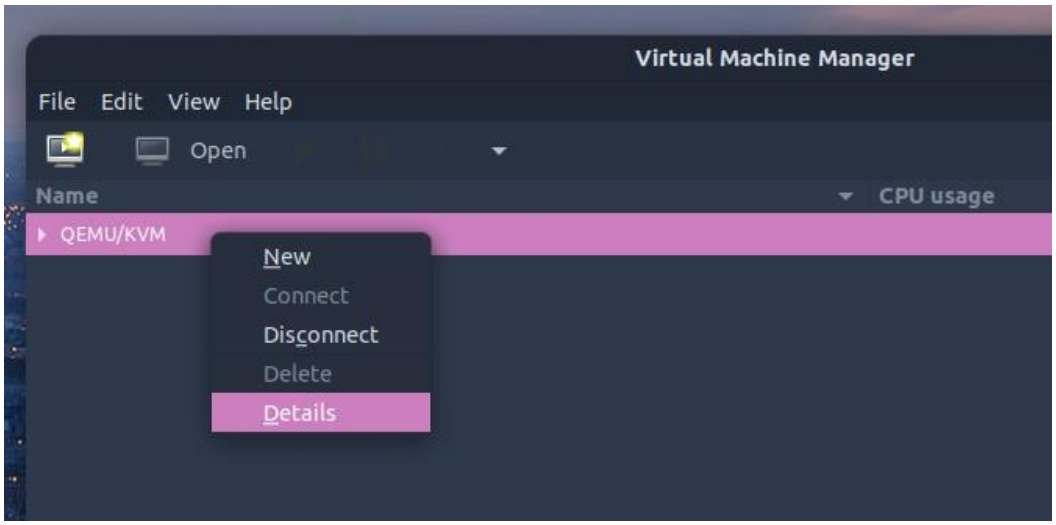
```
estudiante@laptop:~$ sudo systemctl status libvirtd
● libvirtd.service - Virtualization daemon
     Loaded: loaded (/lib/systemd/system/libvirtd.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2023-10-10 08:35:59 CST; 2min 47s ago
TriggeredBy: ● libvirtd.socket
             ● libvirtd-ro.socket
             ● libvirtd-admin.socket
       Docs: man:libvirtd(8)
             https://libvirt.org
   Main PID: 4060 (libvirtd)
      Tasks: 21 (limit: 32768)
     Memory: 10.5M
        CPU: 409ms
     CGroup: /system.slice/libvirtd.service
             ├─4060 /usr/sbin/libvirtd
             ├─4202 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf ...
             └─4203 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf ...
```

Se agrega el usuario actual al grupo *kvm* y *libvirt.*

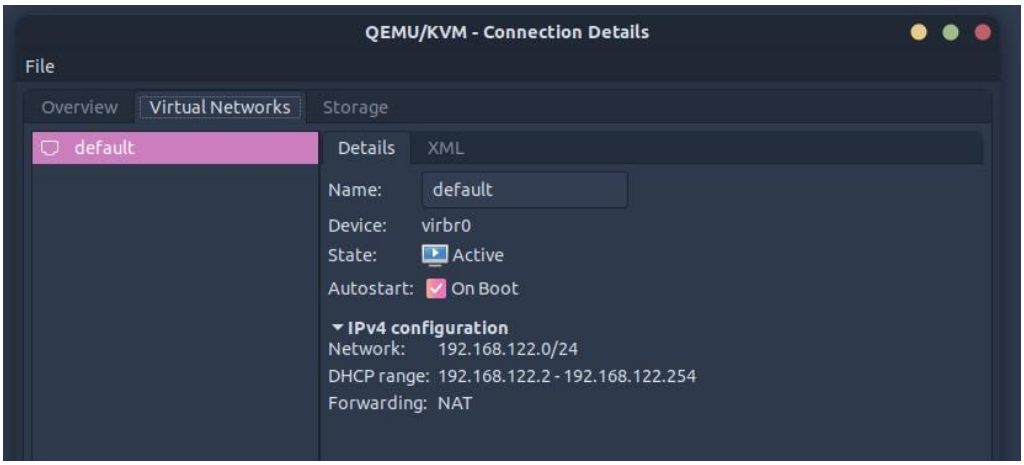
```
estudiante@laptop:~$ sudo usermod -aG kvm $USER
estudiante@laptop:~$ sudo usermod -aG libvirt $USER
```

Creación de Redes Virtuales

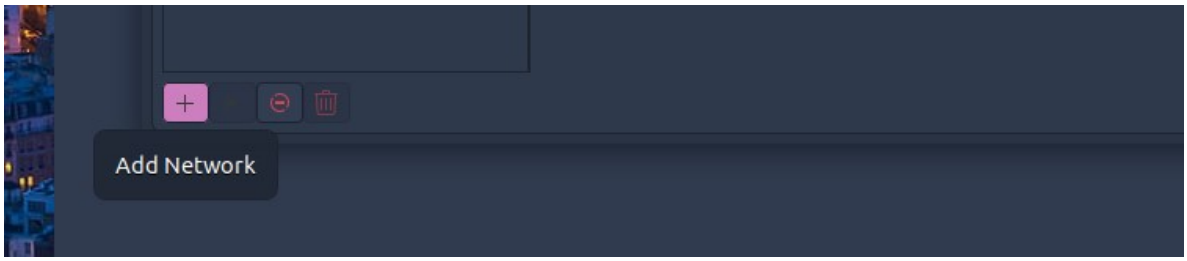
En *virt-manager* se pueden crear múltiples redes para conectar a las máquinas virtuales. Para llevar a cabo la creación de una red nueva, se debe seleccionar la conexión a *QEMU/KVM*, y con *click* derecho se abre un menú donde seleccionamos “detalles”.



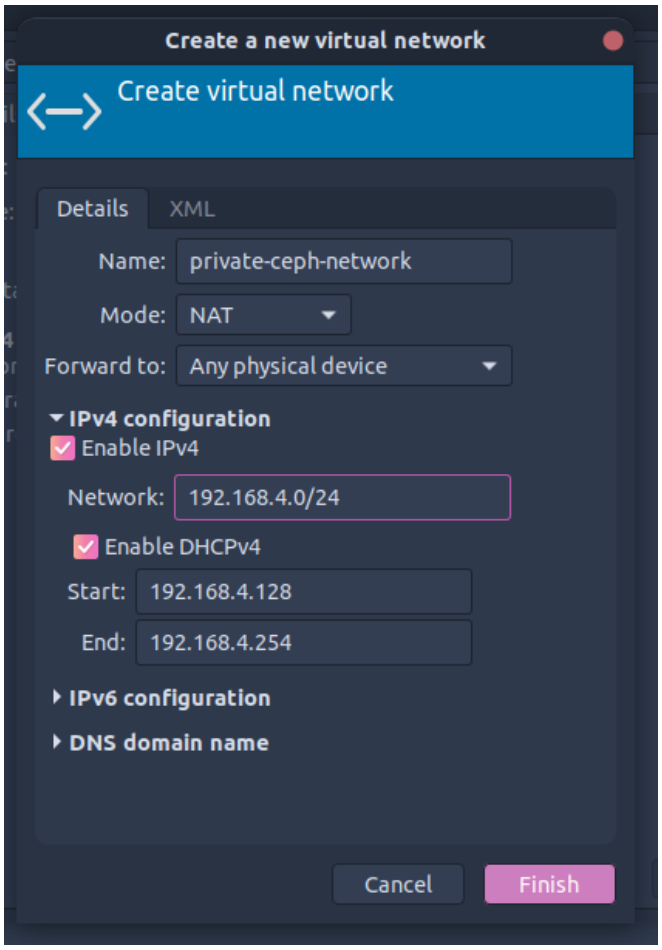
Se abre una ventana con una red creada por defecto.



En la esquina inferior izquierda se encuentran botones para administrar redes, de las cuales debemos seleccionar el icono de “+” o “agregar red”.



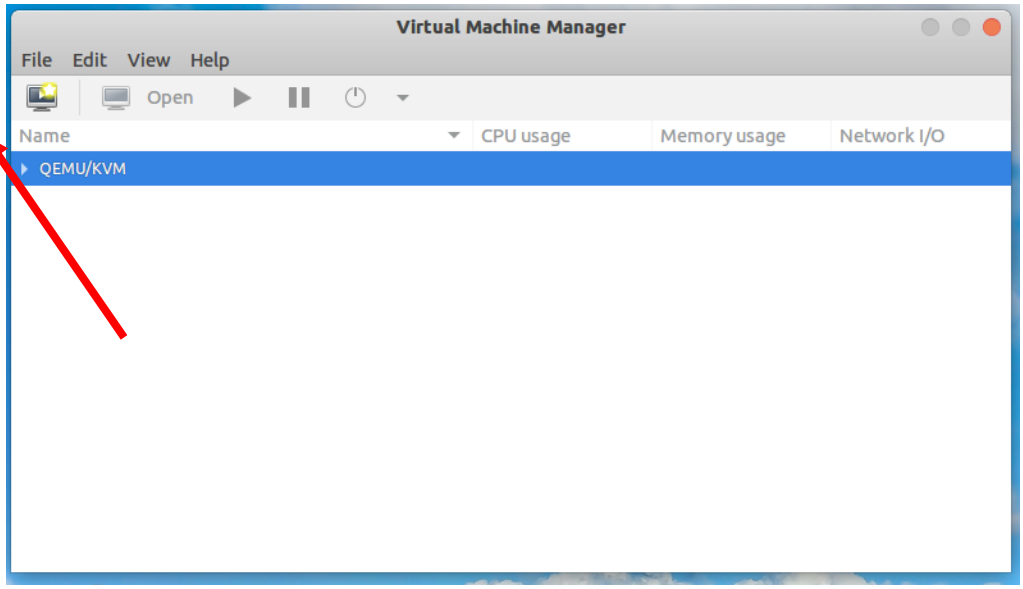
Se abre una nueva ventana donde debemos configurar los parámetros que tendrá la nueva red. Se configura el nombre, el tipo de conexión con la maquina anfitrión, y el tipo de redirección. Además de configurar el protocolo IPv4 como la dirección de red, y permite habilitar un servidor *DHCP* para un cierto rango de direcciones IP configurables.



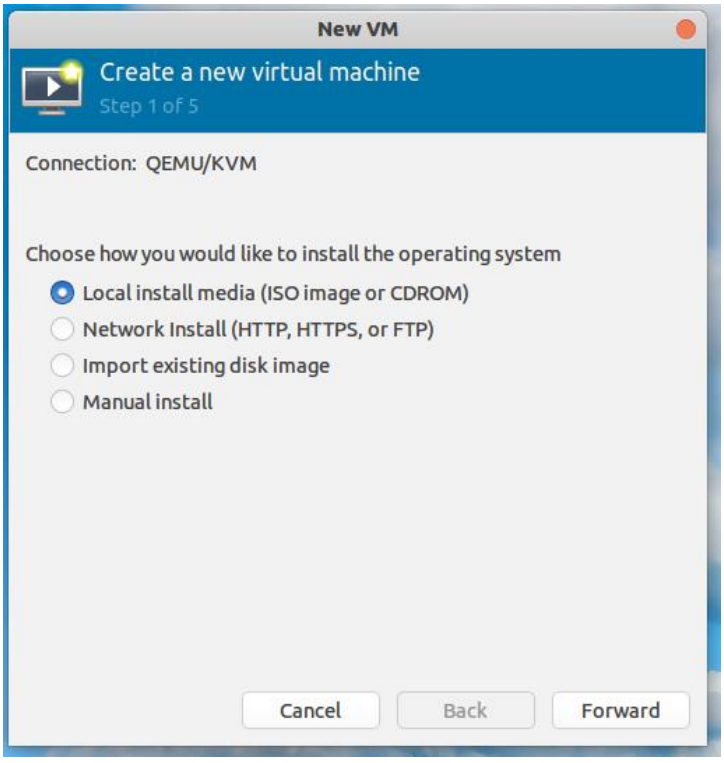
Al finalizar la opción "*Finish*" o "Finalizar" para crear la nueva red con los parámetros ingresados.

Creación de Máquinas Virtuales

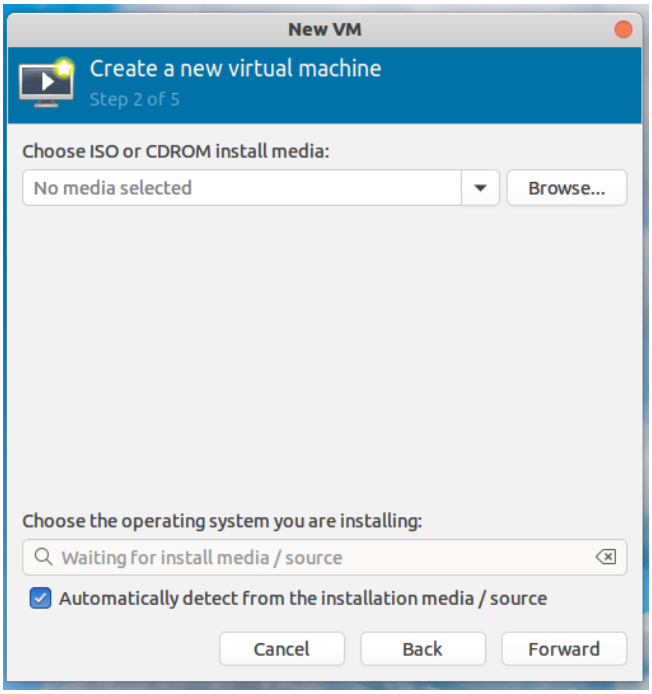
Para la creación de máquinas virtuales se debe escoger el icono de creación de maquina virtuales en la interfaz de *virt-manager*.



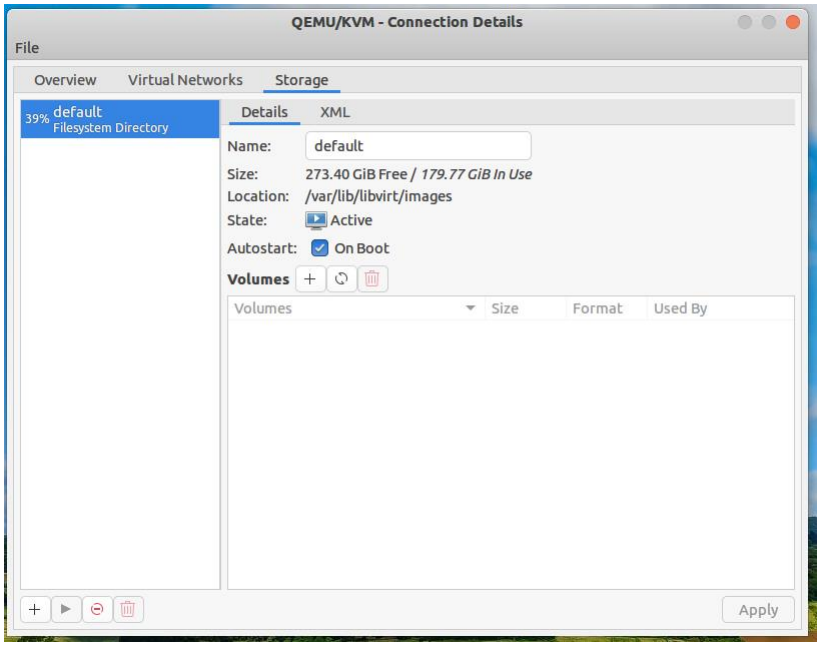
Se abre la ventana de creación de máquina virtual donde se debe indicar la forma de crear la máquina virtual, la forma empleada en este caso es Instalación Local vía imagen *ISO* o *CDROM*.



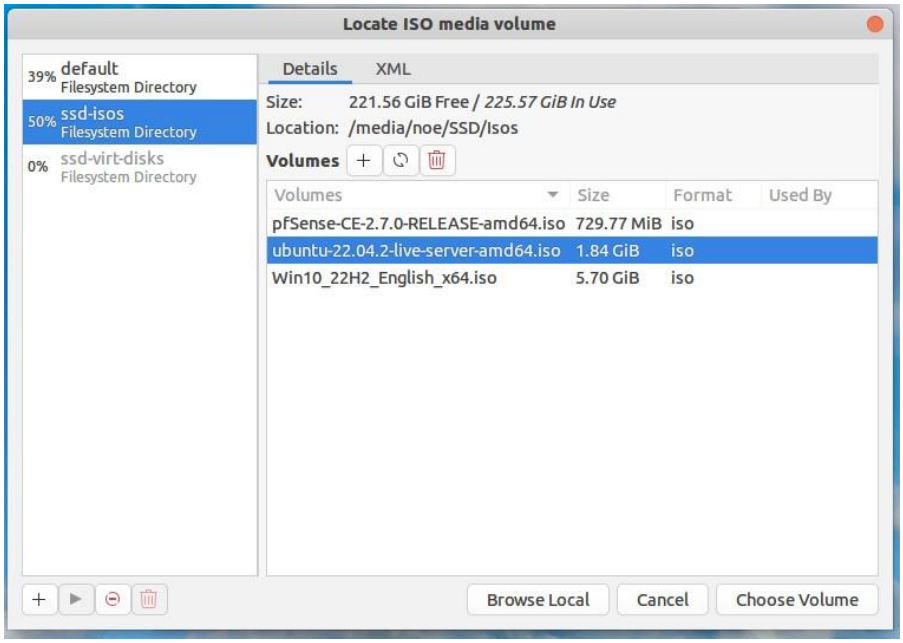
El siguiente paso es escoger el medio de instalación, ya sea unidad de *CD* o un archivo de imagen *ISO*. Ya que la instalación más tradicional es vía *ISO*, seleccionamos el botón de “Buscar” o “*Browse*”.



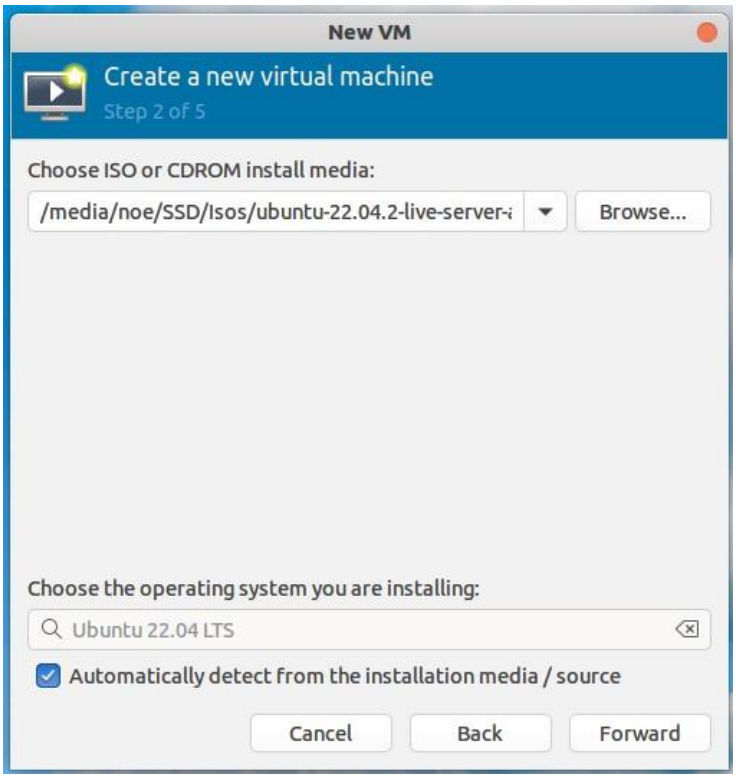
Por defecto hay un directorio ya habilitado que es donde se guardan los archivos de disco virtual, pero se puede agregar directorios donde se recuperen las imágenes de instalación del mismo modo que las redes, mediante el botón de la esquina inferior izquierda.



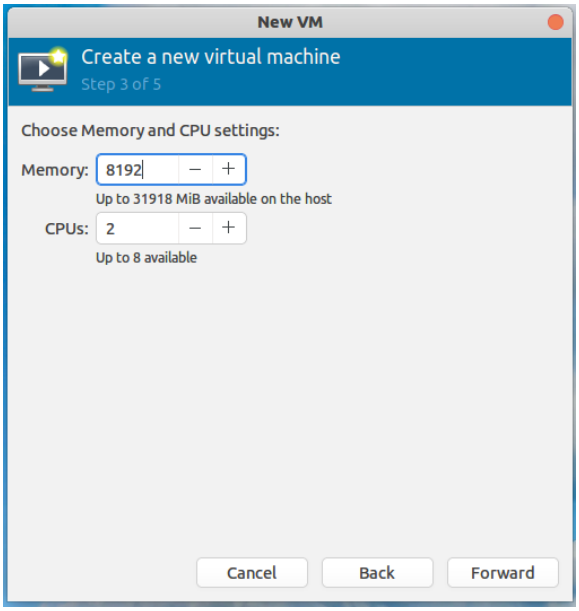
Una vez agregado el directorio donde se encuentra la imagen *ISO* del sistema a instalar, seleccionamos la imagen a usar. En este ejemplo, es la imagen de *Ubuntu Server* 22.04. Cuando ya esté seleccionada se da *click* en el botón “Elegir Volumen” o “*Choose Volume*”.

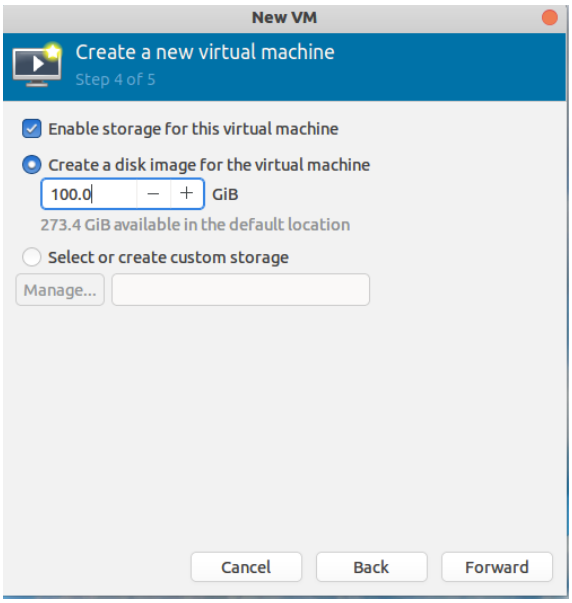


El programa detecta el sistema operativo de la imagen para crear el entorno optimizado.

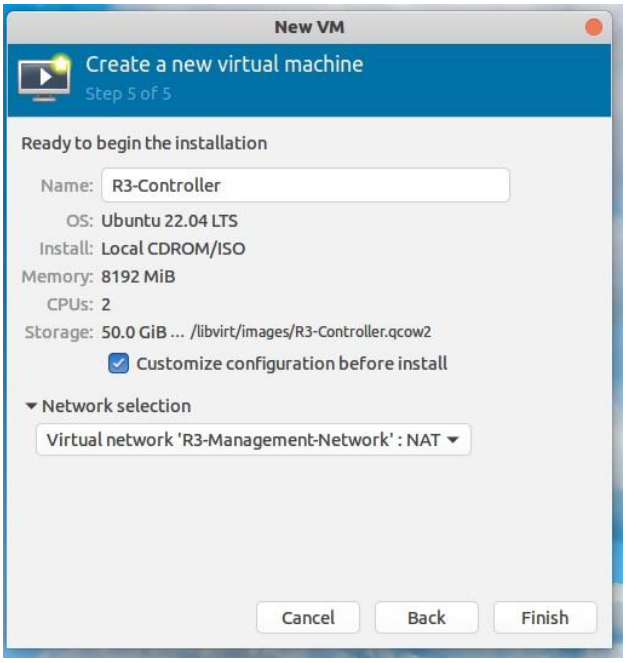


Se siguen las indicaciones para asignar hardware virtual como la memoria RAM o el almacenamiento.

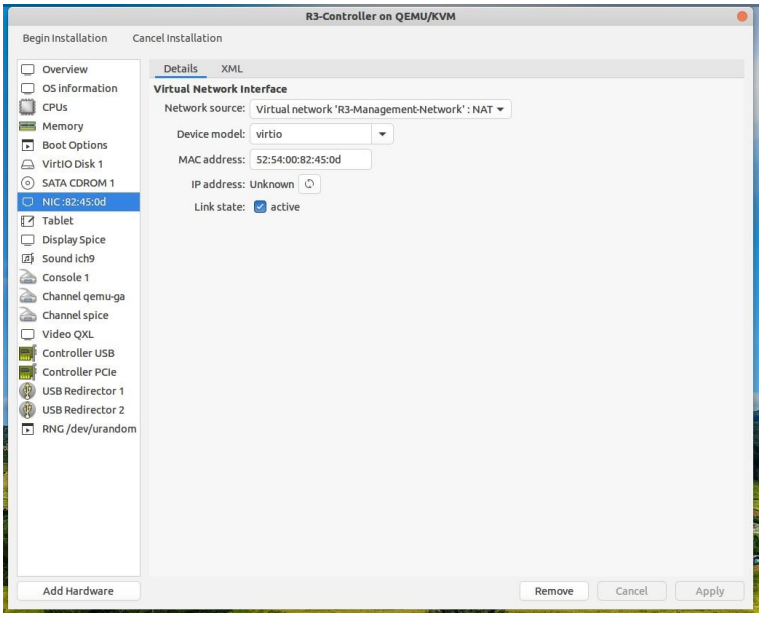




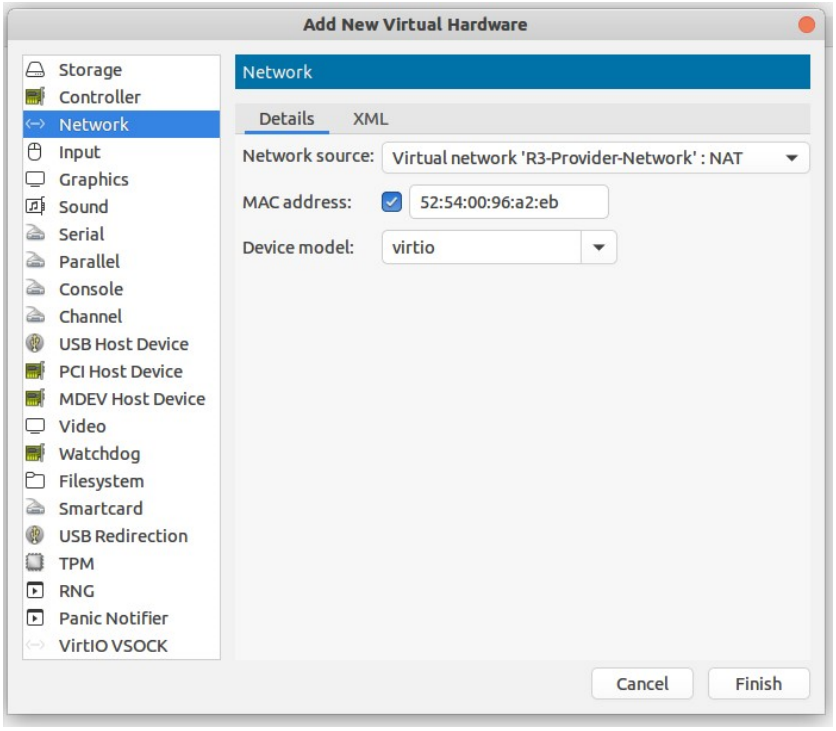
En la última pantalla se configura el nombre de la máquina virtual y la red que tendrá la interfaz por defecto. Se puede seleccionar la opción de configurar más parámetros antes de iniciar la instalación del sistema operativo. En este caso lo seleccionamos para agregarle otra interfaz de red antes de iniciar la instalación.



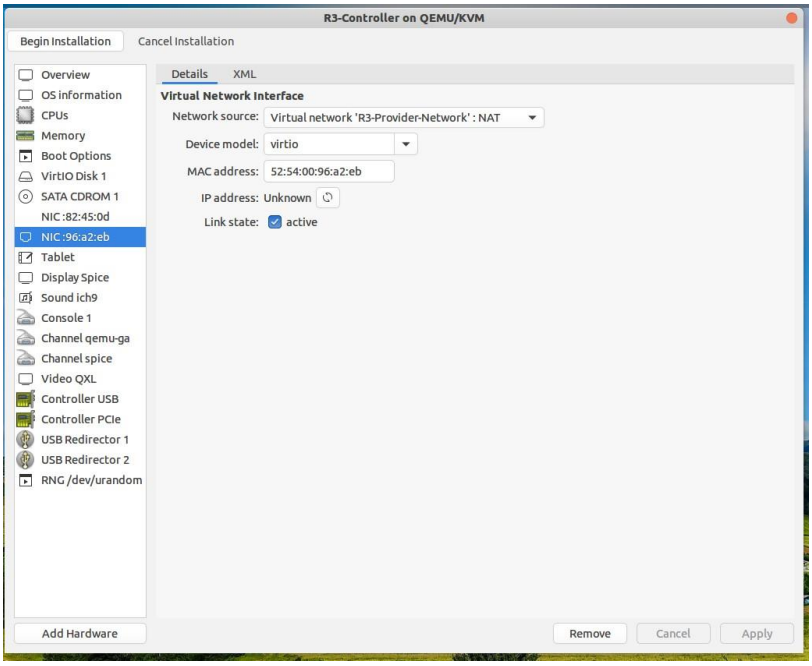
Se abre la ventana de editar la máquina virtual donde aparte de editar las características por defecto, podemos agregar más hardware virtual con el botón de la esquina inferior izquierda.



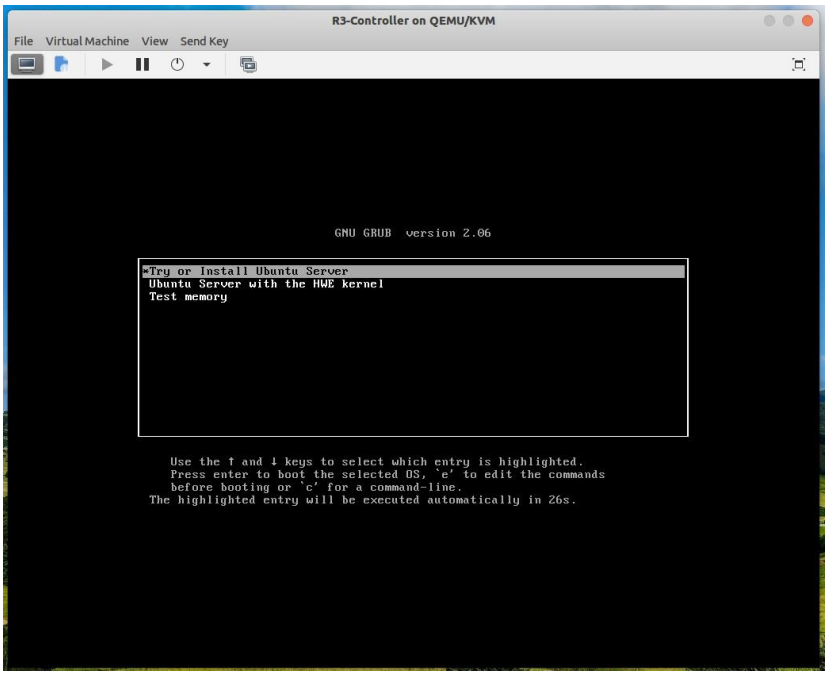
Al “Agregar Hardware” se muestra una nueva ventana para seleccionar el tipo de hardware virtual a agregar y también configurar sus parámetros. En este ejemplo se agrega una nueva “*Network*” o “Red” que crea una nueva interfaz de red en la máquina virtual. Para confirmar se selecciona “*Finish*” o “Finalizar”.



Una vez aplicado el nuevo hardware, se puede ver una nueva interfaz de red o “*NIC*” con una dirección MAC diferente. Para iniciar la instalación se selecciona el botón de la esquina superior izquierda “*Begin Instalation*” o “Iniciar Instalación”.



Se muestra una pantalla en la cual se puede ver la interfaz, ya sea terminal o grafica del sistema operativo a instalar.

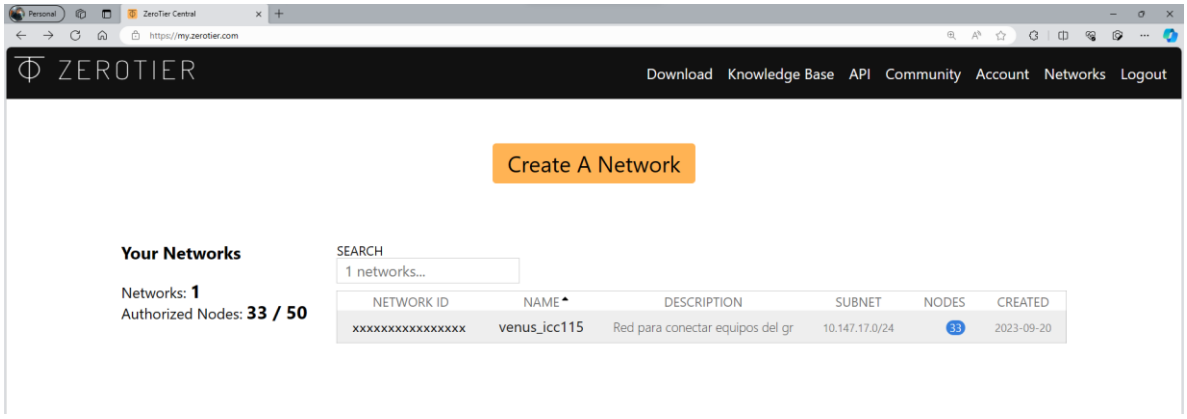


Ya en este punto, los pasos siguientes dependen de cada sistema operativo a instalar.

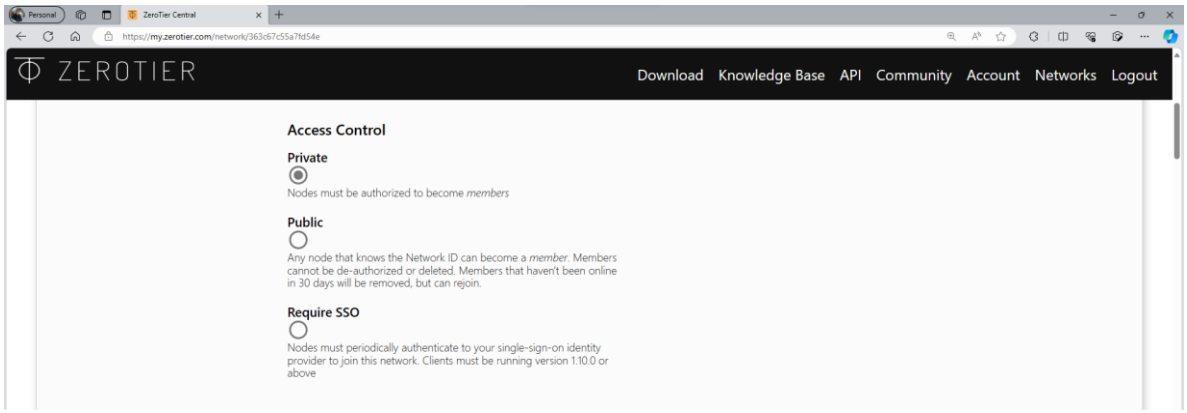
Configuración de la VPN

La *VPN* a configurar es *ZeroTier*, la cual es un servicio de Redes Definidas por Software (*Software Defined Networks – SDN*).

Se ha creado una cuenta en la página web, y automáticamente crea una red, a la cual se le ha cambiado el nombre y la descripción.



Por seguridad, cambiar el tipo de acceso a la red, para que se deba autorizar cada vez que un nuevo nodo, quiera unirse a la red.



Una vez se ha creado la red, se debe instalar el paquete de *zerotier* en los servidores que van a ser parte de esta.

Primero se descarga y desencripta la clave GPG de *ZeroTier.*

```
admin@servidor:~$ curl -fsSL \
https://raw.githubusercontent.com/zerotier/ZeroTierOne/master/doc/contact%40zerotier.com.gpg | \
sudo gpg --dearmor -o /usr/share/keyrings/zerotier.gpg
```

Instalar *ZeroTier* con el script de instalación.

```
admin@servidor:~$ curl -s https://install.zerotier.com | sudo bash

*** Supported architectures vary by OS / distribution. We try to support
*** every system architecture supported by the target.
*** Please report problems to contact@zerotier.com and we will try to fix.
*** Detecting Linux Distribution

*** Found Ubuntu, creating /etc/apt/sources.list.d/zerotier.list
*** Installing zerotier-one package...
*** Enabling and starting ZeroTier service...
Synchronizing state of zerotier-one.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable zerotier-one

*** Waiting for identity generation...

*** Success! You are ZeroTier address [ 821f5006ed ].
admin@servidor:~$
```

Editar el archivo de lista de fuentes de *apt* para *ZeroTier*:

```
admin@servidor:~$ sudo nano /etc/apt/sources.list.d/zerotier.list
```

Modificar la lista de fuentes para apuntar al repositorio de *ZeroTier* y utiliza la clave GPG que se descargó en el paso anterior.

```
deb [arch=amd64 signed-by=/usr/share/keyrings/zerotier.gpg] http://download.zerotier.com/debian/jammy jammy main
```

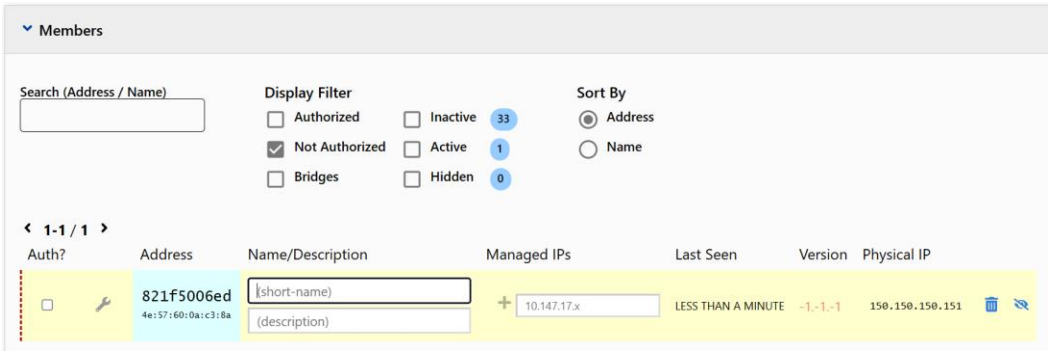
Actualizar la información a la más reciente en los repositorios, y en la salida verificar que aparezca *ZeroTier*.

```
admin@servidor:~$ sudo apt update
...
Hit:3 http://download.zerotier.com/debian/jammy jammy InRelease
...
```

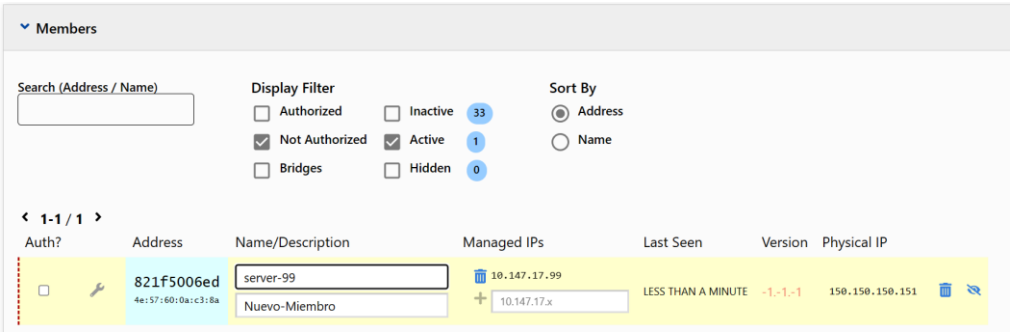
Como último paso de la configuración en el servidor es unirse a una red de *ZeroTier*, con el comando de *CLI* de *ZeroTier* y el ID de la red.

```
admin@servidor:~$ sudo zerotier-cli join 1234567898765432
200 join OK
```

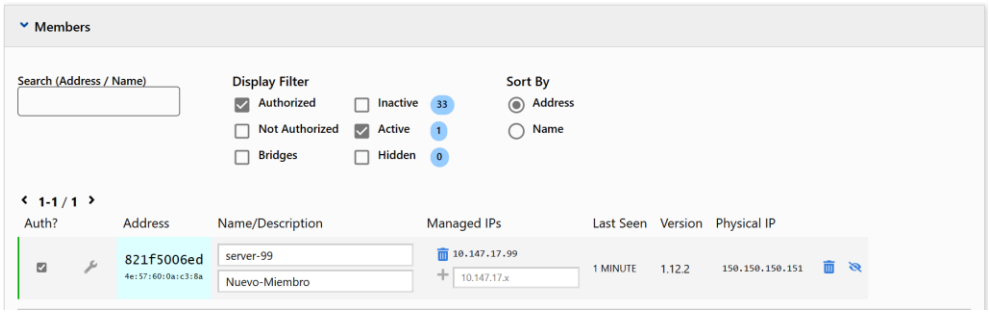
En el panel de la página web de *ZeroTier*, aparece una nueva solicitud, se puede filtrar por No Autorizado, para verlo.



Para una mejor organización, se le agrega un nombre o descripción, y una dirección IP de la red, antes de autorizarlo.



Luego de agregarle la información, marcamos el *check-box* de autorización y se le agrega la dirección IP asignada.



En el servidor se verifica la configuración de las interfaces y en la nueva interfaz de *ZeroTier* debe tener asignada la dirección IP. Ver interfaz *zt6ntbqeau.*

```
admin@servidor:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    ...
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    ...
3: zt6ntbqeau: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 2800 qdisc fq_codel state UNKNOWN group default qlen 1000
    link/ether 4e:57:60:0a:c3:8a brd ff:ff:ff:ff:ff:ff
    inet 10.147.17.99/24 brd 10.147.17.255 scope global zt6ntbqeau
       valid_lft forever preferred_lft forever
    inet6 fe80::4c57:60ff:fe0a:c38a/64 scope link
       valid_lft forever preferred_lft forever
```

Configuración de Componentes

DNS

La configuración del *DNS* cuenta de una sola máquina virtual el cual tiene alojado la resolución de nombre del dominio *venus-icc115.net*.

Cantidad	Nombre	Recursos
1	DNS	2GB RAM, 2CPU, 50GB Almacenamiento

Se tiene conectado el servidor a una red creada en *virt-manager* y a la VPN que sirve de conexión con las regiones.

Nombre de red	IPv4	Descripción
Infraestructura-Network	13.0.0.14	Interfaz conectada a Red de la Organización
VPN	10.147.17.33	Interfaz conectada a la VPN

Configuraciones previas

Interfaz

Para configurar la dirección IP de la máquina, se editó el archivo *netplan*.

```
admin@dns:~$ sudo nano /etc/netplan/00-installer-config.yaml
```

Se actualizó con el contenido de la siguiente imagen.

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp1s0:
      addresses:
        - 13.0.0.14/24
      routes:
        - to: default
          via: 13.0.0.1
      nameservers:
        addresses: [10.147.17.33,13.0.0.1]
```

Instalación de paquetes

El servicio que se utilizó para servir el *DNS* es *bind9*.

```
admin@dns:~$ sudo apt install bind9 -y
```

Configuración del servicio

Se creó la zona directa para dominio *venus-icc115.net.*

```
admin@dns:~$ sudo nano /etc/bind/named.conf.local
```

Se agregó las líneas que se visualizan en la siguiente imagen para que se registrase la zona con los archivos correspondientes.

```
zone "venus-icc115.net" {
        type master;
        file "/etc/bind/db.venus-icc115.net";
        forwarders {
                8.8.8.8;    # Google DNS
                1.1.1.1;    # Cloudflare DNS
        };
};

zone "17.147.10.in-addr.arpa" {
        type master;
        file "/etc/bind/db.17.147.10";
        forwarders {
                8.8.8.8;    # Google DNS
                1.1.1.1;    # Cloudflare DNS
        };
};
```

Creamos los archivos *db.venus-icc115.net* y *db.17.147.10*.

```
admin@dns:~$ sudo touch /etc/bind/db.venus-icc115.net
admin@dns:~$ sudo touch /etc/bind/db.17.147.10
```

Editamos el archivo de zona directa.

```
admin@dns:~$ sudo nano /etc/bind/db.venus-icc115.net
```

Agregamos el contenido de la imagen siguiente.

```
$TTL    60
@       IN      SOA     ns-1.venus-icc115.net. root.venus-icc115.net. (
                              7         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                          86400 )       ; Negative Cache TTL
;
@       IN      NS      ns-1.venus-icc115.net.

$ORIGIN venus-icc115.net.

ns-1    IN      A       10.147.17.33
main    IN      A       10.147.17.13
db      IN      A       10.147.17.36
service IN      A       10.147.17.39
ntp     IN      CNAME   service
rabbit  IN      CNAME   service
cache   IN      CNAME   service
etcd    IN      CNAME   service
identity        IN      A       10.147.17.19
regionone       IN      A       10.147.17.21
regiontwo       IN      A       10.147.17.22
regionthree     IN      A       10.147.17.23
ceph-admin      IN      A       10.147.17.41
ceph-mon        IN      A       10.147.17.44
ceph-osd-1      IN      A       10.147.17.47
ceph-osd-2      IN      A       10.147.17.50
ceph-osd-3      IN      A       10.147.17.53
```

Editamos el archivo de zona inversa.

```
admin@dns:~$ sudo nano /etc/bind/db.17.147.10
```

Agregamos el contenido de la siguiente imagen.

```
$TTL    60
@       IN      SOA     ns-1.venus-icc115.net. root.venus-icc115.net. (
                              7         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                          86400 )       ; Negative Cache TTL
;
@       IN      NS      ns-1.venus-icc115.net.
;$ORIGIN 17.147.10.in-addr.arpa.
33      IN      PTR     ns-1.venus-icc115.net.
13      IN      PTR     main.venus-icc115.net.
36      IN      PTR     db.venus-icc115.net
39      IN      PTR     service.venus-icc115.net.
19      IN      PTR     identity.venus-icc115.net.
21      IN      PTR     regionone.venus-icc115.net.
22      IN      PTR     regiontwo.venus-icc115.net.
23      IN      PTR     regionthree.venus-icc115.net.
41      IN      PTR     ceph-admin.venus-icc115.net.
44      IN      PTR     ceph-mon.venus-icc115.net.
47      IN      PTR     ceph-osd-1.venus-icc115.net.
50      IN      PTR     ceph-osd-2.venus-icc115.net.
53      IN      PTR     ceph-osd-3.venus-icc115.net.
```

Verificar la correcta configuración de las zonas con el comando de *CLI* de *bind9.*

```
admin@dns:~$ named-checkzone venus-icc115.net /etc/bind/db.venus-icc115.net
zone venus-icc115.net/IN: loaded serial 7
OK
admin@dns:~$ named-checkzone 17.147.10.in-addr.arpa /etc/bind/db.17.147.10
zone 17.147.10.in-addr.arpa/IN: loaded serial 7
OK
```

Reiniciamos el servicio en el servidor.

```
admin@dns:~$ sudo service bind9 restart
```

Clúster de base de datos de MariaDB

El clúster de base de datos se ha configurado con 4 máquinas virtuales.

Cantidad	Nombre	Recursos
1	MaxScale	1GB RAM, 2 CPU, 25 GB Almacenamiento
3	Maria1, Maria2, Maria3	1GB RAM, 2 CPU, 25 GB Almacenamiento

Se usó 3 redes creadas en *virt-manager* para el funcionamiento del clúster. Adicional, se ocupó una red *VPN* para que las regiones tengan acceso al servicio de base de datos.

Nombre de red	IPv4	Descripción
Infraestructura-Network	13.0.0.0/24	Red pública para la comunicación en los clientes y el clúster.
Private-Maxscale-Network	192.168.2.0/24	Red de comunicación entre el balanceador y los nodos.
Public-MariaDB-Network	192.168.3.0/24	Red de sincronización entre los servidores de mariadb
VPN	10.147.17.0/24	VPN

Asignación de IP a los nodos.

Nodo	Interfaz	IP
Maxscale	enp1s0	13.0.0.15
Maxscale	enp2s0	192.168.2.5
Maxscale	zt6ntbqeau	10.147.17.36
Maria1	enp1s0	192.168.2.11
Maria1	enp2s0	192.168.3.11
Maria2	enp1s0	192.168.2.12
Maria2	enp2s0	192.168.3.12
Maria3	enp1s0	192.168.2.13
Maria3	enp2s0	192.168.3.13

Configuraciones previas

Interfaces

Para configurar la dirección IP de la máquina, se editó el archivo *netplan*.

```
sudo nano /etc/netplan/00-installer-config.yaml
```

Se actualizó con el contenido de la siguiente imagen, según ip asignada en cada nodo de mariadb.

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp1s0:
      addresses:
        - 192.168.2.12/24
      routes:
        - to: default
          via: 192.168.2.1
      nameservers:
        addresses: [192.168.2.1]
    enp2s0:
      addresses:
        - 192.168.3.12/24
      routes:
        - to: 192.168.3.0/24
          via: 192.168.3.1
```

Para el nodo de *maxscale* se configuró con el contenido de la siguiente imagen.

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp1s0:
      addresses:
        - 13.0.0.15/24
      routes:
        - to: default
          via: 13.0.0.1
      nameservers:
        addresses: [10.147.17.33]
    enp2s0:
      addresses:
        - 192.168.2.5/24
      routes:
        - to: 192.168.2.0/24
          via: 192.168.2.1
```

Resolución de nombres

Para utilizar los nombres en lugar de las direcciones IP en las configuraciones que se realizaron, editamos el archivo hosts.

```
sudo nano /etc/hosts
```

Agregamos en cada uno de los nodos de mariadb el contenido que se visualiza en imagen siguiente**.**

```
127.0.0.1 localhost
127.0.1.1 ubuntu

192.168.3.11 maria-1
192.168.3.12 maria-2
192.168.3.13 maria-3

# The following lines are desirable for IPv6 capable hosts
::1             ip6-localhost ip6-loopback
fe00::0         ip6-localnet
ff00::0         ip6-mcastprefix
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
```

Agregamos la configuración del contenido de la siguiente imagen en el nodo de *maxscale.*

```
127.0.0.1 localhost
127.0.1.1 ubuntu

13.0.0.15 maxscale
192.168.2.2 maria-1
192.168.2.3 maria-2
192.168.2.4 maria-3

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localne
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Configuración del repositorio

Se agregó los repositorios de mariadb a la lista de repositorios de *Ubuntu* en las 4 máquinas virtuales ya que los paquetes de mariadb y *maxscale* están en los repositorios de mariadb.

```
admin@maria-2:~$ curl -Lss https://r.mariadb.com/downloads/mariadb_repo_setup | sudo bash
# [info] Checking for script prerequisites.
# [info] MariaDB Server version 11.1 is valid
# [info] Repository file successfully written to /etc/apt/sources.list.d/mariadb.list
# [info] Adding trusted package signing keys...
# [info] Running apt-get update...
# [info] Done adding trusted package signing keys
```

Una vez configurado el repositorio, se actualizó los paquetes en los nodos de *mariadb* y *maxscale*.

```
admin@maria-2:~$ sudo apt update
Hit:1 http://sv.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:4 https://dlm.mariadb.com/repo/mariadb-server/11.1/repo/ubuntu jammy InRelease [4714 B]
Hit:5 http://sv.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:6 http://sv.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:7 https://dlm.mariadb.com/repo/maxscale/latest/apt jammy InRelease [9344 B]
Hit:3 https://downloads.mariadb.com/Tools/ubuntu jammy InRelease
Fetched 14.1 kB in 1s (12.4 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
```

Con los paquetes actualizados, se instaló *mariadb-server* en los 3 nodos de *mariadb*.

```
admin@maria-2:~$ sudo apt install mariadb-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  galera-4 libconfig-fast-perl libcgi-pm-perl libclone-perl
  libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl
  libencode-locale-perl libfcgi-bin libfcgi-perl libfcgi0ldbl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl
  liblwp-mediatypes-perl libmariadb3 libmysqlclient21 libncdt16 libpem1
  libtimedate-perl liburi-perl liburing2 mariadb-client mariadb-client-compat
  mariadb-client-core mariadb-common mariadb-server mariadb-server-compat
  mariadb-server-core mysql-common pv socat
Suggested packages:
  libmldbm-perl libnet-daemon-perl libsql-statement-perl libdata-dump-perl
  libipc-sharedcache-perl libbusiness-isbn-perl libwww-perl mailx
  mariadb-test doc-base
The following NEW packages will be installed:
  galera-4 libconfig-fast-perl libcgi-pm-perl libclone-perl
  libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl
  libencode-locale-perl libfcgi-bin libfcgi-perl libfcgi0ldbl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl
  liblwp-mediatypes-perl libmariadb3 libmysqlclient21 libncdt16 libpem1
  libtimedate-perl liburi-perl liburing2 mariadb-client mariadb-client-compat
  mariadb-client-core mariadb-common mariadb-server mariadb-server-compat
  mariadb-server-core mysql-common pv socat
0 upgraded, 36 newly installed, 0 to remove and 1 not upgraded.
Need to get 30.1 MB of archives.
After this operation, 233 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Finalmente, se instaló el servicio de *maxscale* en el nodo de maxscale.

```
admin@maxscale:~$ sudo apt install maxscale
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
   libltdl7 libodbc2
Suggested packages:
   odbc-postgresql tdsodbc
The following NEW packages will be installed:
   libltdl7 libodbc2 maxscale
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 120 MB of archives.
After this operation, 607 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Configuración del clúster de Mariadb

Esta configuración se realizó en los tres nodos de Mariadb para configurarlos como un clúster. Creamos el archivo de configuración *mariadb.cnf*.

```
admin@maria-2:~$ sudo nano /etc/mysql/conf.d/mariadb.cnf
```

Se agregó el contenido de la siguiente imagen. En cada nodo se cambió las últimas dos líneas con la información correspondiente a cada uno.

```
[mysqld]
query_cache_size=0
binlog_format=ROW
default-storage-engine=innodb
innodb_autoinc_lock_mode=2
query_cache_type=0
bind-address=0.0.0.0

# Galera Provider Configuration.
# 'wsrep_on' activa el cluster
wsrep_on=ON
wsrep_provider=/usr/lib/galera/libgalera_smm.so

# Galera Cluster Configuration.
wsrep_cluster_name="ClusterMaria"
# 'wsrep_cluster_address'
# Indicamos las IPs de los nodos que van a formar parte del cluster
wsrep_cluster_address="gcomm://192.168.3.11,192.168.3.12,192.168.3.13"

# Galera Synchronization Configuration.
wsrep_sst_method=rsync

# Galera Node Configuration.
# 'wsrep_node_address' dirección IP del nodo que estamos configurando
wsrep_node_address="192.168.3.12"
wsrep_node_name="maria-2"
```

Posteriormente, se cambió de igual forma en todos los nodos de mariadb la configuración para que el servicio de mariadb escuche por todas las interfaces.

```
admin@maria-2:~$ sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

Configuramos el bind-address en 0.0.0.0.

```
[server]

[mariadb]
user            = mysql
pid-file        = /run/mysqld/mysqld.pid
basedir         = /usr
datadir         = /var/lib/mysql
tmpdir          = /tmp
bind-address    = 0.0.0.0

#
# * Fine Tuning
#
key_buffer_size     = 128M
max_allowed_packet  = 1G
thread_stack        = 192K
thread_cache_size   = 8
# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
myisam_recover_options = BACKUP
```

Inicialización del clúster

Una vez creado el archivo de configuración del clúster en todos los nodos de *maríadb*, lo activamos con el comando desde el nodo 2 con el siguiente comando.

```
admin@maria-2:~$ sudo galera_new_cluster
```

Luego se procedió a revisar el estado del clúster. Al estar todo correcto, y los 3 nodos sincronizados, tenemos la siguiente salida.

```
admin@maria-2:~$ sudo mariadb -u root -p -e "SHOW STATUS LIKE 'wsrep_cluster_size'"
Enter password:
+--------------------+-------+
| Variable_name      | Value |
+--------------------+-------+
| wsrep_cluster_size |   3   |
+--------------------+-------+
```

Configuración del balanceador

Instalación de paquetes

Se instaló *maxscale.*

```
admin@maxscale:~$ sudo apt update
admin@maxscale:~$ sudo apt install maxscale -y
```

Configuración de usuario

Se creó y configuró un usuario para el monitoreo y peticiones al clúster. Es necesario destacar que al estar en funcionamiento el clúster, solo hizo falta realizar esto en un nodo del clúster y se replicó a los demás.

Se ingresó a la *CLI* de *mariadb/mysql.*

```
admin@maria-2:~$ sudo mariadb
```

Se ejecutaron las consultas correspondientes a la creación de usuario y asignación de permisos sobre las diferentes acciones.

```
CREATE USER 'maxscale'@'%' IDENTIFIED BY 'ultrasecreta';
GRANT SELECT ON mysql.user TO 'maxscale'@'%';
GRANT SELECT ON mysql.db TO 'maxscale'@'%';
GRANT SELECT ON mysql.tables_priv TO 'maxscale'@'%';
GRANT SELECT ON mysql.columns_priv TO 'maxscale'@'%';
GRANT SELECT ON mysql.proxies_priv TO 'maxscale'@'%';
GRANT SELECT ON mysql.procs_priv TO 'maxscale'@'%';
GRANT SELECT ON mysql.roles_mapping TO 'maxscale'@'%';
GRANT SHOW DATABASES ON *.* TO 'maxscale'@'%';
GRANT REPLICATION CLIENT on *.* to 'maxscale'@'%';
GRANT ALL ON infinidb_vtable.* TO 'maxscale'@'%';
EXIT;
```

Posteriormente, se configuró el archivo *maxscale.cnf.*

```
[maxscale]
threads=auto
log_augmentation = 1
ms_timestamp = 1
syslog = 1
admin_host=0.0.0.0
admin_secure_gui=false
[maria-1]
type=server
address=192.168.2.11
port=3306
protocol=MariaDBBackend
[maria-2]
type=server
address=192.168.2.12
port=3306
protocol=MariaDBBackend
[maria-3]
type=server
address=192.168.2.13
port=3306
protocol=MariaDBBackend
[Galera-Monitor]
type=monitor
module=galera_mon
servers=maria-1,maria-2,maria-3
user=maxscale
password=ultrasecreta
monitor_interval=2000ms
[Read-Con-Route-Galera-Service]
type=service
router=readconnroute
servers=maria-1,maria-2,maria-3
user=maxscale
password=ultrasecreta
[Read-Con-Route-Galera-Listener]
type=listener
service=Read-Con-Route-Galera-Service
protocol=MariaDBClient
port=3306
address=13.0.0.15
[Read-Con-Route-Galera-VPN-Listener]
type=listener
service=Read-Con-Route-Galera-Service
protocol=MariaDBClient
port=3306
address=10.147.17.36
```

Se inicializó el servicio.

```
admin@maxscale:~$ sudo service maxscale start
```

Finalmente, se monitoreó los nodos asociados con el siguiente comando.

```
admin@maxscale:~$ sudo maxctrl list servers
```

Server	Address	Port	Connections	State	GTID	Monitor
maria-1	192.168.2.11	3306	0	Master, Synced, Running		Galera-Monitor
maria-2	192.168.2.12	3306	0	Slave, Synced, Running		Galera-Monitor
maria-3	192.168.2.13	3306	0	Slave, Synced, Running		Galera-Monitor

Clúster de CEPH

Para el clúster de CEPH se configuró 5 máquinas virtuales.

Cantidad	Nombre	Recursos
1	ceph-admin	4GB RAM, 2CPU, 100 GB Almacenamiento
1	ceph-mon	4GB RAM, 2CPU, 100 GB Almacenamiento
3	ceph-osd1, ceph-osd2, ceph-osd3	2GB RAM, 2 CPU, 2x100 GB Almacenamiento

Asignación de IP a los nodos.

Nodo	Interfaz	IP
ceph-admin	enp1s0	13.0.0.41
ceph-admin	enp7s0	192.168.4.41
ceph-admin	zt6ntbqeau	10.147.17.43
ceph-mon	enp1s0	13.0.0.42
ceph-mon	enp7s0	192.168.4.42
ceph-mon	zt6ntbqeau	10.147.17.46
ceph-osd1	enp1s0	13.0.0.43
ceph-osd1	enp7s0	192.168.4.43
ceph-osd1	zt6ntbqeau	10.147.17.49
ceph-osd2	enp1s0	13.0.0.44
ceph-osd2	enp7s0	192.168.4.44
ceph-osd2	zt6ntbqeau	10.147.17.52
ceph-osd3	enp1s0	13.0.0.45
ceph-osd3	enp7s0	192.168.4.45
ceph-osd3	zt6ntbqeau	10.147.10.55

Configuraciones previas

Interfaces

Para configurar la dirección IP de la máquina, se editó el archivo *netplan*.

```
admin@ceph-admin:~$ sudo nano /etc/netplan/00-installer-config.yaml
```

Se agregó el contenido de la imagen siguiente en cada nodo de acuerdo con su ip asignada.

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp1s0:
      addresses:
        - 13.0.0.41/24
      routes:
        - to: default
          via: 13.0.0.1
      nameservers:
        addresses: [13.0.0.1]
    enp7s0:
      addresses:
        - 192.168.4.41/24
      routes:
        - to: 192.168.4.0/24
          via: 192.168.4.1
```

Resolución de nombres

Para utilizar los nombres en lugar de las direcciones IP en las configuraciones que se realizaron, editamos el archivo *hosts*.

```
admin@ceph-admin:~$ sudo nano /etc/hosts
```

Agregamos el contenido de la imagen siguiente según corresponde en cada nodo del clúster de *ceph*.

```
127.0.0.1 localhost
127.0.1.1 ceph-admin

192.168.4.41 ceph-admin
192.168.4.42 ceph-mon
192.168.4.43 ceph-osd-1
192.168.4.44 ceph-osd-2
192.168.4.45 ceph-osd-3

# The following lines are desirable for IPv6 capable hosts
::1       ip6-localhost ip6-loopback
fe00::0   ip6-localnet
ff00::0   ip6-mcastprefix
ff02::1   ip6-allnodes
ff02::2   ip6-allrouters
```

Instalación de paquetes

Se realizó las siguientes instalaciones en todos los nodos del clúster.

Instalación de *Chrony*.

```
admin@ceph-admin:~$ sudo apt install chrony -y
```

Instalación de paquetes necesarios para *Docker.*

```
admin@ceph-admin:~$ sudo apt install apt-transport-https ca-certificates curl gnupg-agent software-properties-common -y
```

Se agregó la clave *GPG* oficial de *Docker.*

```
admin@ceph-admin:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

```
admin@ceph-admin:~$ echo "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -sc) stable" | sudo tee /etc/apt/sources.list.d/docker-ce.list
```

Se actualizó el índice de paquetes de los repositorios.

```
admin@ceph-admin:~$ sudo apt update
```

Se instaló los paquetes de *Docker.*

```
admin@ceph-admin:~$ sudo apt install docker-ce docker-ce-cli containerd.io -y
```

Se hizo corrección de la forma de almacenar las claves *GPG* en *Ubuntu* 22.04.

```
admin@ceph-admin:~$  sudo mv /etc/apt/trusted.gpg /etc/apt/trusted.gpg.d/
```

Configuración del servicio

Creación de usuario

Se creó un usuario en el **nodo ceph-admin** para poder ser usado vía conexión ssh desde un cliente para la recuperación de archivos de configuración.

Para esto, fue necesario cambiarnos a la cuenta root. El usuario creado es *cephadmin* y la contraseña se estableció como icc115.

```
admin@ceph-admin:~$ sudo su
root@ceph-admin:/home/admin# useradd -m -s /bin/bash cephadmin
root@ceph-admin:/home/admin# passwd cephadmin
New password:
Retype new password:
passwd: password updated successfully
```

Agregamos el usuario al grupo sudo para tener permiso de super usuario.

```
root@ceph-admin:/home/admin# echo "cephadmin ALL=(ALL:ALL) NOPASSWD:ALL" >> /etc/sudoers.d/cephadmin
root@ceph-admin:/home/admin# chmod 0440 /etc/sudoers.d/cephadmin
```

Preparación de los nodos OSD y monitor

Para que el nodo admin pudiera conectarse a los otros nodos fue necesario permitir conexión por ssh en nodos osd y al monitor. Este proceso requiere realizarse desde la cuenta root y se edita el archivo *sshd_config*.

```
admin@ceph-mon:~$ sudo su
root@ceph-mon:/home/admin# nano /etc/ssh/sshd_config
```

Se busca la línea ***PermitRootLogin*** y se establece en ***yes.***

```
...
PermitRootLogin yes
...
```

Luego de realizado el cambio, se reinició el servicio.

```
root@ceph-mon:/home/admin# /etc/init.d/ssh restart
```

Con la conexión por ssh habilitada al usuario root, se estableció la contraseña ***icc115*** en todos los nodos osd y el nodo monitor.

```
root@ceph-mon:/home/admin# passwd
New password:
Retype new password:
passwd: password updated successfully
```

Descarga de CEPH

En el nodo ceph-admin se instaló ceph manualmente. Para esto, se descargó la utilidad *cephadm* y se instaló desde la cuenta de *root*.

```
root@ceph-admin:/home/admin# wget -q https://github.com/ceph/ceph/raw/pacific/src/cephadm/cephadm -P /usr/bin/
```

Se cambió los permisos a la utilidad.

```
root@ceph-admin:/home/admin# chmod +x /usr/bin/cephadm
```

Despliegue del servicio

Se inició un nuevo clúster con el comando *bootstrap* de *cephadm*.

La dirección IP del parámetro *–mon-ip* corresponde con la IP pública que tiene asignada el nodo ceph-admin. Ceph toma por defecto la red de esta dirección IP como la red pública del clúster. Se debe agregar además el *parámetro --cluster-network* para que Ceph sepa que tenemos una red privada dedicada a la comunicación y sincronización de los nodos.

```
cephadmin@ceph-admin:$ sudo cephadm bootstrap --mon-ip 13.0.0.41 --cluster-network 192.168.4.0/24
```

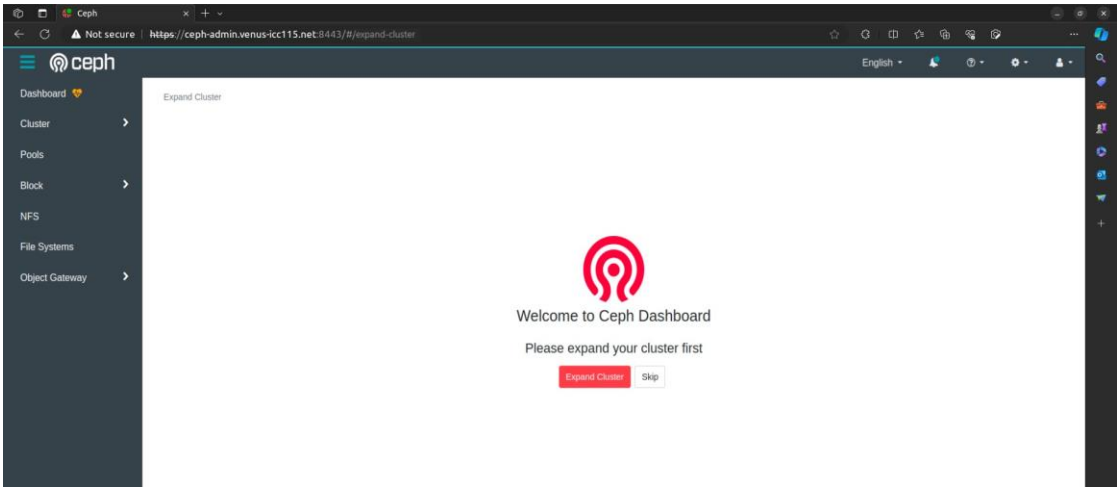
La ejecución de este comando tarda un poco ya que realiza varias comprobaciones y acciones sobre los nodos osd y monitor.

Cuando haya terminado la ejecución, al no haber ningún problema, se obtuvo la siguiente salida.

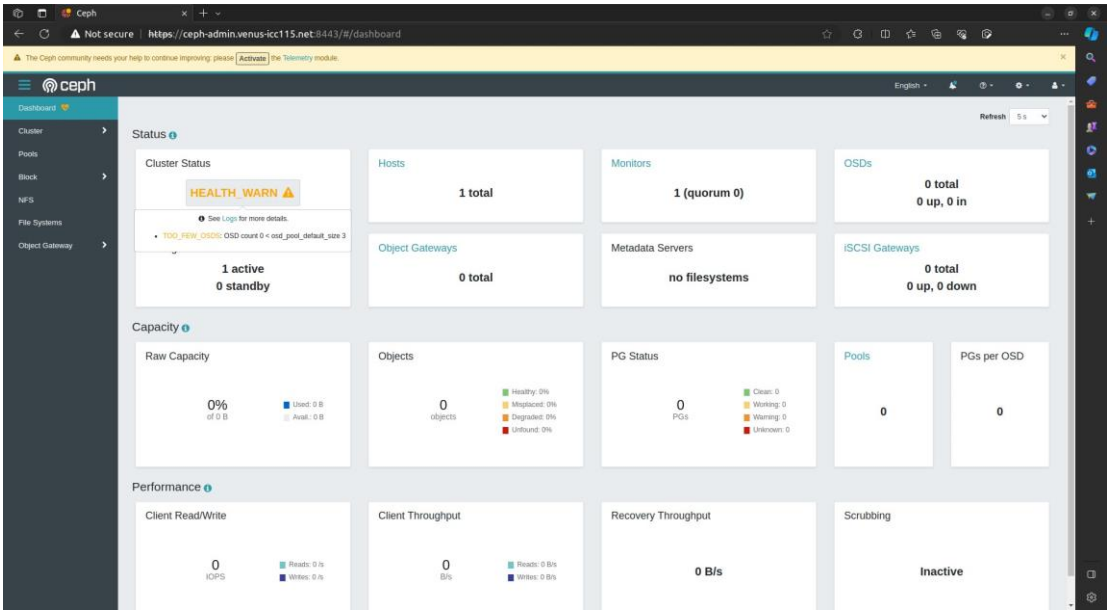
```
Ceph Dashboard is now available at:

URL: https://ceph-admin:8443/
User: admin
Password: zhdhir9235

Enabling client.admin keyring and conf on hosts with "admin" label
Enabling autotune for osd_memory_target
You can access the Ceph CLI as following in case of multi-cluster or non-default config:

sudo /usr/bin/cephadm shell --fsid 96df340c-704d-11ee-acc9-051960837027 -c /etc/ceph/ceph.conf -k /etc/ceph/ceph.client.admin.keyring

Or, if you are only running a single cluster on this host:

sudo /usr/bin/cephadm shell

Please consider enabling telemetry to help improve Ceph:

ceph telemetry on

For more information see:

https://docs.ceph.com/en/pacific/mgr/telemetry/

Bootstrap complete.
```

La consola indica que se puede acceder al servicio de administración web desde un nodo de la red desde un navegador en *https://ceph-admin:8443* ya que es el hostname que he hemos configurado al inicio. También nos podemos conectar si agregamos ese nombre al archivo hosts en nuestra maquina física o también desde *https://13.0.0.41:8443*. Pero ya que se ha configurado una VPN y un DNS accedemos directamente a *https://ceph-admin.venus-icc115.net:8443.*

Con los datos de *user* y *password* obtenidos en la salida del despliegue, ingresamos al *dashboard*.



Se siguen los pasos por defecto para expandir el cluster y se debe mostrar el dashboard de la siguiente forma.



Aparece una advertencia porque aún no se han agregados nodos OSD.

Al tener desplegado el servicio, se activa *Ceph CLI* con el comando que muestra la salida del comando de despliegue.

```
cephadmin@ceph-admin:$ sudo /usr/bin/cephadm shell --fsid 96df340c-704d-11ee-acc9-051960837027 -c /etc/ceph/ceph.conf -k /etc/ceph/ceph.client.admin.keyring
```

Lo anterior, pone en ejecución un contenedor en forma interactiva. Para confirmar que está correcto Ceph CLI se debe ejecutar el ***ceph -s*** con el cual se podrá ver el detalle de la salud del clúster.

```
root@ceph-admin:/# ceph -s
cluster:
  id: 96df340c-704d-11ee-acc9-051960837027
  health: HEALTH_WARN
  OSD count 0 < osd_pool_default_size 3

services:
  mon: 1 daemons, quorum ceph-admin (age 13m)
  mgr: ceph-admin.htdejn(active, since 13m)
  osd: 0 osds: 0 up, 0 in

data:
  pools: 0 pools, 0 pgs
  objects: 0 objects, 0 B
  usage: 0 B used, 0 B / 0 B avail
  pgs:
```

Adicionalmente, es recomendable disponer de librerías comunes de acceso desde la terminal, para ello, se ejecutó los siguientes comandos.

```
root@ceph-admin:/# sudo cephadm add-repo --release pacific
root@ceph-admin:/# sudo cephadm install ceph-common
```

Con el comando *exit* salimos del contenedor e instalamos también en el servidor *ceph-common*.

```
admin@ceph-admin:~$ sudo cephadm install ceph-common
```

Para listar los componentes del clúster ejecutar.

```
admin@ceph-admin:~$ sudo ceph orch host ls
```

Se muestra una salida como la siguiente.

```
HOST       ADDR      LABELS   STATUS
ceph-admin 13.0.0.41 _admin
1 hosts in cluster
```

Registro del nodo monitor

El registro del nodo monitor se realiza desde el nodo ceph-admin.

Para este proceso, primeramente, se copia la clave pública del usuario al archivo *authorized_keys* del usuario root hacia el servidor ceph-mon.

```
cephadmin@ceph-admin:$ sudo ssh-copy-id -f -i /etc/ceph/ceph.pub root@ceph-mon
```

Luego se agrega el nodo monitor a la lista de los hosts administrados por el orquestador ceph-admin.

```
cephadmin@ceph-admin:$ sudo ceph orch host add ceph-mon
```

Finalmente, se agrega la etiqueta mon/osd al nodo ceph-mon recién agregado.

```
cephadmin@ceph-admin:$ sudo ceph orch host label add ceph-mon mon/osd
```

Registro de los nodos OSD

El registro de los nodos OSD también debe realizarse desde el nodo ceph-admin. Copiamos primeramente la clave pública desde ceph-admin a los nodos ceph-osd. Escribimos yes en la consola interactiva y colocamos la contraseña *icc115*.

```
cephadmin@ceph-admin:~$ for i in ceph-osd-1 ceph-osd-2 ceph-osd-3; do sudo ssh-copy-id -f -i /etc/ceph/ceph.pub root@$i; done
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/etc/ceph/ceph.pub"
The authenticity of host 'ceph-osd-1 (192.168.4.43)' can't be established.
ED25519 key fingerprint is SHA256:u0jg0mzky+zLkzKCCfYRErtstvD5djnwd6mzTYaaxc0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
root@ceph-osd-1's password:
Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@ceph-osd-1'"
and check to make sure that only the key(s) you wanted were added.

/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/etc/ceph/ceph.pub"
The authenticity of host 'ceph-osd-2 (192.168.4.44)' can't be established.
ED25519 key fingerprint is SHA256:u0jg0mzky+zLkzKCCfYRErtstvD5djnwd6mzTYaaxc0.
This host is known by the following other names/addresses:
~/.ssh/known_hosts:4: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
root@ceph-osd-2's password:
Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@ceph-osd-2'"
and check to make sure that only the key(s) you wanted were added.

/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/etc/ceph/ceph.pub"
The authenticity of host 'ceph-osd-3 (192.168.4.45)' can't be established.
ED25519 key fingerprint is SHA256:u0jg0mzky+zLkzKCCfYRErtstvD5djnwd6mzTYaaxc0.
This host is known by the following other names/addresses:
~/.ssh/known_hosts:7: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
root@ceph-osd-3's password:
Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@ceph-osd-3'"
and check to make sure that only the key(s) you wanted were added.

cephadmin@ceph-admin:~$
```

Luego se registran en el clúster.

```
cephadmin@ceph-admin:~$ sudo ceph orch host add ceph-osd-1
Added host 'ceph-osd-1' with addr '192.168.4.43'
cephadmin@ceph-admin:~$ sudo ceph orch host add ceph-osd-2
Added host 'ceph-osd-2' with addr '192.168.4.44'
cephadmin@ceph-admin:~$ sudo ceph orch host add ceph-osd-3
Added host 'ceph-osd-3' with addr '192.168.4.45'
cephadmin@ceph-admin:~$
```

Finalmente, agregamos las etiquetas a los nodos osd que se agregaron al clúster.

```
cephadmin@ceph-admin:~$ for i in ceph-osd-1 ceph-osd-2 ceph-osd-3; do sudo ceph orch host label add $i osd; done
Added label osd to host ceph-osd-1
Added label osd to host ceph-osd-2
Added label osd to host ceph-osd-3
cephadmin@ceph-admin:~$
```

Listamos nuevamente para ver los cambios en el clúster.

```
cephadmin@ceph-admin:~$ sudo ceph orch host ls
HOST        ADDR         LABELS     STATUS
ceph-admin  13.0.0.41    _admin
ceph-mon    192.168.4.42  mon/osd
ceph-osd-1  192.168.4.43  osd
ceph-osd-2  192.168.4.44  osd
ceph-osd-3  192.168.4.45  osd
5 hosts in cluster
cephadmin@ceph-admin:~$
```

Inflando el clúster con los OSD

Preparación de discos

Se tuvo que configurar la unidad de almacenamiento adicional que tienen los OSD. Para esto, se realiza los siguientes pasos en todos los nodos OSD.

Listamos los discos disponibles en el host.

```
admin@ceph-osd-1:~$ sudo fdisk -l
```

Creamos el volumen LVM *vg01* en el dispositivo */dev/vdb* que es el storage adicional de 100GB.

```
admin@ceph-osd-1:~$ sudo vgcreate vg01 /dev/vdb
Physical volume "/dev/vdb" successfully created.
Volume group "vg01" successfully created
admin@ceph-osd-1:~$
```

Creamos el nuevo volumen lógico de 100GB llamado *lv01* en el grupo de volúmenes *vg01*.

```
admin@ceph-osd-1:~$ sudo lvcreate -L 100G -n lv01 vg01
```

Agregación de los volúmenes al clúster

Agregamos los demonios OSD al clúster de Ceph. Los demonios se crean automáticamente en los nodos ceph-osd1, ceph-osd2, ceph-osd3 y se utiliza el volumen lógico *lv01* que se encuentra en el grupo de volúmenes *vg01*.

Los siguientes comandos se ejecutan en el nodo ceph-admin.

```
cephadmin@ceph-admin:~$ sudo ceph orch daemon add osd ceph-osd-1:vg01/lv01
Created osd(s) 0 on host 'ceph-osd-1'
cephadmin@ceph-admin:~$ sudo ceph orch daemon add osd ceph-osd-2:vg01/lv01
Created osd(s) 1 on host 'ceph-osd-2'
cephadmin@ceph-admin:~$ sudo ceph orch daemon add osd ceph-osd-3:vg01/lv01
Created osd(s) 2 on host 'ceph-osd-3'
cephadmin@ceph-admin:~$
```

Servicios Fundamentales

Se ha configurado un servidor para la provisión de los servicios de rabbitmq, Chrony y memcached.

Cantidad	Nombre	Recursos
1	services	4GB RAM, 2CPU, 50GB

Se usó una sola red creada en virt-manager para el funcionamiento del servidor. De igual forma, se utilizó una VPN para que las regiones tengan acceso.

Nombre de red	IPv4	Descripción
Infraestructura-Network	13.0.0.12	Interfaz de comunicación con la red de la Organización
VPN	10.147.17.39	VPN

Configuraciones previas

Interfaz

Para configurar la dirección IP de la máquina, se editó el archivo netplan.

```
admin@services:~$ sudo nano /etc/netplan/00-installer-config.yaml
```

Se actualizó con el contenido de la imagen siguiente.

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp1s0:
      addresses:
        - 13.0.0.12/24
      routes:
        - to: default
          via: 13.0.0.1
      nameservers:
        addresses: [10.147.17.33]
```

Instalación de paquetes

Se realizó la instalación de los paquetes de rabbitmq, chrony y memcached.

```
admin@services:~$ sudo apt install rabbitmq-server chrony memcached
```

Configuración de Rabbit

Se creó y configuró un usuario para el usuario administrador con los siguientes comandos.

```
admin@services:~$ sudo rabbitmqctl add_user admin icc115
Adding user "admin" ...
Done. Don't forget to grant the user permissions to some virtual hosts!
See 'rabbitmqctl help set_permissions' to learn more.
admin@services:~$ sudo rabbitmqctl add_vhost /main
Adding vhost "/main" ...
admin@services:~$ sudo rabbitmqctl set_permissions -p /main main ".*" ".*" ".*"
Setting permissions for user "main" in vhost "/main" ...
admin@services:~$
```

Se habilitó el panel WEB.

```
admin@services:~$ sudo rabbitmq-plugins enable rabbitmq_management
Enabling plugins on node rabbit@identity:
rabbitmq_management
The following plugins have been configured:
  rabbitmq_management
  rabbitmq_management_agent
  rabbitmq_web_dispatch
Applying plugin configuration to rabbit@identity...
The following plugins have been enabled:
  rabbitmq_management
  rabbitmq_management_agent
  rabbitmq_web_dispatch
started 3 plugins.
```

Se concedieron permisos de administrador al usuario *main.*

```
admin@services:~$ sudo rabbitmqctl set_user_tags main administrator
Setting tags for user "main" to [administrator] ...
```

Finalmente se crearon las colas para las regiones.

```
admin@services:~$ sudo rabbitmqctl add_user regionone icc115
admin@services:~$ sudo rabbitmqctl add_vhost /regionone
admin@services:~$ sudo rabbitmqctl set_permissions -p /regionone regionone ".*" ".*" ".*"
admin@services:~$ sudo rabbitmqctl add_user regiontwo icc115
admin@services:~$ sudo rabbitmqctl add_vhost /regiontwo
admin@services:~$ sudo rabbitmqctl set_permissions -p /regiontwo regiontwo ".*" ".*" ".*"
admin@services:~$ sudo rabbitmqctl add_user regionthree icc115
admin@services:~$ sudo rabbitmqctl add_vhost /regionthree
admin@services:~$ sudo rabbitmqctl set_permissions -p /regionthree regionthree ".*" ".*" ".*"
```

Configuración de Chrony

Editamos el archivo de configuración de chrony.

```
admin@services:~$ sudo nano /etc/chrony/chrony.conf
```

```
# Agregar el servidor ntp
server ntp.ues.edu.sv iburst

# Agregar al final del archivo las redes que se va  permitir consumir el servicio ntp
# Allow organization network
allow 13.0.0.0/24

# Allow VPN
allow 10.147.17.0/24
```

Reiniciamos el servicio.

```
admin@services:~$ sudo service chrony restart
```

Verificamos las fuentes.

```
admin@services:~$ chronyc sources
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^? 168.232.49.175                0   6     0     -     +0ns[   +0ns] +/-    0ns
^- prod-ntp-5.ntp4.ps5.cano>     2   6    17     1  -1107us[  +18us] +/-   74ms
^+ alphyn.canonical.com          2   6    17     3   -637us[ +488us] +/-   63ms
^* prod-ntp-3.ntp4.ps5.cano>     2   6    17     2   +946us[+2071us] +/-   73ms
^- prod-ntp-4.ntp4.ps5.cano>     2   6    17     3    +32ms[  +33ms] +/-  104ms
^? 5-243-119-74.ritternet.c>     0   6     0     -     +0ns[   +0ns] +/-    0ns
^+ ntp4.glypnod.com              2   6    17     4    -14ms[  -13ms] +/-   58ms
^- li1210-167.members.linod>     2   6    17     5    +18ms[  +20ms] +/-  109ms
^+ ec2-18-119-130-247.us-ea>     2   6    17     6  +2377us[+3502us] +/-  116ms
```

Configuración de Memcached

Editamos el archivo de configuración de memcached.

```
admin@services:~$ sudo nano /etc/memcached.conf
```

Configuramos las interfaces por las que va a escuchar peticiones.

```
-d
logfile /var/log/memcached.log
-m 64
-p 11211
-u memcache
-l 10.147.17.39 13.0.0.12
-P /var/run/memcached/memcached.pid
```

Reiniciamos el servicio.

```
admin@services:~$ sudo service memcached restart
```

Verificamos que el servicio esté corriendo correctamente.

```
admin@services:~$ sudo service memcached status
● memcached.service - memcached daemon
     Loaded: loaded (/lib/systemd/system/memcached.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2023-10-10 05:35:29 UTC; 2s ago
       Docs: man:memcached(1)
   Main PID: 3159 (memcached)
      Tasks: 10 (limit: 2220)
     Memory: 1.7M
        CPU: 176ms
     CGroup: /system.slice/memcached.service
             └─3159 /usr/bin/memcached -m 64 -p 11211 -u memcache -l 10.147.17.39 -P /var/run/memcached/memcached.pid
```

Servicio de LDAP

Se ha configurado un servidor para albergar el servicio de federación de usuario por el software de *openldap*.

Cantidad	Nombre	Recursos
1	identity	4GB RAM, 2CPU, 50GB

Se tiene conectado el servidor a una red creada en virt-manager y a la VPN que sirve de conexión con las regiones.

Nombre de red	IPv4	Descripción
Infraestructura-Network	13.0.0.13	Interfaz conectada a Red de la Organización
VPN	10.147.17.39	Interfaz conectada a la VPN

Configuraciones Previas

Interfaz

Para configurar la dirección IP de la máquina, se editó el archivo netplan.

```
admin@identity:~$ sudo nano /etc/netplan/00-installer-config.yaml
```

Se actualizó con el contenido de la imagen siguiente.

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp1s0:
      addresses:
        - 13.0.0.13/24
      routes:
        - to: default
          via: 13.0.0.1
      nameservers:
        addresses: [10.147.17.33]
```

Instalación de paquetes

Actualizar los paquetes de Ubuntu y luego instalar el paquete del servicio *openldap* y utilidades necesarias.

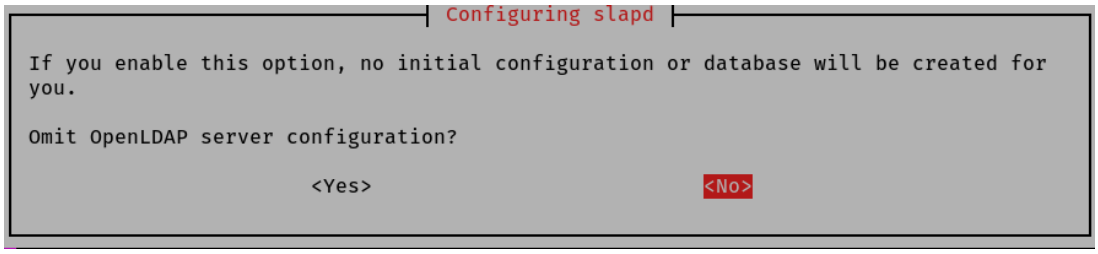
```
admin@identity:~$ sudo apt update
admin@identity:~$ sudo apt install slapd ldap-utils
```

Configurar el servicio con los datos correctos.

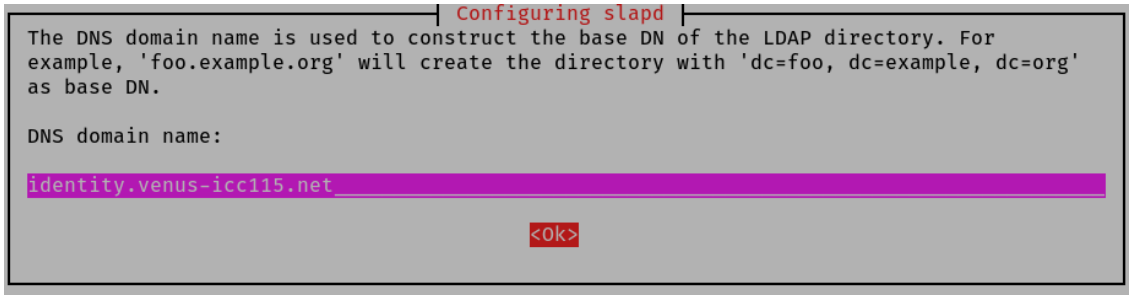
```
admin@identity:~$ sudo dpkg-reconfigure slapd
```

A continuación, se mostrarán en la consola algunas preguntas, las cuales se responden de la siguiente forma:

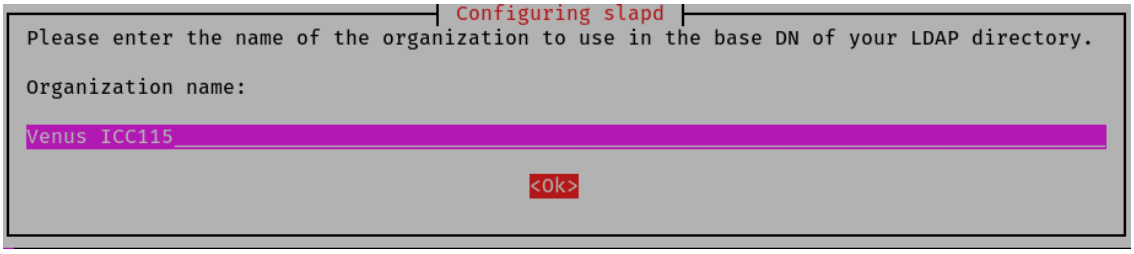
La siguiente imagen muestra la pregunta: ¿Omitir la configuración del servidor LDAP?, para la cual seleccionaremos la opción NO.



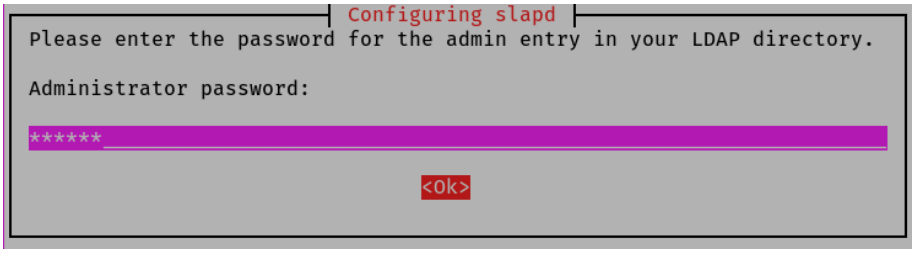
Se nos preguntará a continuación el nombre del dominio DNS, introducimos el siguiente: identity.venus-icc115.net.



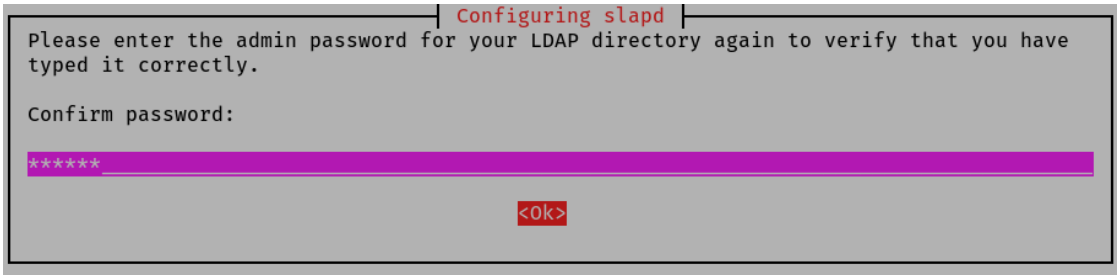
A continuación, se solicita ingresar el nombre de la organización, colocamos el siguiente: Venus ICC115.



Se solicita la contraseña del administrador, en este caso introducimos como contraseña: icc115

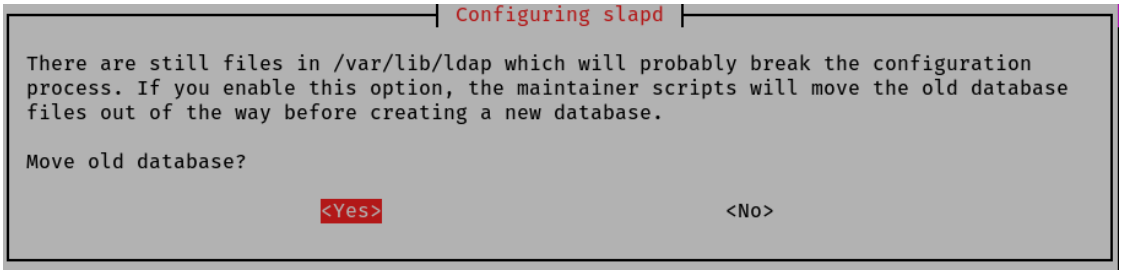


Se solicita confirmar la contraseña escrita anteriormente



Se nos pregunta si deseamos eliminar la base de datos cuando se purge slapd y respondemos que NO.

Por último, se nos pregunta si deseamos mover la última base de datos, seleccionamos SI y de esta manera finalizamos con la instalación.



Configuración del servicio

Para crear el árbol de directorios necesario para el caso de estudio se debe crear un archivo con extensión. *ldif*, en este caso lo llamamos *setup.ldif* con el siguiente contenido.

```
dn: cn=accounts,dc=identity,dc=net
objectClass: organizationalRole
cn: accounts

dn: cn=users,cn=accounts,dc=identity,dc=net
objectClass: organizationalRole
cn: users

dn: cn=groups,cn=accounts,dc=identity,dc=net
objectClass: organizationalRole
cn: groups
```

Para importar estas configuraciones al servidor se ejecuta el comando.

```
admin@identity:~$ ldapadd -x -D "cn=admin,dc=identity,dc=venus-icc115,dc=net" -W -f setup.ldif
Enter LDAP Password:
adding new entry "cn=accounts,dc=identity,dc=venus-icc115,dc=net"
adding new entry "cn=users,cn=accounts,dc=identity,dc=venus-icc115,dc=net"
adding new entry "cn=groups,cn=accounts,dc=identity,dc=venus-icc115,dc=net"
```

El comando anterior pide la contraseña maestra del servicio *ldap*.

Se crea otro archivo para la creación del usuario que tendrá los permisos para conectarse desde keystone para hacer las lecturas de los otros usuarios del ldap.

```
dn: uid=svc-ldap,cn=users,cn=accounts,dc=identity,dc=venus-icc115,dc=net
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
uid: svc-ldap
sn: svc-ldap
cn: Service LDAP
userPassword: {SSHA}By3Sj7srspoDYsY54I7v+Gyzgh1Gi8q3

dn: cn=grp-openstack,cn=groups,cn=accounts,dc=identit,dc=net
objectClass: top
objectClass: groupOfNames
cn: grp-openstack
member: uid=svc-ldap,cn=users,cn=accounts,dc=identity,dc=venus-icc115,dc=net
```

Ejecutamos de igual forma el comando para importar los datos al servicio.

```
admin@identity:~$ ldapadd -x -D "cn=admin,dc=identity,dc=venus-icc115,dc=net" -W -f user_lookup.ldif
Enter LDAP Password:
adding new entry "uid=svc-ldap,cn=users,cn=accounts,dc=identity,dc=venus-icc115,dc=net"
adding new entry "cn=grp-openstack,cn=groups,cn=accounts,dc=identity,dc=venus-icc115,dc=net"
```

Adicionalmente se crea otro archivo para creación de otros usuarios, los cuales son los que se van a autenticar en keystone.

```
dn: uid=johndoe,cn=users,cn=accounts,dc=identity,dc=venus-icc115,dc=net
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
uid: johndoe
cn: John Doe
sn: Doe
givenName: John
mail: johndoe@venus-icc115.net
userPassword: {SSHA}1j4NFOo4aSjE7FRBpsyFKMH4FF/3yUBW

dn: uid=janedoe,cn=users,cn=accounts,dc=identity,dc=venus-icc115,dc=net
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
uid: janedoe
cn: Jane Doe
sn: Doe
givenName: Jane
mail: janedoe@venus-icc115.net
userPassword: {SSHA}1j4NFOo4aSjE7FRBpsyFKMH4FF/3yUBW
```

Importamos los usuarios igualmente.

```
admin@identity:~$ ldapadd -x -D "cn=admin,dc=identity,dc=venus-icc115,dc=net" -W -f users.ldif
Enter LDAP Password:
adding new entry "uid=johndoe,cn=users,cn=accounts,dc=identity,dc=venus-icc115,dc=net"
adding new entry "uid=janedoe,cn=users,cn=accounts,dc=identity,dc=venus-icc115,dc=net"
```

Configuración del certificado para comunicación

Copiamos el certificado del servicio ldap al servidor que contiene Keystone. Lo copiamos a la carpeta home del usuario.

```
admin@identity:~$ scp /etc/ssl/certs/ca-certificates.crt admin@main.venus-icc115.net:/home/admin
The authenticity of host 'db.venus-icc115.net (10.147.17.13)' can't be established.
ED25519 key fingerprint is SHA256:z9Aqa07VfSuYdTHhvh5OjRjZCSAn1jbnnPJB78jeN4M.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'main.venus-icc115.net' (ED25519) to the list of known hosts.
admin@main.venus-icc115.net's password:
ca-certificates.crt                                   100%  204KB  59.4MB/s   00:00
```

Generamos con el certificado *.crt* un archivo *.pem.*

```
admin@identity:~$ openssl x509 -in ca-certificates.crt -out ca-certificates.pem -outform PEM
```

Copiamos o movemos ambos archivos a la carpeta de certificados de Ubuntu.

```
admin@main:~$ sudo cp ca-certificates.pem /usr/local/share/ca-certificates/
admin@main:~$ sudo cp ca-certificates.crt /usr/local/share/ca-certificates/
```

Actualizamos el registro de certificados de Ubuntu.

```
admin@main:~$ sudo update-ca-certificates
```

Configuración de LDAP-Account-Manager

Instalación

Se instala *PHP* y el servidor web *Apache2.*

```
admin@identity:~$ sudo apt -y install apache2 php php-cgi \
libapache2-mod-php php-mbstring php-common php-pear
```

A continuación, active la extensión PHP *php-cgi.*

```
admin@identity:~$ sudo a2enconf php*-cgi
Enabling conf php8.1-cgi.
To activate the new configuration, you need to run:
  systemctl reload apache2
admin@identity:~$ sudo systemctl reload apache2
```

Instalamos el paquete principal de LAM.

```
admin@identity:~$ sudo apt -y install ldap-account-manager
```

Una vez instalado se recomienda configurar para restringir el acceso al panel únicamente a hosts o redes permitidas.

```
admin@identity:~$ sudo nano /etc/apache2/conf-enabled/ldap-account-manager.conf
```

En esta configuración, permitimos que se acceda desde la red Organización y a través de la red VPN comentando la línea *Require all granted* y agregando la siguiente.

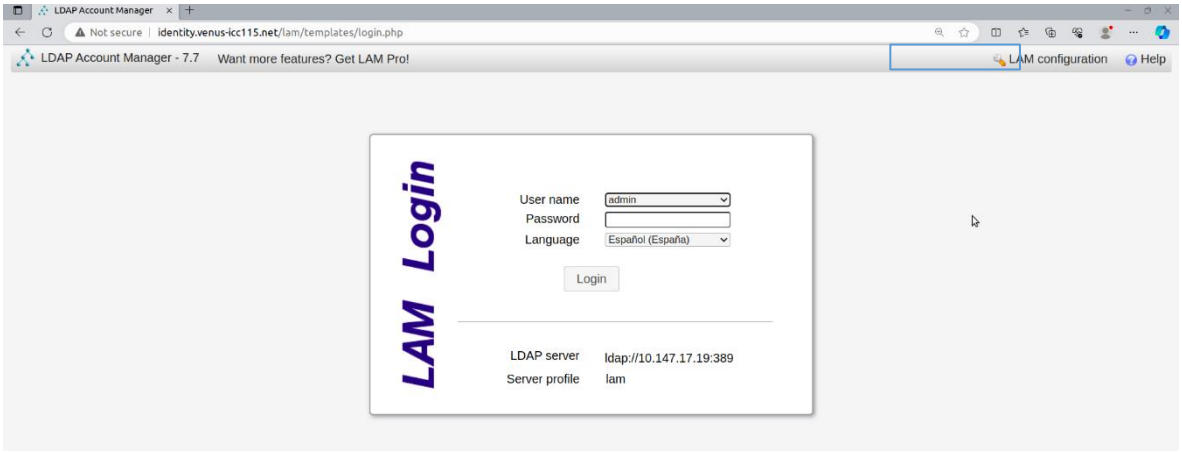
```
#Require all granted
Require ip 13.0.0.0/24 10.147.17.0/24
```

Para aplicar los cambios, reiniciar el servidor web *apache.*

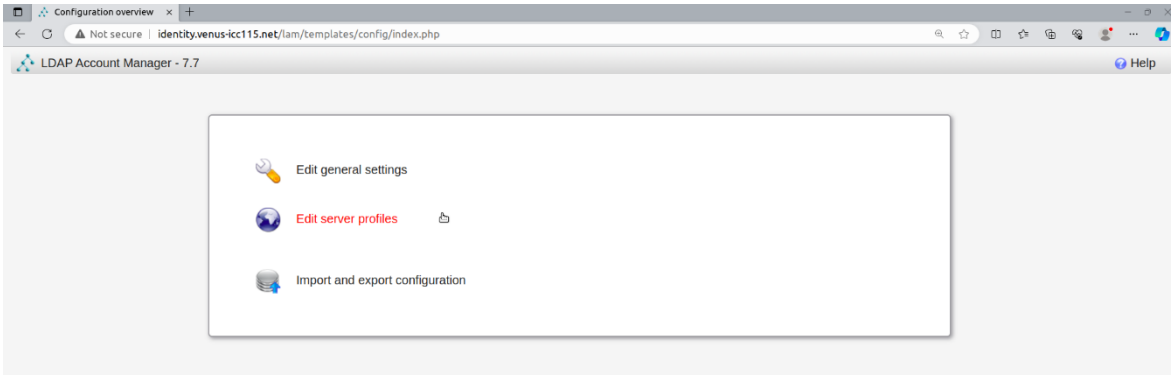
```
admin@identity:~$ sudo systemctl restart apache2
```

Configuración de Interfaz

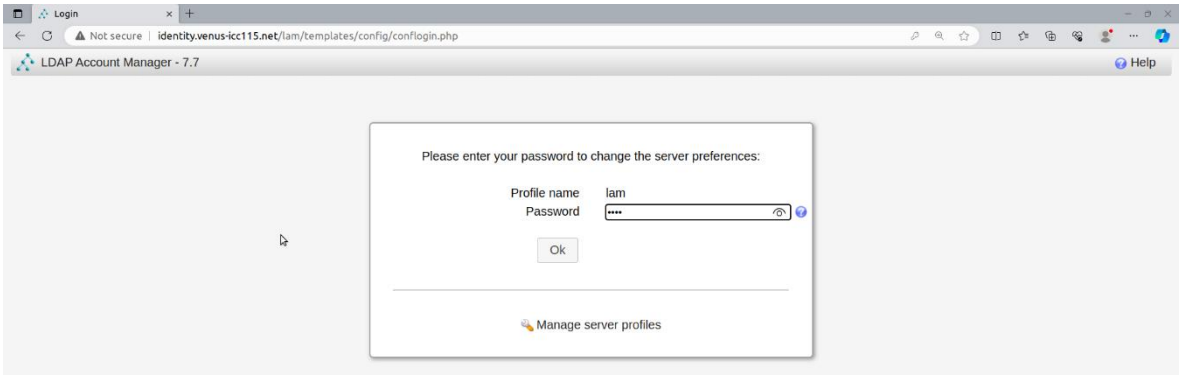
Para realizar la configuración de la interfaz LAM para la gestión y administración de LDAP, se da clic en la opción *LAM configuration*.



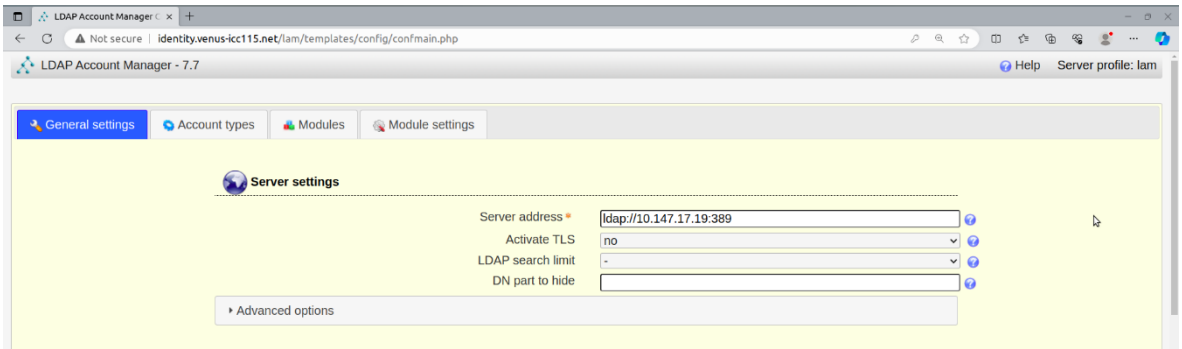
Se despliegan las siguientes opciones, de las cuales seleccionaremos la segunda: *Edit server profiles.*



Se solicitará ingresar una contraseña, en este caso se ingresará: *lam*, la cual es la contraseña por defecto de esta interfaz.

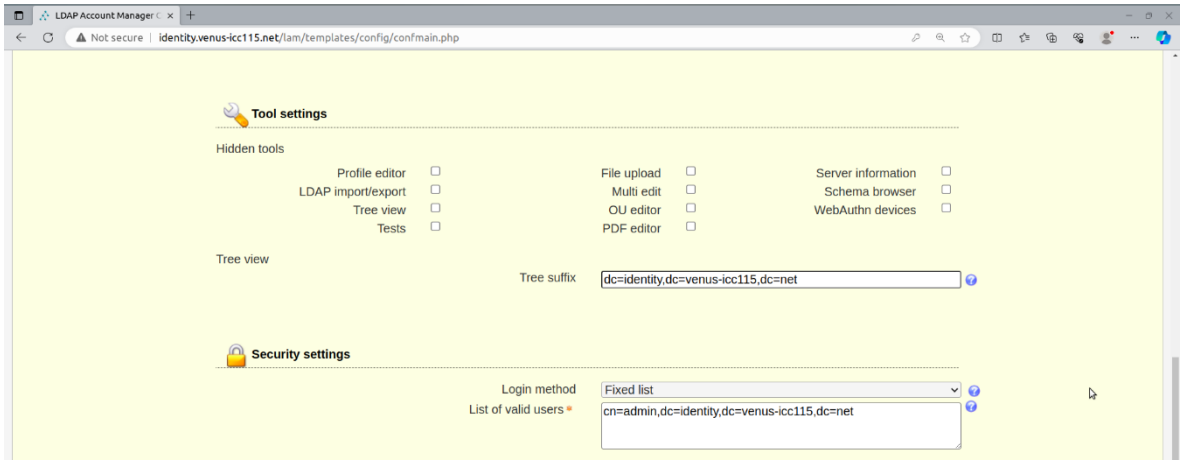


Se mostrarán las opciones de configuración del servidor, en nuestro caso en la parte de dirección del servidor, ingresamos la ip del servidor: *ldap://10.147.17.19:389* y dejamos las demás opciones como aparecen por defecto.



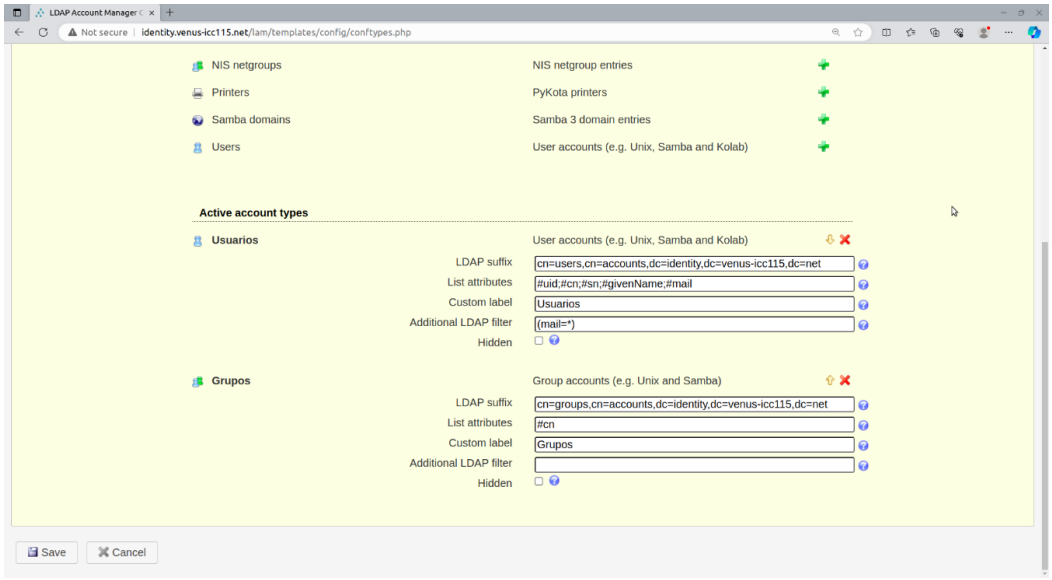
Especificamos el sufijo del árbol en la configuración de herramientas de la siguiente forma: *dc=identity, dc=venus-icc115, dc=net*.

Al igual que especificamos un método para el inicio de sesión en el apartado de configuración de seguridad, seleccionando *Fixed list*. En la misma sección especificamos la lista de usuarios válidos, para nuestro caso: *cn=admin, dc=identity, dc=net*.



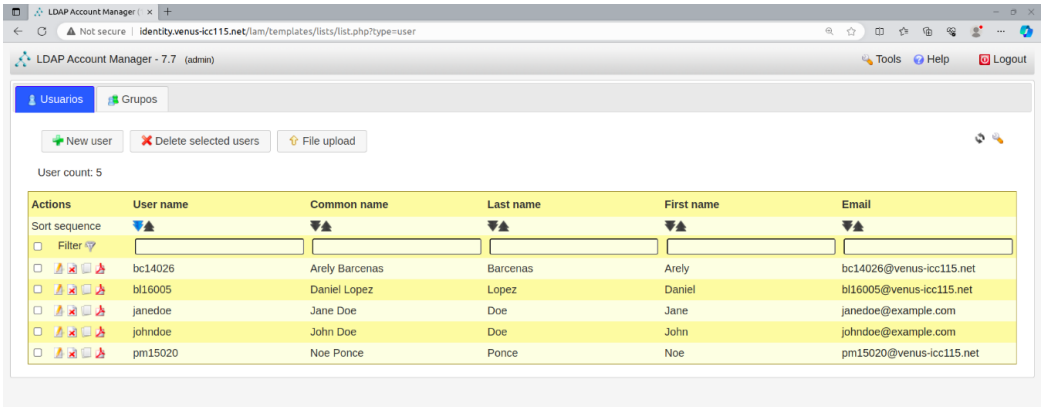
En la parte de configuración de los tipos de cuentas activas, especificamos los usuarios y grupos tal como se muestra en la figura.

Al estar seguros de la información que hemos ingresado, dar clic en el botón guardar.

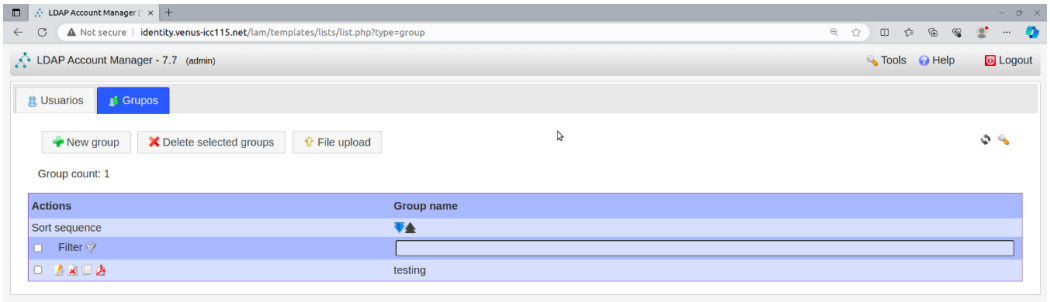


Al finalizar este proceso, se cerrará el panel de administración y se nos solicitará ingresar el usuario y la contraseña.

Luego del inicio de sesión se nos mostrará la lista de usuarios y las opciones para la gestión y administración de los mismos.



Al igual que se podrá observar la lista de grupos creados y las opciones para gestionar y administrar más grupos.



Configuración del Servidor Principal

OpenStack Controller Infraestructura

El nodo controller es el principal componente para dar el servicio de administración de regiones, por eso los recursos con los que cuenta debe ser suficiente para mantener sus servicios siempre disponibles.

Cantidad	Nombre	Recursos
1	Main	8GB RAM, 2 CPU, 50 GB Almacenamiento

El controller hace uso de una red física y una virtual. Se creó 3 redes para el funcionamiento del clúster.

Nombre de red	IPv4	Descripción
Infraestructura-Network	13.0.0.0/24	Red pública para la comunicación en los clientes y el clúster.
VPN	10.147.17.0/24	Red Privada Virtual por la cual se comunican las regiones con la infraestructura.

Asignación de IP.

Nodo	Interfaz	IP
Controller	enp1s0	13.0.0.13
Controller	zt6ntbqeau	10.147.17.13

Configuraciones Previas

Interfaces

Configuración del archivo netplan para definir la IP estática.

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp1s0:
      addresses:
        - 13.0.0.11/24
      routes:
        - to: default
          via: 13.0.0.1
      nameservers:
        addresses: [10.147.17.33]
```

Archivo de credenciales para openstack

Contienen las variables que se van a configurar en la sesión para usar con openstack.

```
admin@main:~$ 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://13.0.0.11:5000/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2
```

Base de Datos para Keystone

Se va a crear una base de datos en el clúster de la infraestructura para guardar los registros de keystone, como los endpoint, la información de usuarios, la información de dominio, regiones, proyectos, etc.

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

Instalación y configuración del servicio Keystone

Para poder instalar la última versión hasta la fecha, hace falta agregar manualmente el repositorio. La última versión hasta la fecha es *OpenStack Antelope* o 2023.1 y el repositorio para Ubuntu 22.04 es el siguiente.

```
admin@main:~$ sudo add-apt-repository cloud-archive:antelope
```

Luego de agregarlo, se actualiza la lista de paquetes de repositorios y se instala la última versión del paquete keystone.

```
admin@main:~$ sudo apt update
admin@main:~$ sudo apt install python3-openstackclient -y
admin@main:~$ sudo apt install keystone -y
```

Las líneas que modificar son las siguientes del archivo de configuración de keystone */etc/keystone/keystone.conf.*

```
[DEFAULT]
log_dir = /var/log/keystone
.
[database]
connection = mysql+pymysql://keystone:icc115@db.venus-icc115.net/keystone_infra
.
[extra_headers]
Distribution = Ubuntu
.
[token]
provider = fernet
.
```

Poblar la base de datos con las tablas y configuraciones básicas de keystone.

```
admin@main:~$ sudo su -s /bin/sh -c "keystone-manage db_sync" keystone
```

Inicializar los repositorios de llaves *fernet.*

```
admin@main:~$ sudo keystone-manage fernet_setup --keystone-user keystone --keystone-group keystone
admin@main:~$ sudo keystone-manage credential_setup --keystone-user keystone --keystone-group keystone
```

Arrancar el servicio de Keystone

Se inicializa keystone con los *endpoints* que van a servir para que los demás componentes se autentiquen.

```
admin@main:~$ sudo keystone-manage bootstrap --bootstrap-password icc115 \
  --bootstrap-admin-url http://main.venus-icc115.net:5000/v3/ \
  --bootstrap-internal-url http://main.venus-icc115.net:5000/v3/ \
  --bootstrap-public-url http://main.venus-icc115.net:5000/v3/ \
  --bootstrap-region-id Infraestructura
```

Configurar el servicio de Apache

Se modifica el archivo de configuración de apache.

```
admin@main:~$ sudo nano /etc/apache2/apache2.conf
```

Cambiar la línea o agregar si hace falta para el *Server Name.*

```
...
ServerName controller
...
```

Reiniciar el servicio de *apache.*

```
admin@main:~$ sudo service apache2 restart
```

Creación del dominio, proyecto y usuarios para keystone

Se va a crear el proyecto en el dominio que viene por defecto en *openstack*. Pero antes debemos cargar las credenciales de autenticación en el entorno.

```
admin@main:~$ . admin-openrc
```

Se crea un servicio que contiene los usuarios de los componentes que van a servir en OpenStack.

```
admin@main:~$ openstack project create --domain default --description "Service Project" service
```

Para verificar que la instalación y configuración de keystone ha sido correcta, creamos un token se sesión.

```
admin@main:~$ openstack token issue
```

Configuración Keystone para comunicarse con LDAP

Se debe crear un nuevo dominio en openstack para comunicarse con el servicio LDAP y autenticar sus usuarios.

Para configurar un nuevo dominio con ldap se empieza creando un directorio para los archivos de configuración de cada dominio.

```
admin@main:~$ sudo mkdir /etc/keystone/domains
admin@main:~$ sudo chown keystone /etc/keystone/domains
```

En el archivo */etc/keystone/keystone.conf* agregar las configuraciones en las secciones siguientes.

```
[assignment]
driver = sql
[identity]
domain_specific_drivers_enabled = true
domain_config_dir = /etc/keystone/domains
```

Creamos un nuevo dominio desde la CLI de openstack.

```
admin@main:~$ openstack domain create LAB
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
| enabled     | True                             |
| id          | 158f875d5b64ad1b2b9ac9dde6f5bca  |
| name        | LAB                              |
| options     | {}                               |
| tags        | []                               |
+-------------+----------------------------------+
```

Se edita el archivo de configuración del dominio creado (LAB).

```
admin@main:~$ sudo nano /etc/keystone/domains/keystone.LAB.conf
```

```
[ldap]
url =  ldap://identity.venus-icc115.net
user = uid=svc-ldap,cn=users,cn=accounts,dc=identity,dc=venus-icc115,dc=net
password = icc115
user_tree_dn = cn=users,cn=accounts,dc=identity,dc=venus-icc115,dc=net
user_objectclass = inetOrgPerson
user_id_attribute = uid
user_name_attribute = uid
user_mail_attribute = mail
user_pass_attribute =
user_allow_create = False
user_allow_update = False
user_allow_delete = False
tls_cacertfile = /usr/local/share/ca-certificates/ca-certificates.crt

[identity]
driver = ldap
```

Se cambia el propietario del archivo de configuración del dominio.

```
admin@main:~$ sudo chown keystone /etc/keystone/domains/keystone.LAB.conf
```

Acceso al nuevo dominio al usuario admin

Para otorgar permisos al usuario admin sobre el nuevo dominio LAB necesitamos obtener el ID del dominio LAB, el ID de usuario admin y el ID del rol “admin”.

```
admin@main:~$ openstack domain show LAB
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
| enabled     | True                             |
| id          | 9158f875d5b64ad1b2b9ac9dde6f5bca |
| name        | LAB                              |
| options     | {}                               |
| tags        | []                               |
+-------------+----------------------------------+
```

```
admin@main:~$ openstack user list --domain default
+----------------------------------+------------+
| ID                               | Name       |
+----------------------------------+------------+
| 79a7eb853a5c4ff2b6e889e9cba8171f | admin      |
| xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx | glance     |
| xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx | placement  |
| xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx | nova       |
| xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx | neutron    |
| xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx | skyline    |
+----------------------------------+------------+
```

```
admin@main:~$ openstack role list
+----------------------------------+--------+
| ID                               | Name   |
+----------------------------------+--------+
| 5700cd09f04945688b328edf342a5120 | admin  |
| xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx | member |
| xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx | reader |
+----------------------------------+--------+
```

Con esos identificadores correctamente definidos, damos acceso al dominio al usuario admin.

```
admin@main:~$ openstack role add --domain 9158f875d5b64ad1b2b9ac9dde6f5bca \
 --user 79a7eb853a5c4ff2b6e889e9cba8171f \
 5700cd09f04945688b328edf342a5120
```

Verificación de conexión y acceso

Se debe reiniciar al servicio Apache2.

```
admin@identity:~$ sudo systemctl restart apache2
```

Listamos los usuarios almacenados en el servidor ldap desde la CLI de openstack en el servidor Keystone.

```
admin@main:~$ openstack user list --domain LAB
+------------------------------------------------------------------+------------+
| ID                                                               | Name       |
+------------------------------------------------------------------+------------+
| 3958419e8cb0e041268dbcd331cb7e2048766f405ef2a9cf888b1142a34019e2 | svc-ldap   |
| 86169f0eac8660b9f86139d08392bc6ba261ee1429b2bf7634c576b146edcfaf | johndoe    |
| 337a33dfb93c2bc073eb3b98d9cb61a23a29ab3bab0123aa270b765263fced0e | janedoe    |
| 7664803beaac1dd9d25b606abeabec401c8fd3d4e3171be87182d6ed748407e6 | bc14026    |
| c70be4717251e693677a7377de39a75821647022250a30eed621487cb00092db | pm15020    |
| e08d8ecae1e1109094a1cb9c18f0d77e68401bcf6f79e107949088047793bb91 | bl16005    |
+------------------------------------------------------------------+------------+
```

Acceso a los usuarios LDAP sobre dominio default

Si se requiere que usuarios almacenados en LDAP tengan acceso y control sobre el dominio default y proyectos del dominio default, se debe asignar roles sobre estos a los usuarios LDAP.

Para asignar permisos de administrador al usuario johndoe se ejecuta los siguiente.

```
admin@main:~$ openstack role add \
--user 86169f0eac8660b9f86139d08392bc6ba261ee1429b2bf7634c576b146edcfaf \
--domain default admin
```

Verificar asignación de roles al usuario johndoe.

```
admin@main:~$ openstack role assignment list --user 86169f0eac8660b9f86139d08392bc6ba261ee1429b2bf7634c576b146edcfaf --names
+-------+-------------+-------+---------+---------+--------+-----------+
| Role  | User        | Group | Project | Domain  | System | Inherited |
+-------+-------------+-------+---------+---------+--------+-----------+
| admin | johndoe@LAB |       |         | Default |        | False     |
+-------+-------------+-------+---------+---------+--------+-----------+
```

Aun son eso, el usuario no puede entrar al panel Horizon ya que no tiene proyectos asignado. Se le asigna permisos sobre el proyecto admin.

```
admin@main:~$ openstack role add --user 86169f0eac8660b9f86139d08392bc6ba261ee1429b2bf7634c576b146edcfaf --project admin admin
```

Verificar nuevos permisos de johndoe.

```
admin@main:~$ openstack role assignment list --user 86169f0eac8660b9f86139d08392bc6ba261ee1429b2bf7634c576b146edcfaf --names
+-------+-------------+-------+---------------+---------+--------+-----------+
| Role  | User        | Group | Project       | Domain  | System | Inherited |
+-------+-------------+-------+---------------+---------+--------+-----------+
| admin | johndoe@LAB |       | admin@Default |         |        | False     |
| admin | johndoe@LAB |       |               | Default |        | False     |
+-------+-------------+-------+---------------+---------+--------+-----------+
```

Ahora el usuario tiene el rol de administrador para el dominio admin y sobre el proyecto admin.

Creación de los endpoint de regiones

Para que la infraestructura pueda tener acceso a los recursos que se encuentran en las regiones, se deben configurar endpoint que apuntan a las regiones.

Pero antes de agregarlos, debemos crear los servicios.

```
admin@main:~$ openstack service create --name glance --description "OpenStack Image" image
admin@main:~$ openstack service create --name placement --description "OpenStack Placement API" placement
admin@main:~$ openstack service create --name nova --description "OpenStack Compute" compute
admin@main:~$ openstack service create --name neutron --description "OpenStack Networking" network
```

Creación de endpoint para una región

Los endpoint del servicio *identity* que se crea para regiones, apuntan a la misma infraestructura, ya que será contra ésta con la cual se van a autenticar de ahora en adelante.

```
admin@main:~$ openstack endpoint create --region RegionOne identity public http://main.venus-icc115.net:5000/v3/
admin@main:~$ openstack endpoint create --region RegionOne identity internal http://main.venus-icc115.net:5000/v3/
admin@main:~$ openstack endpoint create --region RegionOne identity admin http://main.venus-icc115.net:5000/v3
```

Los otros endpoint, si deben apuntar a la región de la cual se quiere tener acceso a los recursos.

```
// GLANCE
admin@main:~$ openstack endpoint create --region RegionOne image public http://regionone.venus-icc115.net:9292
admin@main:~$ openstack endpoint create --region RegionOne image internal http://regionone.venus-icc115.net:9292
admin@main:~$ openstack endpoint create --region RegionOne image admin http://regionone.venus-icc115.net:9292

// PLACEMENT
admin@main:~$ openstack endpoint create --region RegionOne placement public http://regionone.venus-icc115.net:8778
admin@main:~$ openstack endpoint create --region RegionOne placement internal http://regionone.venus-icc115.net:8778
admin@main:~$ openstack endpoint create --region RegionOne placement admin http://regionone.venus-icc115.net:8778

// NOVA
admin@main:~$ openstack endpoint create --region RegionOne compute public http://regionone.venus-icc115.net:8774/v2.1
admin@main:~$ openstack endpoint create --region RegionOne compute internal http://regionone.venus-icc115.net:8774/v2.1
admin@main:~$ openstack endpoint create --region RegionOne compute admin http://regionone.venus-icc115.net:8774/v2.1

// NEUTRON
admin@main:~$ openstack endpoint create --region RegionOne network public http://regionone.venus-icc115.net:9696
admin@main:~$ openstack endpoint create --region RegionOne network internal http://regionone.venus-icc115.net:9696
admin@main:~$ openstack endpoint create --region RegionOne network admin http://regionone.venus-icc115.net:9696
```

Para otras regiones se cambia el parametro de la region y la URL, por ejemplo.

```
admin@main:~$ openstack endpoint create --region RegionTwo identity public http://main.venus-icc115.net:5000/v3/
admin@main:~$ openstack endpoint create --region RegionTwo identity internal http://main.venus-icc115.net:5000/v3/
admin@main:~$ openstack endpoint create --region RegionTwo identity admin http://main.venus-icc115.net:5000/v3

admin@main:~$ openstack endpoint create --region RegionTwo image public http://regiontwo.venus-icc115.net:9292
admin@main:~$ openstack endpoint create --region RegionTwo image internal http://regiontwo.venus-icc115.net:9292
admin@main:~$ openstack endpoint create --region RegionTwo image admin http://regiontwo.venus-icc115.net:9292
```

Configuración de Horizon

Instalación y Configuración

Ya que se configuró el repositorio de la versión de antelope, se puede instalar la última versión de Horizon.

```
admin@main:~$ sudo apt install openstack-dashboard -y
```

Abrimos el archivo de la configuración del dashboard.

```
admin@main:~$ sudo nano /etc/openstack-dashboard/local_settings.py
```

Los cambios necesarios en ese archivo son los siguientes.

```
ALLOWED_HOSTS = ['*']
SESSION_ENGINE = 'django.contrib.sessions.backends.cache'
CACHES = {
    'default': {
         'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',
         'LOCATION': 'cache.venus-icc115.net:11211',
    }
}
OPENSTACK_HOST = "main.venus-icc115.net"
OPENSTACK_KEYSTONE_URL = "http://%s:5000/v3" % OPENSTACK_HOST

TIME_ZONE = "America/El_Salvador"

OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True
OPENSTACK_KEYSTONE_MULTIREGION_SUPPORT = True

OPENSTACK_API_VERSIONS = {
    "identity": 3,
    "image": 2,
    "volume": 3,
}
OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = "Default"
OPENSTACK_KEYSTONE_DEFAULT_ROLE = "user"
```

Revisamos el archivo de configuración de apache para Horizon.

```
admin@main:~$ sudo nano /etc/apache2/conf-available/openstack-dashboard.conf
```

Se debe asegurar que esté la siguiente línea.

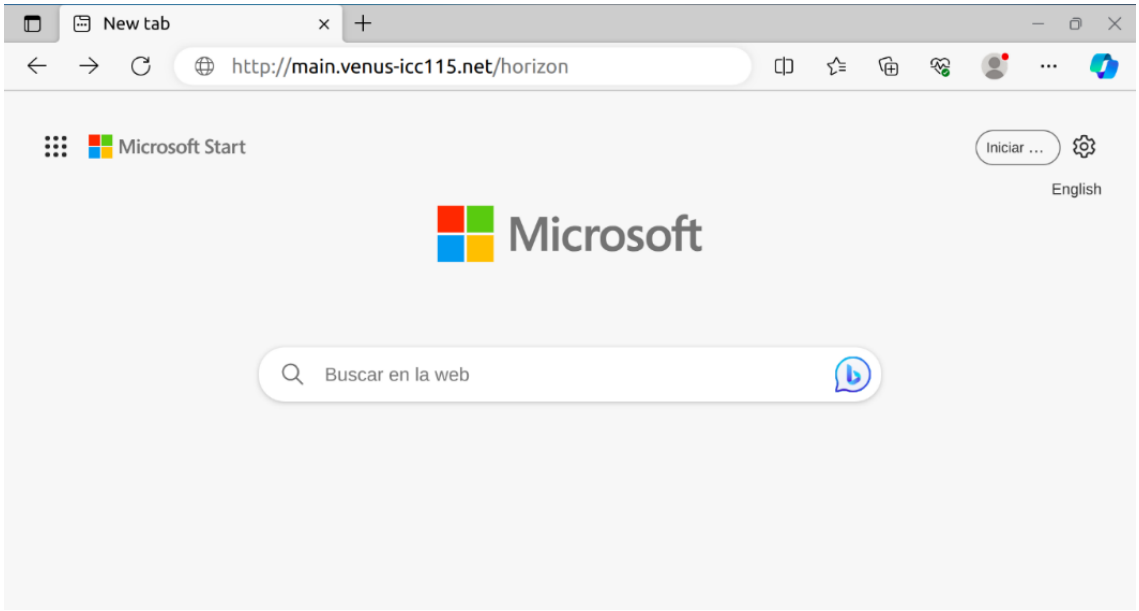
```
WSGIApplicationGroup %{GLOBAL}
```

Una vez confirmado, reiniciar apache.

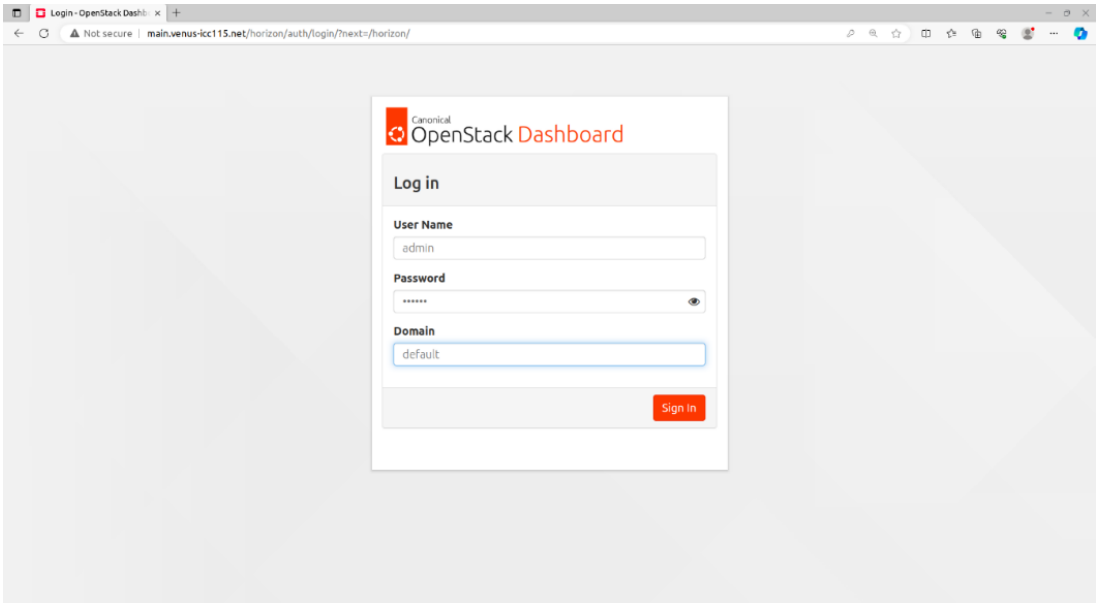
```
admin@main:~$ sudo systemctl reload apache2.service
```

Habiendo reiniciado el servicio de apache, solo resta ir al navegador y poner la dirección ip o nombre de dominio del nodo admin (controller de la infraestructura).

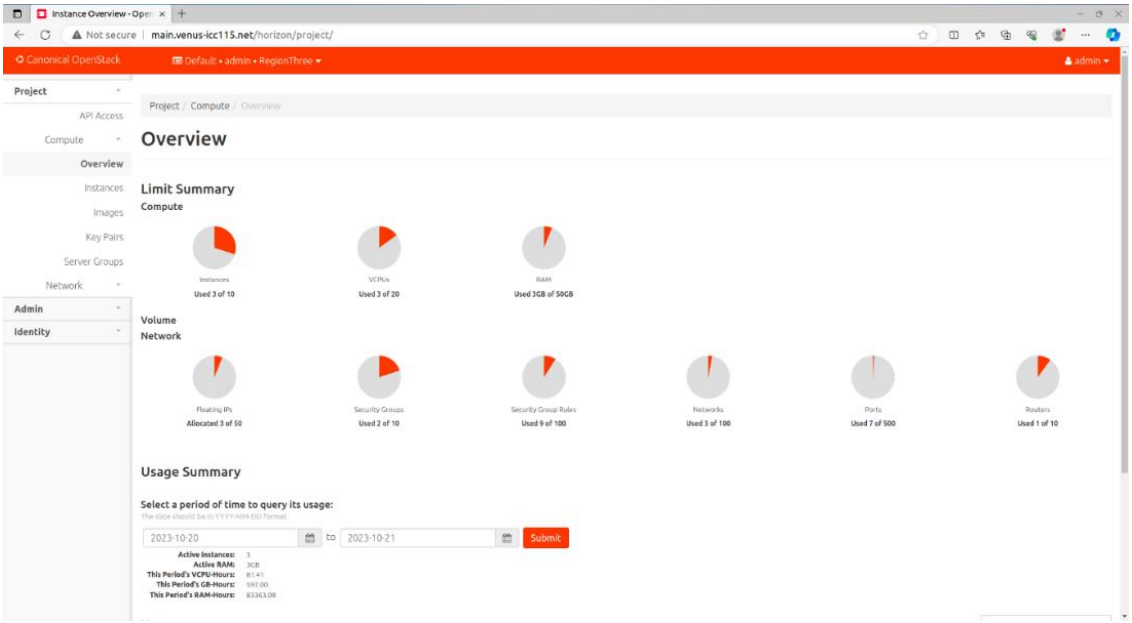
Si se tiene conexión al servidor DNS de la infraestructura simplemente es poner la dirección main.venus-icc115.net/horizon.



Carga la interfaz de Horizon y las credenciales son “admin” y la contraseña configurada en el comando Bootstrap o inicialización de keystone.



Una vez la sesión ha sido iniciada, muestra la pantalla de resumen de una de las regiones conectadas a la infraestructura.



Conectar Regiones a la Infraestructura

Regiones ya configuradas

Para unir regiones de nube con openstack completamente funcionales, aparte de cambiar los archivos de configuración, es necesario planificar los procedimientos de consumir los servicios de otros servidores.

En el caso de que una región esté usando una base de datos local y quiera usar la que se encuentra en la infraestructura, se necesita una buena planificación, respaldo.

Una vez se cuente con un plan y respaldos de los datos, se puede empezar a configurar los archivos de los componentes de openstack. El propósito es autenticarse en la infraestructura, y consumir los servicios.

Glance

Cambiamos los endpoint de la base de datos, y en este caso, el nombre de la base de datos ha cambiado para evitar posibles conflictos. Se le ha agregado el prefijo “regionone_” antes del nombre de la base de datos.

Se está consumiendo también el servicio de cache y los *endpoints* de autenticación apuntan al nodo *main* de la infraestructura que es el que tienen el componente de Keystone.

```
[DEFAULT]
...
[database]
connection = mysql+pymysql://glance:icc115@db.venus-icc115.net/regionone_glance
...
[keystone_authtoken]
www_authenticate_uri = http://main.venus-icc115.net:5000
auth_url = http://main.venus-icc115.net:5000
memcached_servers = cache.venus-icc115.net:11211
...
[oslo_limit]
auth_url = http://main.venus-icc115.net:5000
...
```

Placement

Al igual que Glance, la autenticación apunta el nodo main (controller) de la infraestructura, así como al servicio de base de datos (clúster de mariadb detrás de un balanceador de carga *maxscale*).

```
[DEFAULT]
...
[keystone_authtoken]
auth_url = http://main.venus-icc115.net:5000/v3
memcached_servers = cache.venus-icc115.net:11211
...
[placement_database]
connection = mysql+pymysql://placement:icc115@db.venus-icc115.net/regionone_placement
...
```

Nova

Para el componente de cómputo de openstack, se agrega además de la autenticación, la configuración de la cola de mensajes. Para el nuestro caso, el usuario que se ha creado en el servidor de cola de mensajes es único para cada región, en este caso "*regionone*" y también un *vhost* dedicado para cada región, en este caso el vhost es *"/regionone"*.

Nótese que en el caso del *vhost*, luego de poner el puerto hay dos barras (/) ya que una es para separar la parte del puerto, y la otra es parte del nombre de vhost.

Otra cosa que hay que tener en cuenta, es que si se cambia el nombre de la base de datos "nova" el nombre de la base de datos de célula "*cell0*" debe llevar el nuevo nombre de la base de datos nova, más el sufijo *"_cell0".*

```
[DEFAULT]
transport_url = rabbit://regionone:icc115@rabbit.venus-icc115.net:5672//regionone
...
[api_database]
connection = mysql+pymysql://nova:icc115@db.venus-icc115.net/regionone_nova_api
...
[database]
connection = mysql+pymysql://nova:icc115@db.venus-icc115.net/regionone_nova
...
[keystone_authtoken]
www_authenticate_uri = http://main.venus-icc15.net:5000/
auth_url = http://main.venus-icc15.net:5000/
memcached_servers = cache.venus-icc115.net:11211
...
[neutron]
auth_url = http://main.venus-icc15.net:5000
...
[placement]
auth_url = http://main.venus-icc15.net:5000/v3
...
[service_user]
auth_url = http://main.venus-icc15.net:5000/v3
...
```

Neutron

Al igual que nova, *neutron* tiene las mismas configuraciones, lo que resulta fácil encontrar los puntos que se van a cambiar.

```
[DEFAULT]
transport_url = rabbit://regionone:icc115@rabbit.venus-icc115.net:5672//regionone
...
[database]
connection = mysql+pymysql://neutron:icc115@db.venus-icc115.net/regionone_neutron
...
[keystone_authtoken]
www_authenticate_uri = http://main.venus-icc15.net:5000
auth_url = http://main.venus-icc15.net:5000
memcached_servers = cache.venus-icc115.net:11211
...
[nova]
auth_url = http://main.venus-icc15.net:5000
...
```

Horizon

Si la región ya cuenta con Horizon, debe cambiarse el host con el que hace la autenticación.

```
import os
...
CACHES = {
    'default': {
        'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',
        'LOCATION': 'cache.venus-icc115.net:11211',
    },
}
...
OPENSTACK_HOST = "main.venus-icc115.net"
OPENSTACK_KEYSTONE_URL = "http://%s:5000/v3" % OPENSTACK_HOST
...
```

Ahora, habiendo configurado las conexiones a la infraestructura, se están consumiendo los servicios desde la infraestructura.