

UNIVERSIDAD DE EL SALVADOR FACULTAD DE
INGENIERIA Y ARQUITECTURA
ESCUELA DE INGENIERIA DE SISTEMAS INFORMATICOS



**DISEÑO DE PROTOTIPO DE UNA INFRAESTRUCTURA
EN LA NUBE PARA MEJORAR EL MONITOREO DE
SERVICIOS Y ENTREGAS DE FUNCIONALIDADES A
LOS CLIENTES DE GRUPO TREMING S.A. DE C.V.**

PRESENTADO POR:

**JOSUÉ DAVID AQUINO BARRERA
ALEJANDRO JOSUÉ MARTÍNEZ LÓPEZ
JOSÉ GABRIEL PINEDA PEÑA
ERICK ALEXANDER QUINTANILLA PEÑATE**

PARA OPTAR AL TITULO DE:

INGENIERO DE SISTEMAS INFORMATICOS

CIUDAD UNIVERSITARIA, NOVIEMBRE DE 2022

RECTOR:

MSC. ROGER ARMANDO ARIAS ALVARADO

SECRETARIO GENERAL:

ING. FRANCISCO ANTONIO ALARCON SANDOVAL

FACULTAD DE INGENIERIA Y ARQUITECTURA

DECANO:

DOCTOR EDGAR ARMANDO PEÑA FIGUEROA

SECRETARIO:

ING. JULIO ALBERTO PORTILLO

ESCUELA DE INGENIERIA DE SISTEMAS INFORMATICOS

DIRECTOR:

ING. RUDY WILFREDO CHICAS VILLEGAS

FACULTAD DE INGENIERIA Y ARQUITECTURA ESCUELA DE
INGENIERIA DE SISTEMAS INFORMATICOS

Trabajo de Graduación previo a la opción al Grado de:

INGENIERO DE SISTEMAS INFORMATICOS

Título:

**DISEÑO DE PROTOTIPO DE UNA INFRAESTRUCTURA
EN LA NUBE PARA MEJORAR EL MONITOREO DE
SERVICIOS Y ENTREGAS DE FUNCIONALIDADES A
LOS CLIENTES DE GRUPO TREMING S.A. DE C.V.**

Presentado por:

**JOSUÉ DAVID AQUINO BARRERA
ALEJANDRO JOSUÉ MARTÍNEZ LÓPEZ
JOSÉ GABRIEL PINEDA PEÑA
ERICK ALEXANDER QUINTANILLA PEÑATE**

Trabajo de Graduación Aprobado por:

Docente Asesor:

MSc. DAMIÁN MORALES

CIUDAD UNIVERSITARIA NOVIEMBRE DE 2022

Trabajo de Graduación Aprobado por:

Docente Asesor:

MSc. DAMIÁN MORALES

Contenido

I.	INTRODUCCIÓN.....	1
1.	CONTEXTO DEL PROYECTO.....	3
1.1.	Antecedentes.....	3
1.2.	Pregunta de investigación.....	3
1.3.	Planteamiento del problema.....	3
1.4.	Objetivos.....	4
1.4.1.	Objetivo General.....	5
1.4.2.	Objetivos específicos.....	5
1.5.	Alcances.....	5
1.6.	Justificación.....	6
2.	FUNDAMENTACIÓN TEÓRICA.....	8
2.1.	Marco teórico.....	8
2.1.1.	Virtualización.....	8
2.1.2.	Tecnologías de virtualización.....	9
2.1.3.	Contenerización.....	10
2.1.3.1.	Contenedores.....	11
2.1.3.2.	Container Runtime Interface (CRI).....	14
2.1.3.3.	Orquestación de contenedores.....	15

2.1.3.4.	Kubernetes	16
2.1.3.5.	Rancher	20
2.1.4	Concepto de Cloud.	20
2.1.5	Modelos de despliegue de servicios Cloud.....	21
2.1.5.1	Cloud Computing modelo público.....	22
2.1.5.2	Cloud Computing modelo privado.	22
2.1.5.3	Cloud Computing modelo híbrido.....	22
2.1.5.4	Cloud Computing modelo comunitario.	23
2.1.5.5	Openstack y sus componentes.....	23
2.1.6	Infraestructura como servicio (IaaS).	38
2.1.6.1	Características de un IaaS.	39
2.1.6.2	Componentes de un IaaS.	40
2.1.6.3	Ventajas y Desventajas de un IaaS.	41
3.	METODOLOGÍA.	42
3.1.	Enfoque de la investigación.....	42
3.1.1.	Enfoque cuantitativo.....	43
3.1.2.	Enfoque cualitativo.	43
3.2.	Población y muestra.....	44
3.3.	Instrumentos.....	45

4.	RESULTADOS Y DISCUSIÓN.....	48
4.1.	Análisis de datos cualitativos.....	48
4.2.	Análisis de datos cuantitativos.	50
5.	DESARROLLO DE LA SOLUCIÓN PROPUESTA	55
5.1.	Caso de estudio Prototipo para una nube privada para la empresa Grupo Treming S.A. de C.V.....	55
5.2.	Herramientas utilizadas.....	55
5.2.1.	OpenStack.....	55
5.2.2.	Kubernetes.....	55
5.2.3.	Rancher.....	55
5.3.	Diseño del prototipo	56
5.4.	Construcción del prototipo.....	57
5.5.	Pruebas de funcionamiento.....	59
5.5.1.	Verificación de la instalación de openstack.....	64
6.	FACTIBILIDADES	67
7.	CONCLUSIONES.....	70
8.	REFERENCIAS BIBLIOGRÁFICAS.....	71
9.	ANEXOS.	75

Índice de Anexos

Anexo A Entrevista a CEO grupo TREMING S.A. de C.V.	75
Anexo B Entrevista al encargado de levantamientos de servicios del departamento de IT	77
Anexo C Cuestionario a departamento de implementación de Sistemas.....	79
Anexo D Cuestionario a departamento de desarrollo de sistemas	82
Anexo E Cuestionario a clientes Grupo TREMING S.A. de C.V.....	85
Anexo F Resultado Entrevista a CEO grupo TREMING S.A. de C.V.....	88
Anexo G Resultado Entrevista al encargado de levantamientos de servicios del departamento de IT	91
Anexo H Resultados cuestionario a departamento de implementación de Sistemas.....	93
Anexo I Resultados cuestionario a departamento de desarrollo de sistemas	97
Anexo J Resultados cuestionario a clientes Grupo TREMING S.A. de C.V.	101
Anexo K Cálculo de precios para nubes públicas y privada.....	104
Anexo L Guía de instalación Rancher en OpenStack.....	110

Índice de Ilustraciones

Ilustración 1 Arquitectura de Kubernetes	18
Ilustración 2 Arquitectura del servicio Horizon	25
Ilustración 3 Arquitectura del servicio Nova	27
Ilustración 4 Arquitectura del servicio Cinder	29
Ilustración 5 Arquitectura del servicio Neutron	31
Ilustración 6 Arquitectura del servicio Glance.....	33
Ilustración 7 Arquitectura del servicio Keystone	35
Ilustración 8 Arquitectura del servicio Octavia.....	38
Ilustración 9 Componentes de OpenStack recuperado de https://www.unixarena.com/2015/08/openstack-architecture-and-components-overview.html/	56
Ilustración 10 Arquitectura Rancher	57
Ilustración 11 Recursos de hardware a utilizar para la construcción del prototipo.	58
Ilustración 12 Servicios necesarios por máquina virtual.....	58
Ilustración 13 Creación máquina virtual	60
Ilustración 14 Instancia creada y visualizada desde Horizon	60
Ilustración 15 Creación de IP flotante para instancia creada.....	61
Ilustración 16 Ip flotante asignada a instancia	61
Ilustración 17 Grupo de seguridad asignado a la instancia	62
Ilustración 18 Acceso SSH a la instancia.....	63

Ilustración 19 Ping a Google.com	63
Ilustración 20 Login Dashboard Horizon	64
Ilustración 21 Endpoints de servicios.....	65
Ilustración 22 Verificación de contenedores.....	65
Ilustración 23 Herramienta instalada en contenedor de cluster kubernetes	66
Ilustración 24 Grafica de gastos acumulados de la solución por proveedores de servicios en la nube	68

AGRADECIMIENTOS.

Agradecimiento José Gabriel Pineda Peña

A mis padres

En primer lugar, agradezco a mis padres por todo el apoyo que me han brindado durante la carrera, gracias a ellos he podido lograr todos mis objetivos y metas. Con su cariño y soporte incondicional he salido adelante y he logrado todo lo que me he propuesto.

A mis compañeros

Gracias por acompañarme en este viaje, han sido muchos con quienes he compartido y hoy en día puedo llamar amigos, les tengo mucho aprecio y los llevare en mi corazón siempre, gracias por las horas compartidas, las desveladas juntos y todas las historias.

A mis docentes

Son muchos los docentes que han sido parte de mi camino universitario, y a todos ellos les quiero agradecer por transmitirme los conocimientos necesarios para hoy poder estar aquí. Sin ustedes los conceptos serían sólo palabras, y las palabras ya sabemos quién se las lleva, el viento.

Finalmente, a la casa de estudios

Por último, agradecer a la universidad que me ha exigido tanto, pero al mismo tiempo me ha permitido obtener mi tan ansiado título. Agradezco a cada directivo por su trabajo y por su gestión, sin lo cual no estarían las bases ni las condiciones para aprender conocimientos

Agradecimiento Alejandro Josué Martínez López

A mi madre

Por su paciencia y apoyo a lo largo de la carrera, siempre me brindo palabras de soporte y coraje para seguir adelante, junto a su cariño incondicional de madre.

A mi hermano

Fue mi guía en materias y momentos difíciles, dando claridad en mis problemas para poder tomar las mejores respuestas.

A mi prima

Por brindarme su apoyo en las noches de desvelos, escuchando mis quejas y dándome palabras de aliento para continuar por mis sueños.

A toda mi familia

Por su amor, apoyo y confianza, viendo en mí, el potencial que tengo y lo más importante, acompañándome en los buenos y malos momentos.

A mis compañeros

Agradezco su dedicación y enfoque el cual fue necesario para poder llegar a este punto, todas esas noches de desvelo y días de trabajo al final han dado su fruto.

Agradecimiento Josué David Aquino Barrera

Quiero primeramente agradecer profundamente a Dios Todopoderoso, que me ha regalado vida, salud y ayuda en cada una de las etapas de mi vida, por darme valor y perseverancia para terminar este camino.

A mis padres: Héctor Salomón y Maribel de los Ángeles, y a mi hermano Salomón los cuales me proveyeron con el cariño y la inspiración para seguir adelante con mis sueños, que con mucho esfuerzo y dedicación me apoyaron para lograr obtener un título universitario, por lo cual estaré agradecido el resto de mi vida.

Adicionalmente, agradezco el apoyo y la importancia en el transcurso de mi vida de mis abuelos: Miguel Barrera, Judith Larín, Josefina Aquino y Pompilio Larín, los cuales siempre estuvieron pendientes de mí y aunque algunos no estén presentes, sé que están felices y orgullosos desde el cielo, a los cuales les dedico este logro.

Dicen que de todas las posesiones la amistad es la más valiosa, es por ello que agradezco muchísimo a mis preciadas amistades: Luis Yanes, Kevin Hernández, Samantha Romero y Marvin Alvarenga, quienes me brindaron su apoyo en los momentos que los necesité y que compartieron conmigo una etapa tan importante en mi vida y en mi formación profesional, gracias a ellos he podido cumplir una meta más en mi vida y así como he aprendido y he sido inspirado por ellos espero que sepan lo mucho que valoro su amistad.

Así mismo agradezco a una persona en especial Celina Vizcarra, que me ha apoyado y es una persona que más admiro y aprecio por incentivar me a seguir adelante y porque nunca me dejó darme por vencido, así también por creer en mis sueños.

A mis compañeros de grupo con quienes logramos culminar juntos una etapa de nuestra formación profesional y quienes también ofrecieron sus conocimientos y me siento agradecido y honrado de poder terminar este proceso junto a colegas como ellos.

Por último, pero no menos importante al Ing. Napoleón Cornejo, director de GeoKapti, que ha sido y sigue siendo una inspiración y un ejemplo para seguir, que me dio una oportunidad y de quién he aprendido mucho y espero seguir aprendiendo, muchísimas gracias por creer en mí.

Agradecimiento Erick Alexander Quintanilla Peñate

A mis padres

Carlos Quintanilla y Marta Julia de Quintanilla quienes con su amor, paciencia y esfuerzo me han permitido llegar a cumplir hoy un sueño más, gracias por inculcar en mí el ejemplo de esfuerzo y valentía, de no temer las adversidades porque Dios está conmigo siempre.

A mi hermana

Karla Quintanilla por su cariño y apoyo incondicional, durante todo este proceso, por estar conmigo en todo momento.

A toda mi familia

Porque con sus oraciones, consejos y palabras de aliento hicieron de mí una mejor persona y de una u otra forma me acompañan en todos mis sueños y metas.

A mi pareja

Karla Méndez, por su apoyo, cariño y carisma que me hicieron levantarme cada vez que intente darme por vencido y estuvo ahí para darme el aliento necesario para seguir adelante en la búsqueda de mis sueños.

I. INTRODUCCIÓN.

El termino *nube* se utiliza como una metáfora de Internet, basado en el dibujo utilizado en el pasado para representar a la red telefónica, y más tarde para representar a Internet en los diagramas de redes de ordenadores.

La **computación en la nube** o *cloud computing*, hace referencia a los servicios que se ofrecen en la nube. Es la evaluación natural de la adopción generalizada de la virtualización y la arquitectura orientada a servicios. La idea básica es que los usuarios finales ya no necesitan tener conocimientos o el control sobre la infraestructura de tecnología en la nube que los apoya.

Por su parte la empresa Grupo Treming, S.A. de C.V., es una entidad que debido a las necesidades que presenta en cuanto a sus clientes y proyectos actuales, como a los futuros, la idea de tener una infraestructura en la nube podría brindarle grandes beneficios tanto a sus clientes como a sus empleados. Grupo Treming, se encarga al desarrollo, implementación y administración de Sistemas de Planificación de Recursos Empresariales, entre algunos de los beneficios podemos mencionar: Ofrecer a sus clientes una mayor confiabilidad y disponibilidad a las implementaciones, ya que al ser sistemas de gestión empresarial se requiere de una disponibilidad alta en todos los momentos de la jornada.

Del mismo modo el contar con entornos de desarrollo e implementación como lo son las Plataformas como Servicios o PaaS beneficiaría el proceso de estos departamentos, ya que este tipo de infraestructuras ofrecen beneficios a este tipo de negocios ya que en muchas ocasiones los requerimientos de software son similares entre todos los clientes.

En el desarrollo del documento se explicará de manera detallada el proceso de desarrollo de prototipo en la nube, en la que se harán dos tipos de implementaciones de servicios, los cuales son: Infraestructura como Servicios (IaaS) y Plataforma como Servicios (PaaS) haciendo uso de las mejores herramientas que nos permitan brindar una solución que permita desarrollar, desplegar y crear ambientes de desarrollo y de producción de forma rápida y eficiente.

1. CONTEXTO DEL PROYECTO.

1.1. Antecedentes

Grupo Treming S.A. de C.V. nace en el 12 de abril del 2010, fundada por William Contreras Dheming. En esos momentos no había en el Salvador empresas locales que atendieran las necesidades de consultoría SAP, es por eso por lo que nace dedicándose exclusivamente a la consultoría SAP. Actualmente es una empresa con tipo de sociedad anónima de capital variable y de tamaño mediana.

Grupo Treming tiene como misión ser una empresa exitosa de conformidad y tener una amplia red de consultores de clase mundial para satisfacer las necesidades de sus clientes y como visión Ser la mejor compañía de la región en soluciones tecnológicas y contribuir de esta forma a mejorar la calidad del trabajo de las personas.

1.2. Pregunta de investigación.

¿Qué grado de beneficio puede obtener Grupo Treming S.A. de C.V. al contar con una infraestructura en la nube que le permita agilizar el despliegue de entornos para sus empleados y sus clientes?

1.3. Planteamiento del problema.

A nivel mundial el uso de servicios sobre computación en la nube o *Cloud Computing* han venido tomando fuerza, según el informe de la IDC (Worldwide Semiannual Public Cloud Services Spending Guide), el crecimiento anual desde el 2018 es del 19.4%, impulsando a la industria con el uso de nubes

SaaS a tal punto que las grandes empresas consideran seguir invirtiendo en esta tecnología, y cada vez más, pequeñas y medianas empresas consideran aumentar la inversión que dedican a este rubro.

A su vez en El Salvador, es muy común que las empresas hagan uso de servicios en la nube proveídos por terceros, como lo pueden ser AWS, DigitalOcean, linode, entre otros, tanto para la publicación de sitios web como para administrar entornos de desarrollo para programadores e implementadores.

Partiendo de lo anterior, parece conveniente y una buena solución ofrecer una solución que satisfaga las necesidades de Grupo Treming, construyendo una nube de tipo IaaS que consistirá en proporcionar al consumidor de procesamiento, almacenamiento y otro tipo de recursos informáticos mediante internet, para que este pueda ser utilizado y ejecutado de forma arbitraria según las necesidades que se presenten en determinadas situaciones. Cabe recalcar que se espera que los usuarios finales tengan la capacidad de administrar aspectos como almacenamiento, aplicaciones y configuraciones básicas de red como puede ser el firewall del host.

A través de este tipo de nube, se espera que Grupo Treming pueda brindar servicios de infraestructura a sus clientes y también utilizar estos servicios para cubrir sus propias necesidades.

1.4. Objetivos.

A continuación, se expondrán los objetivos a llevar a cabo en el desarrollo de este documento, los cuales fueron escogidos tomando en cuenta la problemática actual y la visión final de la investigación.

1.4.1. Objetivo General.

Construir un modelo de prototipo de servicio Infraestructura como Servicio y Plataforma como Servicio, orientado a un despliegue que se ajuste a las necesidades identificadas, haciendo uso de herramientas que permitan una solución eficaz y aplicando conocimientos obtenidos sobre análisis y diseño de soluciones en la Nube en Grupo Treming S.A. de C.V.

1.4.2. Objetivos específicos.

- Diseñar el prototipo a construir tomando como base la información recolectada.
- Desarrollar un prototipo de nube privada tipo IaaS para satisfacer las necesidades identificadas en la institución.
- Desarrollar un prototipo de Plataforma como Servicio.
- Ejecutar pruebas de funcionalidad en los prototipos construidos.
- Aplicar técnicas de recolección de datos para obtener los requerimientos que ayuden a construir la solución planteada.
- Analizar la situación actual de Grupo Treming S.A. de C.V. a fin de conocer las necesidades que busca cubrir y poder solventarlas

1.5. Alcances

- Diseñar el prototipo de una infraestructura en la nube que ayude a agilizar las entregas de Grupo Treming S.A. de C.V.
- El prototipo abarca únicamente a los proyectos en desarrollo y pruebas de Grupo Treming S.A. de C.V.

- Entrega de manual de despliegue de Kubernetes en OpenStack que ayude a Grupo Treming S.A. de C.V. a desplegar un nuevo orquestador de contenedores con Rancher en caso sea necesario (Ver anejo L).

1.6. Justificación.

Actualmente Grupo Treming, se auxilia de las nubes públicas que existen en el mercado para configurar sus entornos de pruebas y para implementar las diferentes soluciones que se han gestionado para sus clientes, así como también para publicar sus propios sitios. Si bien estos servicios contratados brindan algunos beneficios como escalabilidad ilimitada, también generan cierto grado de incertidumbre en cuanto a calidad y seguridad debido a que la empresa no tiene un control total sobre la infraestructura, además el costo de estas no tiene fin ya que, aunque ofrecen servicio de escalabilidad ilimitada su pago es mensual o anual, lo cual representará un costo perpetuo, siempre y cuando, el servicio en cuestión sea utilizado, ya sea para host de los sitios empresariales o sitios contratados para uso de los clientes de Grupo Treming.

Partiendo de lo anterior, y dado que Grupo Treming tiene planes de crecer en las áreas de desarrollo e infraestructura, se ha considerado que esta podría mejorar su situación si contara con su propia nube, es decir, una nube privada que le permita tener un control total sobre dichos recursos y que a la vez sea lo suficientemente robusta como para soportar el tráfico que generan las aplicaciones actuales y las que tiene proyectado desarrollar. Así, con la presente investigación se pretende construir un prototipo de nube privada con un modelo de servicio IaaS que brinde una solución de hosting privado

para los clientes de la institución, así como PaaS para los entornos de prueba y desarrollo de uso interno, de esta forma creando una oportunidad a la empresa de sobresalir, respecto a sus principales competencias en el servicio de implementación y desarrollo de soluciones tecnológicas.

2. FUNDAMENTACIÓN TEÓRICA.

2.1. Marco teórico.

(Kavis, 2014) describe el concepto de nube de la siguiente manera, “hubo un tiempo en que todos los hogares, pueblos, granjas o pueblo tenía su propio pozo de agua. Hoy, los servicios públicos compartidos nos brindan acceso a agua limpia simplemente abriendo el grifo; la computación en la nube funciona de manera similar. Como el agua desde el grifo de su cocina, los servicios de computación en la nube pueden ser encendido o apagado rápidamente según sea necesario. como en el agua distribuida por empresas, hay un equipo de profesionales dedicados a hacer asegurarse de que el servicio proporcionado sea seguro y esté disponible 24/7. Cuando el grifo no está abierto, no solo estás ahorrando agua, también no estás pagando por recursos que actualmente no necesitas.”

A continuación, se hablarán de temas arraigados o muy conectados con los conceptos de nube, los cuales nos ayudarán a tener una mejor percepción de los distintos modelos de servicio de la nube, tipos de nube, contenedores y tipos de contenedores.

2.1.1. Virtualización.

La virtualización crea un entorno informático simulado, o virtual, en lugar de un entorno físico. A menudo, incluye versiones de hardware, sistemas operativos, dispositivos de almacenamiento, etc., generadas por un equipo. Esto permite a las organizaciones particionar un equipo o servidor físico en varias máquinas virtuales. Cada máquina virtual puede interactuar de forma independiente y ejecutar sistemas operativos o aplicaciones diferentes mientras comparten los recursos de una sola máquina host. (Microsoft Azure, 2022).

Para que esta virtualización se lleve a cabo se utiliza un software llamado “hipervisor” este se conecta con el hardware y permite dividirlo de manera segura, cabe recalcar que las máquinas virtuales (VM) dependen de su hipervisor para que tengan una capacidad adecuada para funcionar.

El hardware físico original en que está instalado el hipervisor se llama "host", y las VM que utilizan estos recursos se llaman "guests". Los guests utilizan los recursos informáticos, como la CPU, la memoria y el almacenamiento, como un conjunto de medios que pueden redistribuirse fácilmente. Los operadores pueden controlar las instancias virtuales de la CPU, la memoria, el almacenamiento y demás recursos, para que los guests reciban lo que necesiten cuando lo necesiten. (Red Hat, 2021).

2.1.2. Tecnologías de virtualización.

Las tecnologías de virtualización disponibles hacen uso del hipervisor, que no es más que la Plataforma de Virtualización misma, es decir, la tecnología que permite utilizar, a la vez, múltiples S.O. en un host anfitrión (Servidor físico).

Existen dos tipos de hipervisores:

- **Tipo 1 (Nativo, Bare-metal):** Estos Hipervisores son una solución de Software ejecutado directamente sobre el HW real del Host anfitrión (Servidor físico) para controlar el HW y monitorizar los múltiples S.O. virtualizados. Los sistemas virtualizados se ejecutan en otro nivel por encima del Hipervisor.
- **Tipo 2 (Hosted):** Estos Hipervisores son una solución de Software que se ejecuta sobre un S.O. convencional (Linux, Windows, Mac OS) para virtualizar otros Sistemas Operativos. De esta forma

la virtualización se produce en una capa más alejada del HW si se compara con los Hipervisores de tipo 1. Lógicamente esto hace que el rendimiento sea menor en los Hipervisores de Tipo 2.

2.1.3. Contenerización.

La contenerización, el acto por el que se crea un contenedor implica extraer solo la aplicación/el servicio que necesita ejecutar, junto con sus dependencias y su configuración, y separarlo del sistema operativo y de la infraestructura subyacente. La imagen de contenedor resultante luego puede ejecutarse en cualquier plataforma de contenedores. Varios contenedores pueden ejecutarse en el mismo hospedaje y compartir el mismo sistema operativo con otros contenedores, en donde cada uno ejecuta procesos aislados dentro de su propio espacio seguro. Debido a que los contenedores comparten el sistema operativo base, el resultado es poder ejecutar cada contenedor con menos recursos que si cada uno fuera una máquina virtual (VM) separada. (Rackspace, 2021)

La contenerización permite a los desarrolladores de software crear e implementar aplicaciones de forma más rápida y segura. Con los métodos tradicionales, puede desarrollar código en un entorno informático específico, lo que a menudo da como resultado errores cuando lo transfiere a una nueva ubicación. Por ejemplo, cuando transfiere código desde su computadora de escritorio a una máquina virtual o desde un sistema operativo Windows a Linux.

La contenerización elimina este problema, ya que permite agrupar el código de solicitud junto con sus archivos de configuración, dependencias y bibliotecas relacionados. Luego, abstrae ese único paquete de software (el contenedor) del sistema operativo host, lo que lo vuelve independiente y portátil; es decir, puede ejecutarse en cualquier plataforma o nube, sin problemas.

Aunque los conceptos de contenerización y aislamiento de procesos tienen décadas de antigüedad, la aparición de un motor de Docker de código abierto en 2013 aceleró la adopción de la tecnología de contenedores de aplicaciones. El motor de Docker se convirtió en un estándar de la industria para el proceso de contenerización con un enfoque de empaquetado universal y herramientas de desarrollo simples.

La industria a menudo se refiere a los contenedores como livianos, lo que significa que comparten el kernel del SO de la máquina y no requieren ninguna sobrecarga para asociar un SO dentro de cada aplicación, como en el caso de la virtualización. Por lo tanto, los contenedores tienen una capacidad inherentemente menor que una máquina virtual y requieren menos tiempo de inicio, lo que permite que más contenedores se ejecuten en una sola capacidad de cómputo como una sola MV. En consecuencia, esto genera una mayor eficiencia del servidor y reduce los costos de licencias y del servidor.

En pocas palabras, la contenerización permite a los desarrolladores escribir las aplicaciones una vez y ejecutarlas en todas partes. Ese nivel de portabilidad es esencial en términos del proceso de desarrollo y compatibilidad con los proveedores. También tiene otros beneficios, como aislamiento de errores, seguridad y facilidad de administración. (Veritas, 2022)

2.1.3.1. Contenedores.

Los contenedores permiten que una aplicación se ejecute de forma consistente y confiable, independientemente del sistema operativo o del ambiente de infraestructura. Los contenedores logran esto agrupando todo lo que un servicio necesita para ejecutarse (el código, el tiempo de ejecución, las

herramientas del sistema, las bibliotecas del sistema y la configuración) y crean un paquete portátil, independiente y ejecutable. (RackSpace, 2021)

Beneficios de los Contenedores

- **Los contenedores son ligeros:** Debido a que están aislados de la capa del sistema operativo, los contenedores son eficientes y requieren menos recursos en comparación con las máquinas virtuales.
- **Los contenedores son portátiles:** Debido a que incluyen todas las dependencias y configuraciones, usted puede desarrollar una vez y moverse entre ambientes.
- **Los contenedores son escalables:** Debido a su tamaño pequeño, los contenedores rápidamente se aceleran, escalan sobre la marcha, se desactivan cuando no están en uso y se reinician rápido cuando es necesario.
- **Los contenedores pueden ser rentables:** Gracias a la reducción de la demanda de recursos y al escalado inteligente, los contenedores ofrecen una solución resistente, dinámica y rentable.
- **Los contenedores requieren que maneje menos infraestructura:** Estos hacen que usted se concentre en los aspectos básicos de lo que realmente necesita, a fin de poder ofrecerles a sus clientes una experiencia que les brinde el mejor servicio. Esto hace que controlar la infraestructura sea más fácil, porque hay menos infraestructura que manejar.
- **Los contenedores ayudan a enfocarse:** Los equipos de TI dedicarán menos tiempo a los sistemas operativos invitados y al hardware físico, lo que les permite participar en proyectos importantes de la empresa.

- **Los contenedores aceleran el desarrollo:** Ofrecen un ambiente estable y predecible, donde la CPU/memoria se optimiza y el código se abstrae de la infraestructura para lograr la portabilidad.
- **Los contenedores aprovechan la arquitectura moderna:** Los programadores, gracias al uso de los contenedores, pueden descomponer aplicaciones en microservicios, lo que puede acelerar el desarrollo y, cuando se implementan, se pueden escalar de manera individual.

Desafíos de los contenedores

- **Los contenedores son relativamente nuevos:** Encontrar expertos en tecnología que sepan trabajar dentro de ambientes contenerizados puede ser una tarea difícil ya que se trata de una "tecnología de moda".
- **No todos los servicios están contenerizados:** Si su aplicación depende de servicios que no están contenerizados, es posible que necesite hacer una inversión importante para que se transforme en una solución de contenedores.
- **Los contenedores requieren cambios en los procesos y en las habilidades:** Estos pueden acelerar su transición a lograr un desarrollo más ágil y eficiente, pero, a la vez, puede significar cambios más importantes en sus procesos actuales de desarrollo, implementación, revisión y monitoreo. Del mismo modo, es posible que se necesite adaptar y volver a capacitar a los equipos existentes.
- **La tecnología evoluciona a gran velocidad:** Esto no solo se aplica a los contenedores, pero la naturaleza dinámica de la tecnología de los contenedores implica que necesita gente disponible (o socios) para tomar decisiones acertadas, reducir el riesgo y garantizar que la implementación no se vea obstaculizada por la inercia corporativa.

- **Los contenedores no son una solución mágica:** Es posible que los contenedores parezcan ideales, pero toda transición requiere de una profunda consideración. Debe entender con qué tiene que trabajar, qué funcionará y qué no, o buscar a alguien que lo ayude.

2.1.3.2. *Container Runtime Interface (CRI).*

Es un componente de software que puede ejecutar contenedores en un sistema operativo host. En una arquitectura en contenedores, los tiempos de ejecución de contenedores son responsables de cargar imágenes de contenedores desde un repositorio, monitorear los recursos del sistema local, aislar los recursos del sistema para el uso de un contenedor y administrar el ciclo de vida del contenedor.

Tipos de Container Runtime

- **De Bajo Nivel**
 - RunC: Creado por Docker y OCI es por ahora el runtime de bajo nivel de facto.
 - Crun: Una implementación de OCI liderada por RedHat y escrito en lenguaje C
 - Containerd: Es el runtime para Linux y Windows, el cual se encarga de administrar el ciclo de vida completo del contenedor de su sistema host
- **De Alto Nivel**
 - Docker(containerd): El sistema de contenedores líder, que ofrece un conjunto completo de funciones, con opciones gratuitas o de pago.
 - CRI-O: Ofreciendo una alternativa ligera a rkt y Docker. Le permite ejecutar pods usando tiempos de ejecución compatibles con OCI, brindando soporte principalmente para runC y Kata
 - Windows Contenedores y Hyper-V Contenedores: Dos alternativas ligeras a las máquinas

virtuales (VM) de Windows, disponibles en Windows Server. Los contenedores de Windows ofrecen abstracción (similar a Docker), mientras que Hyper-V proporciona virtualización. Los contenedores de Hyper-V son fáciles de transportar, ya que cada uno tiene su propio kernel, por lo que puede ejecutar aplicaciones incompatibles en su sistema host.

2.1.3.3. *Orquestación de contenedores*

La orquestación de contenedores consiste en la automatización de la mayoría de las operaciones necesarias para ejecutar cargas de trabajo y servicios en contenedores. En sistemas a gran escala, las aplicaciones contenerizadas se vuelven difíciles de gestionar manualmente ya suelen incluir cientos e incluso miles de contenedores. Así que, la orquestación de contenedores es esencial para reducir la complejidad operacional a la hora de ejecutarlos y brindar una solución escalable. Entre las ventajas de usar un orquestador de contenedores se encuentran:

- Permite escalar y reiniciar contenedores y clústers automáticamente.
- Mejora la seguridad al reducir las posibilidades de cometer errores humanos, gracias a la automatización.
- Ayuda a los equipos de IT a automatizar parte del trabajo y a aprovechar todos los beneficios de usar contenedores.
- Reduce la complejidad operacional a la hora de gestionar contenedores.

Entre los orquestadores de contenedores más populares se encuentran:

- **Kubernetes:** es una plataforma portable y extensible de código abierto para administrar cargas de trabajo y servicios. Kubernetes facilita la automatización y la configuración declarativa. Tiene

un ecosistema grande y en rápido crecimiento. El soporte, las herramientas y los servicios para Kubernetes están ampliamente disponibles.

- **Zun:** Es el servicio de contenedores de OpenStack. Provee una API dentro de OpenStack para correr aplicaciones en contenedores sin la necesidad de administrar servidores o clusters.
- **Docker Swarm:** es un software creado por los programadores de Docker que permite agrupar una serie de hosts de Docker en un clúster y gestionar los clústeres de forma centralizada, así como orquestar los contenedores.

2.1.3.4. *Kubernetes*

Kubernetes (también conocida como k8s o "kube") es una plataforma open source para la organización en contenedores que automatiza muchos de los procesos manuales involucrados en la implementación, la gestión y el ajuste de las aplicaciones que se alojan en ellos.

2.1.3.4.1. **Ventajas de K8's**

La ventaja principal de usar Kubernetes en su entorno es que obtiene una plataforma para programar y ejecutar los contenedores en clústeres de máquinas virtuales (VM) o físicas, especialmente si se encuentra en pleno proceso de optimización del desarrollo de las aplicaciones para la nube.

A grandes rasgos, le permite implementar una infraestructura basada en contenedores en los entornos de producción, y depender completamente de ella. Además, puede realizar muchas de las mismas tareas que haría con otras plataformas de aplicaciones o sistemas de gestión, pero en sus contenedores, ya que Kubernetes automatiza las tareas operativas.

Que se puede hacer:

- Organizar los contenedores en varios hosts
- Hacer un mejor uso del hardware para aprovechar al máximo los recursos necesarios en la ejecución de las aplicaciones empresariales
- Controlar y automatizar las implementaciones y actualizaciones de las aplicaciones
- Agregar almacenamiento para ejecutar aplicaciones con estado
- Ampliar las aplicaciones en contenedores y sus recursos según sea necesario
- Gestionar los servicios de forma declarativa para garantizar que las aplicaciones implementadas siempre se ejecuten correctamente
- Realizar comprobaciones de estado y autorregeneraciones de sus aplicaciones con ubicación, reinicio, replicación y adaptación automáticos

2.1.3.4.2. términos generales de K8's

- **Plano de control:** conjunto de procesos que controlan los nodos de Kubernetes. Es donde se originan todas las asignaciones de tareas.
- **Nodos:** máquinas que ejecutan las tareas solicitadas que asigna el plano de control.
- **Pod:** grupo de uno o más contenedores implementados en un solo nodo. Todos los contenedores de un pod comparten la dirección IP, la IPC, el nombre de host y otros recursos. Los pods extraen la red y el almacenamiento del contenedor subyacente, lo cual le permite trasladar los contenedores en el clúster con mayor facilidad.

- **Controlador de replicación:** controla la cantidad de copias idénticas de un pod que deben ejecutarse en algún lugar del clúster.
- **Servicio:** separa las definiciones de las tareas de los pods. Los proxies de servicios de Kubernetes envían las solicitudes de servicio al pod correspondiente de forma automática, sin importar a dónde se traslade en el clúster ni si se lo reemplaza.
- **Kubelet:** este servicio se ejecuta en los nodos y se encarga de leer los manifiestos del contenedor y de garantizar el inicio y el funcionamiento de los contenedores definidos.
- **kubectl:** es la herramienta de configuración de la línea de comandos de Kubernetes.

2.1.3.4.3. Funcionamiento de Kubernetes

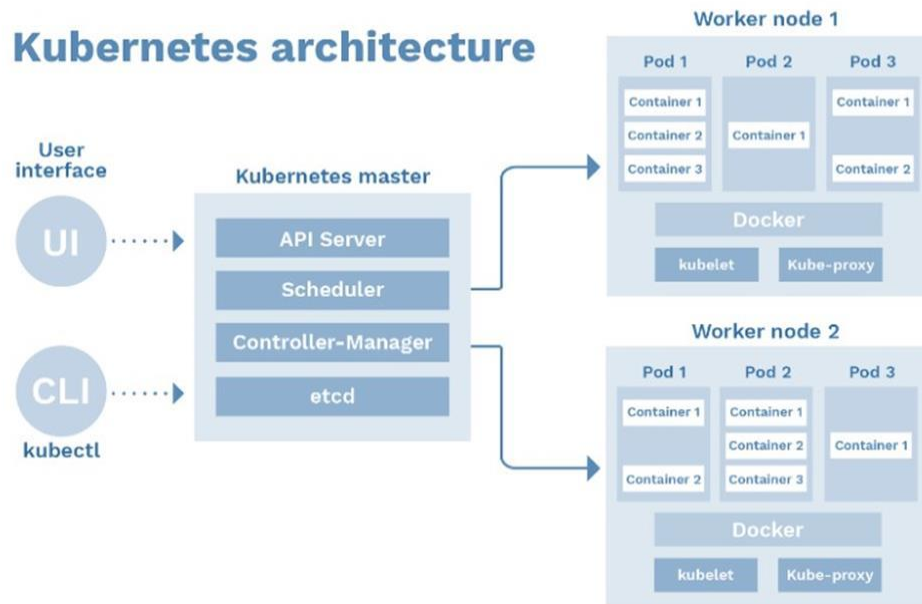


Ilustración 1 Arquitectura de Kubernetes

Una implementación de Kubernetes en funcionamiento se denomina clúster. Se puede visualizar en dos partes: el plano de control y las máquinas informáticas o los nodos. Cada nodo puede ser una máquina física o virtual y es su propio entorno Linux® que ejecuta los pods, los cuales están formados por contenedores. El plano de control es el encargado de mantener el estado deseado del clúster y de controlar, por ejemplo, las aplicaciones que se ejecutan y las imágenes de contenedores que se utilizan. Las máquinas informáticas son las que ejecutan las aplicaciones y las cargas de trabajo. Kubernetes controla el sistema operativo (por ejemplo, Red Hat® Enterprise Linux®) e interactúa con los pods de los contenedores que se ejecutan en los nodos.

El plano de control de Kubernetes recibe las instrucciones del administrador (o del equipo de DevOps) y las trasmite a las máquinas informáticas. Esta transferencia trabaja con una gran cantidad de servicios para decidir automáticamente cuál es el nodo más adecuado para la tarea. Luego, distribuye los recursos y asigna los pods que se encuentran en ese nodo para cumplir con la tarea solicitada.

El estado deseado de un clúster de Kubernetes define las aplicaciones o las cargas de trabajo que deben ejecutarse, junto con las imágenes que utilizan, los recursos que deben estar disponibles y otros ajustes similares. Desde el punto de vista de la infraestructura, esto no difiere mucho de la forma en la que gestiona los contenedores; solo que puede controlarlos en un nivel superior, lo cual le permite supervisar mejor los elementos sin la necesidad de gestionar cada contenedor o nodo por separado.

Usted se encarga de configurar Kubernetes y definir los nodos, los pods y los contenedores dentro de ellos, mientras que Kubernetes los organiza. Puede ejecutar Kubernetes donde quiera: en servidores dedicados (bare metal), en máquinas virtuales, en proveedores de nube pública, en nubes privadas o en

entornos de nube híbrida. Una de sus ventajas principales es que funciona en varios tipos de infraestructura.

2.1.3.5. *Rancher*

Rancher es un software para administrar clusters de Kubernetes en cualquier lugar con cualquier proveedor, eso incluye no solo la gestión de clusters existentes, sino que también la posibilidad de crear nuevos clústeres. Rancher facilita el aprovisionamiento y la administración de clústeres de Kubernetes. Se pueden importar clústeres existentes, personalizados o administrados, o bien definir e implementar tus propios clústers.

Rancher te permite administrar la seguridad de los clústeres fácilmente y permite definir definir usuarios, incluso implementar métodos de autenticación externos como LDAP, asignar permisos a estos usuarios y asignarlos a recursos en cualquier clúster administrado por Rancher.

2.1.4 Concepto de Cloud.

Durante años, los desarrolladores y administradores de red han representado Internet en los documentos de diseño como una nube. Al abstraer las tecnologías de Internet y los protocolos implícitos como una simple nube, los desarrolladores podrían ignorar temporalmente la complejidad de la comunicación y simplemente suponer que los mensajes fluirían con éxito de una red conectada a Internet a otra.

Hoy en día, el término computación en la nube describe la abstracción de computadoras, recursos y servicios basados en la web que los desarrolladores de sistemas pueden utilizar para implementar sistemas complejos basados en la web.

El National Institute of Standards and Technology (NIST, 2011) define computación en la nube como “un modelo para permitir el acceso a la red ubicuo, conveniente y bajo demanda a un conjunto compartido de recursos informáticos configurables que se pueden aprovisionar y liberar rápidamente con un esfuerzo de administración o interacción del proveedor del servicio”.

A menudo, estos recursos basados en la nube se consideran virtuales, lo que significa que, si un sistema o solución necesita más recursos, como procesadores o espacio en disco, los recursos se pueden agregar simplemente a pedido y, por lo general, de forma transparente a la aplicación que los usa. A través de su naturaleza virtual, las soluciones basadas en la nube se pueden ampliar o reducir en tamaño, y las empresas cuyas soluciones residen en las nubes normalmente pagan solo por los recursos que consumen.

Por lo tanto, las empresas que alguna vez dependieron de costosos centros de datos para albergar sus recursos de procesamiento ahora pueden cambiar sus costos y esfuerzos de mantenimiento a alternativas de pago por uso, escalables y basadas en la nube.

2.1.5 Modelos de despliegue de servicios Cloud.

Las aplicaciones basadas en la nube proporcionan una amplia gama de soluciones a un gran número de usuarios. De forma general se referirá a una solución en la nube en términos de su modelo de implementación y modelo de servicios. Estos dos términos se originaron en un documento de computación en la nube del Instituto Nacional de Estándares y Tecnología (NIST).

Un modelo de implementación en la nube especifica cómo se comparten los recursos dentro de la nube. Según (Jamsa, 2022), existen cuatro modelos principales de implementación de la nube: nube

privada, nube pública, nube comunitaria y nube híbrida. Cada modelo influye en la escalabilidad, confiabilidad, seguridad y costo correspondientes.

2.1.5.1 Cloud Computing modelo público.

La nube pública es el modelo común altamente conocido en la industria. Prácticamente toda la computación en la nube hoy en día se implementa de acuerdo con este modelo. Un usuario puede acceder y utilizar esto a través de la Internet pública. La diferencia entre ahora y cuando se hizo la definición es quizás una mayor capacidad para mantener la nube pública privada con la ayuda del cifrado de datos en tránsito y en reposo.

Este modelo está disponible para uso del público en general. Puede ser propiedad de una gran organización o empresa que ofrece servicios en la nube. Debido a su apertura, la nube puede ser menos segura. Una nube pública suele ser la solución menos costosa. (Marinescu, 2022)

2.1.5.2 Cloud Computing modelo privado.

La nube privada es cuando los servicios en la nube se ofrecen en una infraestructura privada. Si bien esta es ciertamente una opción teórica y algunas organizaciones han avanzado en la oferta de una gama limitada de servicios en la nube como una nube privada para desarrolladores internos, este no es un modelo generalizado y pierde la mayoría, si no todos, los beneficios de la nube. “Básicamente, es solo otra forma de ejecutar un data center en las instalaciones.” (Lisdorf, 2021)

2.1.5.3 Cloud Computing modelo híbrido.

El modelo híbrido es una combinación de dos o más de estos modelos. Debido a que los dos primeros son en su mayoría construcciones teóricas, este también es irrelevante al menos en su

formulación NIST. Sin embargo, la idea de una nube híbrida en un sentido diferente es mucho más común, ya que la mayoría de las organizaciones ejecutan infraestructuras híbridas con múltiples proveedores de nube y sus propios centros de datos (no en la nube). En cierto sentido, la nube híbrida está muy extendida. Simplemente no en la definición original ofrecida por NIST (2011).

2.1.5.4 Cloud Computing modelo comunitario.

Se habla de nube comunitaria cuando una comunidad específica de consumidores se une para construir una nube que puedan usar de manera similar a una nube de club privado. De acuerdo con Jamsa (2022), la nube es compartida por dos o más organizaciones, normalmente con intereses compartidos (como escuelas dentro de una universidad).

Hace una década con NIST (2011) se hablaba mucho más de él, pero hoy en día se implementa muy poca computación en la nube de esta manera. Al momento de esta investigación no se ha podido verificar ninguna implementación a gran escala existente de este tipo.

Ampliando la definición, se podría argumentar que los llamados GovClouds (Nube de Gobierno) de los grandes proveedores de nube actúan como nubes comunitarias. Los subconjuntos de sus ofertas en la nube están hechos a la medida de los clientes gubernamentales, por lo que podrían encajar en esta descripción. Sin embargo, también se ajustan a la descripción de una variante específica de una nube pública.

2.1.5.5 Openstack y sus componentes

Openstack es una plataforma de tecnología de código abierto que utiliza recursos virtuales agrupados para diseñar y gestionar nubes privadas e híbridas. Sus componentes se encargan de los

servicios principales de computación en la nube, brindando redes, almacenamiento, imágenes, identificación, etc.

Elementos:

1) Dashboard (Horizon)

OpenStack Dashboard proporciona una interfaz gráfica de usuario para que los usuarios y administradores realicen operaciones como crear y lanzar instancias, administrar redes y configurar el control de acceso.

Este servicio cuenta con los siguientes componentes:

Componente	Descripción
Openstack-dashboard	aplicación web desarrollada con Django, la cual proporciona acceso al panel de control desde cualquier navegador web
Apache HTTP server (httpd service)	host donde se encuentra la aplicación.

La siguiente imagen muestra de forma general la arquitectura del servicio Horizon:

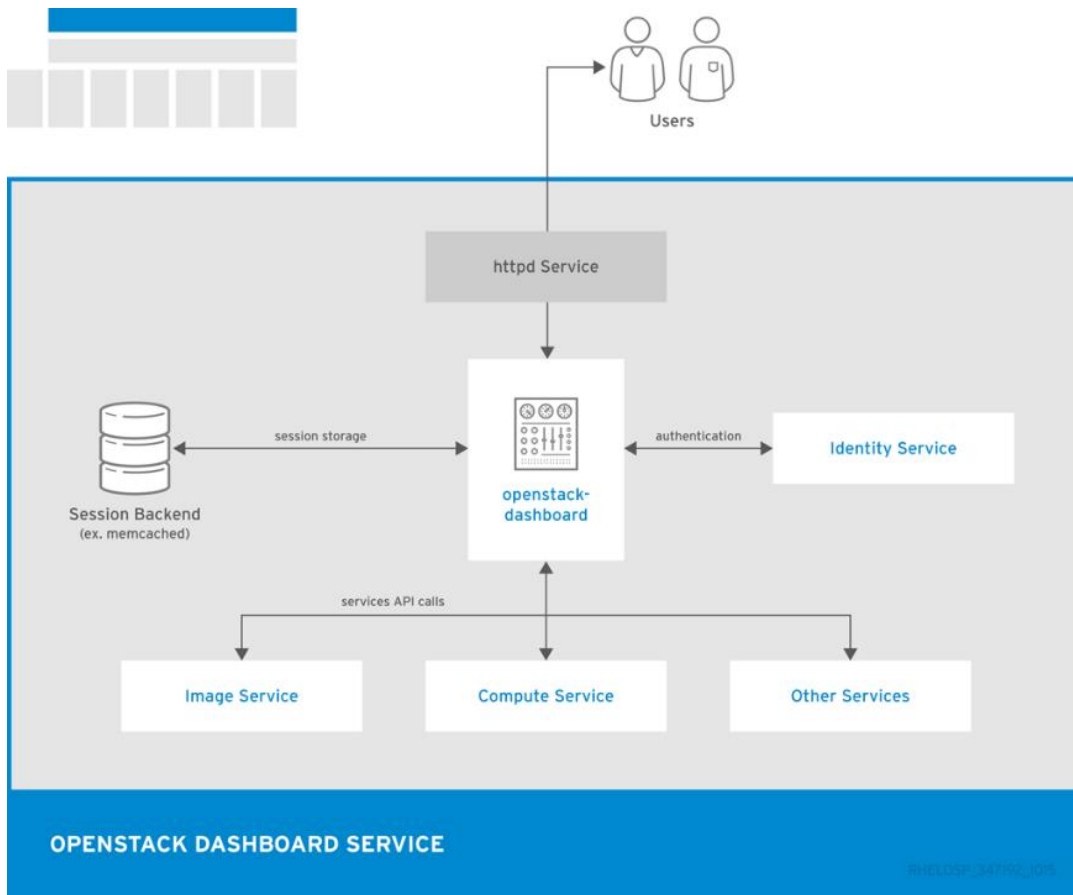


Ilustración 2 Arquitectura del servicio Horizon

2) Compute (Nova)

Nova brinda servicios de computación, haciendo uso de libvirt para comunicarse con el hypervisor KVM, núcleo de la operación de instancias virtuales bajo demanda. Proporciona un marco para el aprovisionamiento y la gestión a gran escala de instancias de cómputo virtual. Similar en funcionalidad y alcance al servicio EC2 de Amazon, le permite crear, administrar y destruir servidores virtuales basados en sus propias imágenes de sistema a través de una API programable haciendo uso del hypervisor.

Este servicio cuenta con los siguientes componentes:

Componente	Descripción
openstack-nova-api	Maneja solicitudes y proporciona acceso a los servicios de Compute, como el arranque de una instancia.
openstack-nova-cert	Proporciona el administrador de certificados.
openstack-nova-compute	Se ejecuta en cada nodo para crear y terminar instancias virtuales. El servicio de computación interactúa con el hipervisor para lanzar nuevas instancias.
openstack-nova-conductor	Proporciona a los nodos de cómputo acceso a la base de datos para reducir los riesgos de seguridad.
openstack-nova-consoleauth	Maneja la autenticación por consola.
openstack-nova-novncproxy	Proporciona un proxy VNC para que los navegadores permitan que las consolas VNC accedan a las máquinas virtuales
openstack-nova-scheduler	Envía solicitudes de nuevas máquinas virtuales al nodo correcto según los pesos y filtros configurados.
nova	Ciente de línea de comandos para acceder a la API de Compute.

La siguiente imagen muestra de forma general la arquitectura del servicio Nova:

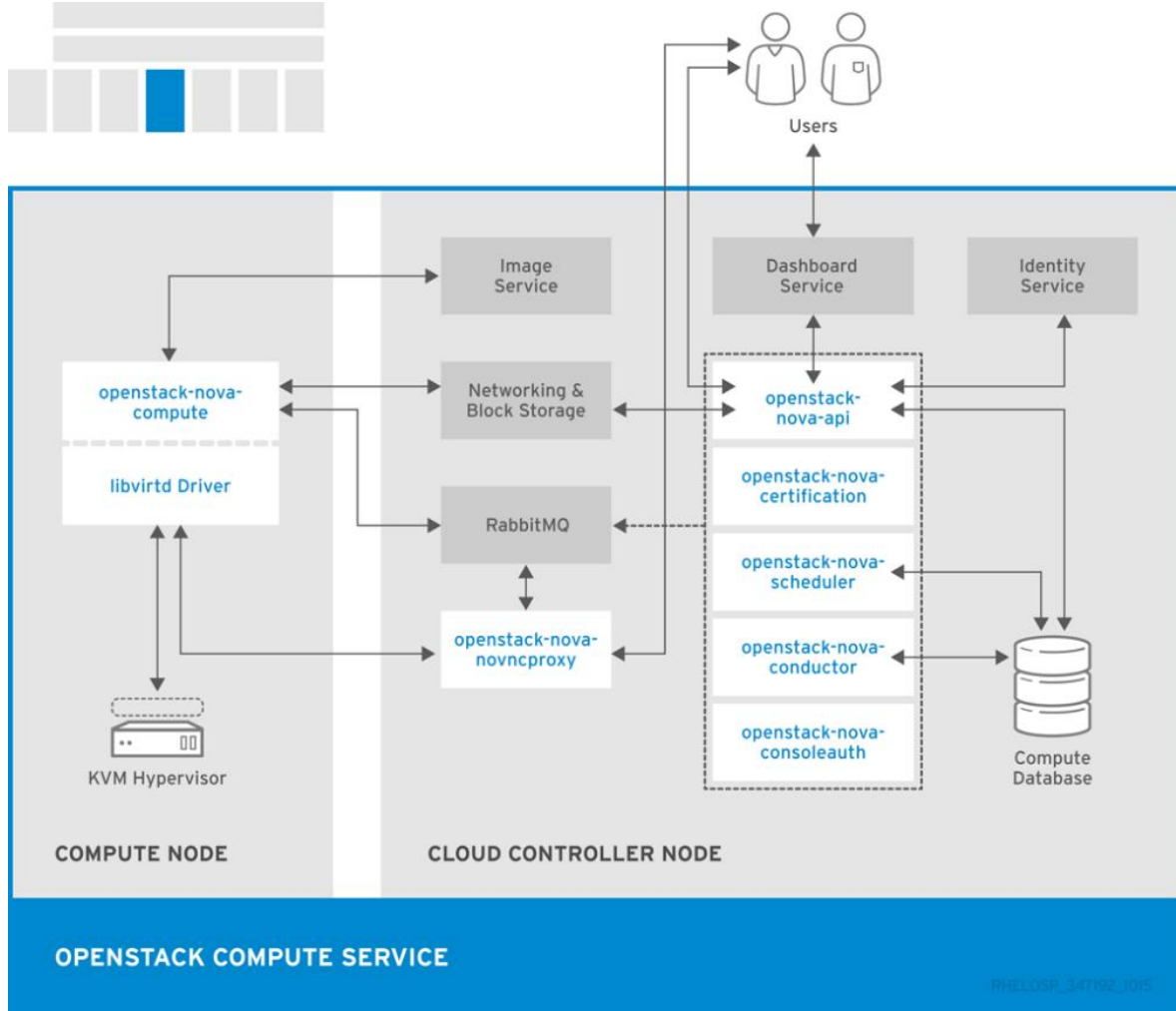


Ilustración 3 Arquitectura del servicio Nova

3) Block Storage (Cinder)

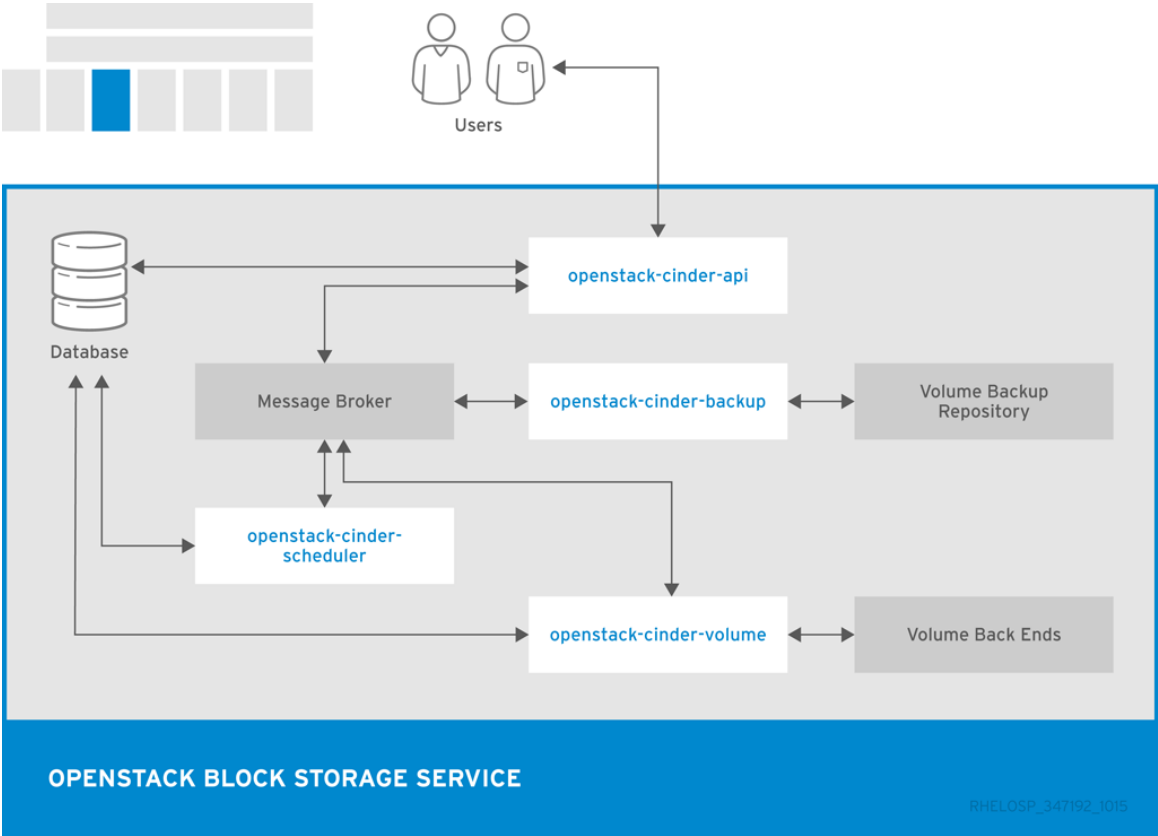
El Block Storage provee una gestión persistente para dispositivos de disco virtuales. Activa y permite a los usuarios crear y eliminar dispositivos de bloques, así como la asociación de dispositivos externos a servidores virtuales.

Este servicio cuenta con los siguientes componentes:

Componente	Descripción
openstack-cinder-api	Responde a las solicitudes y las coloca en la cola de mensajes. Cuando se recibe una solicitud, el servicio de API verifica que se cumplan los requisitos de identidad y traduce la solicitud en un mensaje que incluye la acción de almacenamiento en bloque requerida.
openstack-cinder-backup	Realiza una copia de seguridad de un volumen de Block Storage en un repositorio de almacenamiento externo. De forma predeterminada, OpenStack utiliza el servicio Object Storage para almacenar la copia de seguridad. También puede utilizar backends Ceph o NFS como repositorios de almacenamiento para copias de seguridad.
openstack-cinderscheduler	Asigna tareas a la cola y determina el servidor de volumen de aprovisionamiento.

<p>openstack-cinder-volume</p>	<p>Designa almacenamiento para máquinas virtuales. El servicio de volumen gestiona la interacción con los dispositivos de almacenamiento en bloque. Cuando llegan solicitudes del planificador, el servicio de volumen puede crear, modificar o eliminar volúmenes.</p>
<p>Cinder</p>	<p>Cliente de línea de comandos para acceder a la API de Block Storage.</p>

La siguiente imagen muestra de forma general la arquitectura del servicio Cinder:



4) Networking (Neutron)

Maneja la creación y administración de una infraestructura de red virtual en la nube de OpenStack. Los elementos de infraestructura incluyen redes, subredes y enrutadores. También puede implementar servicios avanzados como firewalls o redes privadas virtuales (VPN). OpenStack Networking proporciona flexibilidad para decidir qué servicios individuales ejecutar en qué sistemas físicos.

Este servicio cuenta con los siguientes componentes:

Componente	Descripción
Network agent	Servicio que se ejecuta en cada nodo de OpenStack para realizar la configuración de red local para las máquinas virtuales de nodo.
neutron-dhcp-agent	Proporciona servicios DHCP a redes de inquilinos.
neutron-ml2	Complemento que administra los controladores de red y proporciona servicios de enrutamiento y conmutación para servicios de red.
neutron-server	Servicio de Python que administra las solicitudes de los usuarios y expone la API de red.

Neutron	Cliente de línea de comandos para acceder a la API.
----------------	---

La siguiente imagen muestra de forma general la arquitectura del servicio Neutron:

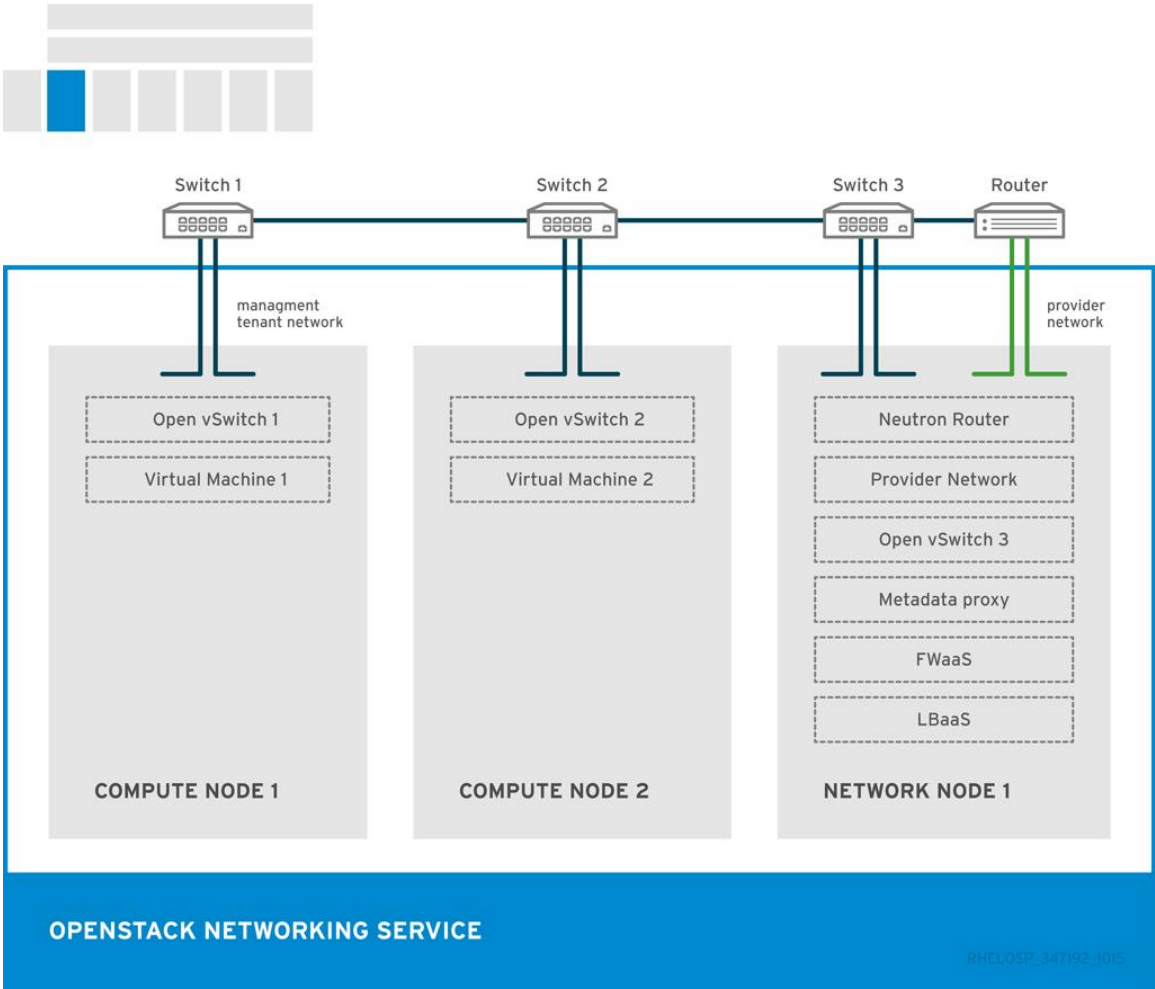


Ilustración 5 Arquitectura del servicio Neutron

5) Image Service (Glance)

actúa como un registro de imágenes de discos virtuales. Los usuarios pueden agregar nuevas imágenes o tomar una instantánea de un servidor existente para su almacenamiento inmediato.

Este servicio cuenta con los siguientes componentes:

Componente	Descripción
openstack-glance-api	Interactúa con los backends de almacenamiento para gestionar las solicitudes de recuperación y almacenamiento de imágenes.
openstack-glance-registry	Gestiona todos los metadatos de cada imagen.
glance	Cliente de línea de comandos para acceder a la API de imágenes.

La siguiente imagen muestra de forma general la arquitectura del servicio Glance:

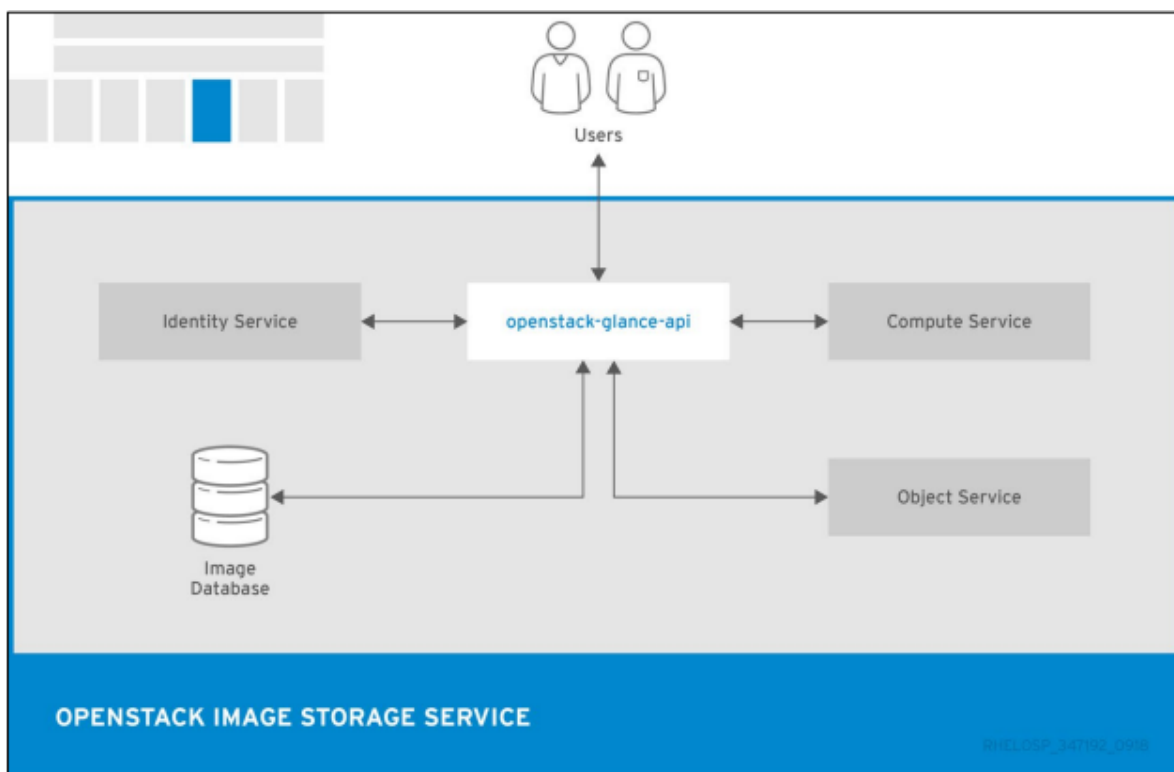


Ilustración 6 Arquitectura del servicio Glance

6) Identity Service (Keystone)

Identity proporciona autenticación y autorización de usuario a todos los componentes de OpenStack. Identity admite múltiples mecanismos de autenticación, que incluyen credenciales de nombre de usuario y contraseña, sistemas basados en tokens e inicios de sesión estilo AWS.

De forma predeterminada, el servicio de identidad utiliza un back-end MariaDB para la información de token, catálogo, política e identidad. Este back-end se recomienda para entornos de desarrollo o para autenticar conjuntos de usuarios más pequeños. También puede utilizar varios backends

de identidad al mismo tiempo, como LDAP y SQL. También puede usar Memcache o Redis para la persistencia del token.

Este servicio cuenta con los siguientes componentes:

Componente	Descripción
Openstack-keystone	proporciona servicios de identidad, junto con las API públicas y administrativas. Se admiten tanto la API de identidad v2 como la API v3.
Keystone	Cliente de línea de comandos para acceder a la API del servicio de identidad.

La siguiente imagen muestra de forma general la arquitectura del servicio Keystone:

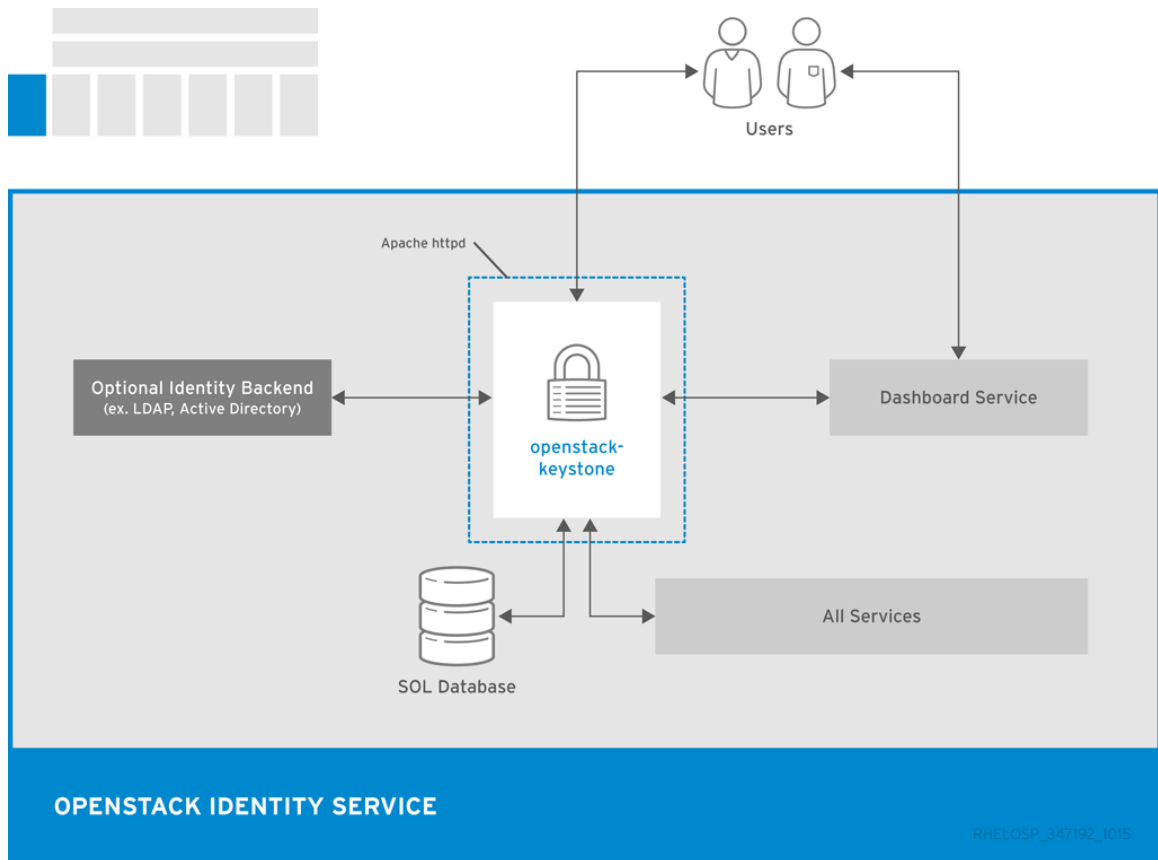


Ilustración 7 Arquitectura del servicio Keystone

7) Barbican

barbican es un servicio de gestión de claves de OpenStack que ofrece almacenamiento, aprovisionamiento y gestión seguros de datos clave. El servicio Barbican proporciona gestión de secretos, claves y certificados a través de múltiples back-ends de almacenamiento. barbican admite la generación de claves simétricas y asimétricas utilizando varios algoritmos.

Este servicio cuenta con los siguientes componentes:

Componente	Descripción
Barbican	una aplicación Web Server Gateway Interface (WSGI) que expone una API REST para secretos/contenedores/pedidos.
Barbican workers	Trabajadores para el procesamiento asíncrono, que se utiliza para diversas tareas de mensajería impulsadas por eventos relacionadas con la generación de certificados.

8) Octavia

Octavia es una solución de balanceador de carga de código abierto diseñado para funcionar con OpenStack. Octavia logra su entrega de servicios de balanceo de carga mediante la administración de un grupo de máquinas virtuales, contenedores o servidores en metal, conocidos colectivamente como "amphorae", que se activa bajo demanda. Esta función de escalado horizontal bajo demanda diferencia a Octavia de otras soluciones de balanceo de carga, lo que hace que Octavia sea verdaderamente adecuado "para la nube".

Este servicio cuenta con los siguientes componentes:

Componente	Descripción
amphorae	Son las máquinas virtuales, contenedores o servidores en metal individuales que cumplen con la entrega de servicios de balanceo de carga a los entornos de aplicaciones.
API Controller	Corre el API de Octavia. Acepta peticiones, las transforma y las enviar al controller worker a través del bus de mensajería de Oslo.
Controller Worker	Toma las peticiones provenientes del API controller y realiza las acciones necesarias para cumplir la petición.
Health Manager	Monitoriza los amphorae individuales para garantizar que están en funcionamiento y buen estado. También maneja eventos de error si un amphorae falla inesperadamente.
Housekeeping Manager	Se encarga de limpiar los registros de bases de datos obsoletos (eliminados) y

	administra la rotación de certificaciones de amphorae .
Driver Agent	Recibe estado y estadísticas de actualizaciones de los controladores.

La siguiente imagen muestra de forma general la arquitectura del servicio Octavia:

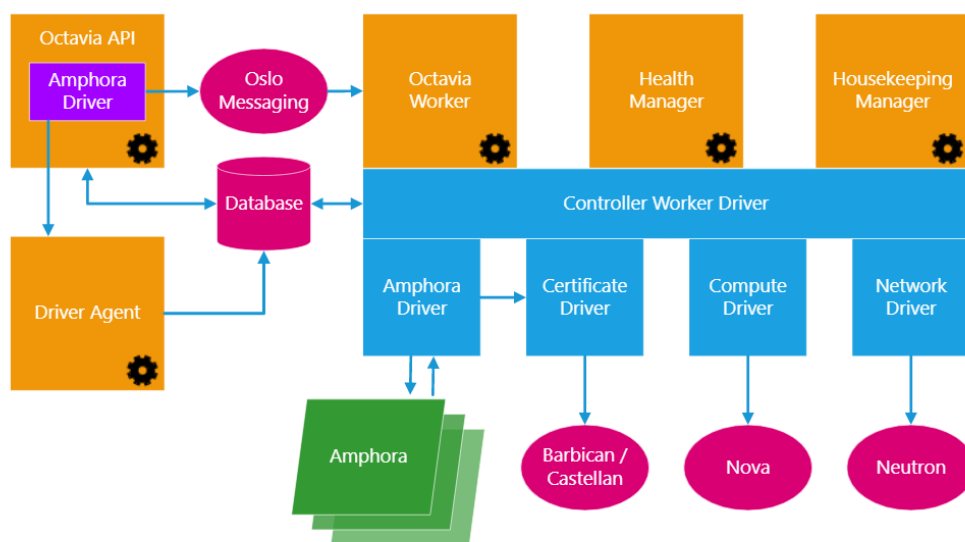


Ilustración 8 Arquitectura del servicio Octavia

2.1.6 Infraestructura como servicio (IaaS).

El modelo de infraestructura como servicio (IaaS) proporciona un centro de datos virtual dentro de la nube. IaaS proporciona servidores (físicos y virtualizados), almacenamiento de datos basado en la nube y más. Dentro de una solución IaaS, los desarrolladores deben instalar su propio sistema operativo,

software de administración de base de datos y software de soporte. Luego, los desarrolladores (o los administradores de sistemas de la empresa) deben administrar tanto el hardware como el software.

Lisdorf (2021) argumenta que la infraestructura como servicio (IaaS) es la forma más básica en la que se proporcionan los recursos informáticos básicos y el consumidor instala y administra el software necesario.

2.1.6.1 Características de un IaaS.

Manvi (2021) nos plantea que para ayudar a comprender el modelo IaaS, se tiene que examinar alguna de sus características claves, incluyendo el escalado dinámico, los niveles de servicio acordados, el alquiler, la medición y el autoservicio. Todas estas características son las mismas en entornos IaaS Cloud públicos y privados.

- **Escalado dinámico:** Una de las mayores ventajas de la IaaS es para las compañías que enfrentan la incertidumbre que los recursos puedan ser automáticamente escalados o reducidos dependiendo en los requerimientos de la aplicación. Si los clientes resultan que necesitan más recursos que los esperados, pueden obtenerlos de manera inmediata (hasta cierto límite, dependiendo del proveedor y del modelo de despliegue).
- **Niveles de servicio:** Los consumidores adquieren servicios IaaS de diferentes maneras. Muchos alquilan estos recursos basados en un modelo a demanda sin ningún contrato. En otras situaciones, el consumidor firma un contrato por una específica cantidad de almacenamiento, por ejemplo.

- **Modelo de alquiler:** Cuando las compañías usan IaaS, se tiene que en una cantidad considerable los servidores, almacenamiento u otra infraestructura de TI sea alquilada por una tarifa basada en la cantidad de los recursos usados y por cuánto tiempo se usen.
- **Medición y costos:** Del uso de IaaS se deriva un beneficio económico en potencia al controlar la cantidad de recursos que se necesitan y pagar por ellos. Para asegurar que los usuarios sean facturados solamente por los recursos que utilicen y soliciten, los proveedores de IaaS necesitan una forma constante y predictiva de medir el uso. Dicho proceso se llama **medición**.

2.1.6.2 Componentes de un IaaS.

IaaS se compone de una colección de recursos físicos y virtualizados que brindan a los consumidores los componentes básicos necesarios para ejecutar aplicaciones y cargas de trabajo en la nube. (IBM 2019):

- **Centros de datos físicos:** Los proveedores de IaaS manejan centros de datos gran capacidad, por lo general distribuidos alrededor del mundo, que contienen las máquinas físicas para soportar la carga de las diferentes capas de abstracción del proveedor.
- **Máquinas Virtuales / Poder computacional:** IBM (2019) asevera que IaaS es típicamente tomado en cuenta como recursos computacionales virtualizados. Los proveedores administran los hipervisores y los usuarios finales pueden aprovisionar "instancias" virtuales con las cantidades deseadas de cómputo y memoria (y, a veces, almacenamiento).

- **Red:** La creación de redes en la nube es una forma de redes definidas por software en la que el hardware de red tradicional, como enrutadores y conmutadores, está disponible mediante programación, generalmente a través de API.
- **Almacenamiento:** Los tres tipos principales de almacenamiento en la nube son el almacenamiento en bloque, el almacenamiento de archivos y el almacenamiento de objetos.

2.1.6.3 Ventajas y Desventajas de un IaaS.

En base a lo propuesto por Lisdorf (2021) y Marinescu (2022), IaaS provee el modelo más flexible y dinámico, con beneficios económicos para el usuario debido a su precio por consumo y su facilidad para desplegar hardware de forma automatizada. Al mismo tiempo, las tareas de mantenimiento son virtualizadas en su mayoría, lo que hace que los empleados tengan más tiempo libre para otras tareas (Lisdorf, 2021).

Sin embargo, Manvi (2021) también resalta que, dependiendo del proveedor, se requiere una curva de aprendizaje, que requiere de entrenamiento para el equipo encargado de la infraestructura al cambiar de un proveedor o al manejar nueva infraestructura. También cabe destacar los riesgos de seguridad de datos debido a la arquitectura de múltiples clientes a los que sirven los proveedores.

3. METODOLOGÍA.

En esta sección se describe el diseño de la metodología de Investigación a utilizar para la obtención de datos e información útil para lograr el objetivo de diseñar un prototipo de infraestructura en la nube para la empresa Grupo Treming.

3.1. Enfoque de la investigación.

La investigación propuesta se realizará con el enfoque cuantitativo, el concepto de enfoque cuantitativo tiene como base la comprobación de teorías por medio de la observación. De acuerdo con Hernández (2004) el enfoque cuantitativo consiste en un conjunto de procesos de orden riguroso y asimismo “parte de una idea que va acotándose y, una vez delimitada, se derivan objetivos y preguntas de investigación [...] De las preguntas se establecen hipótesis y se determinan variables; se traza un plan para probarlas”.

Se identificaron tres áreas principales a las cuales se les dará énfasis en la recolección de información y que además van a permitir validar los resultados de la investigación. Se detallan a continuación:

- Componentes de infraestructura a utilizar para la creación de la propuesta de solución de nube.
- Herramientas y componentes que faciliten el monitoreo y el mantenimiento de la solución propuesta.
- Consideraciones, procedimientos y guías de seguridad para proteger la información almacenada y procesada en la infraestructura de nube propuesta.

3.1.1. Enfoque cuantitativo.

Este documento adoptará mayormente el enfoque de investigación cuantitativo, primordialmente porque permite asignar un valores o grados representativos a las variables, sobre todo para cuantificar el rendimiento de la infraestructura a desarrollar, los cuales nos permiten tener un aspecto objetivo y externo que ayude al desarrollo de los objetivos de esta investigación.

Tal como lo indica Gómez (2006) que en el enfoque cuantitativo la recolección de datos es equivalente a medir, ya mencionado anteriormente, se plantea recolectar información no solo de métricas que se pueden obtener en la misma infraestructura, como también en las pruebas que se realizaran, asegurando que se obtenga una información descriptiva que se pueda representar con modelos numéricos para reflejar el cumplimiento de los objetivos.

3.1.2. Enfoque cualitativo.

Con este enfoque se conoce de primera mano la situación actual de la empresa, se debe tomar en cuenta la perspectiva de todos los individuos en cuanto a infraestructura en la nube se refiere y como esto puede ayudarles o retrasarles en sus tareas diarias, gracias a este enfoque se tiene información más precisa y con esta podemos idear unas solución que beneficie de mejor manera a la institución ya que se toma como base la opinión de todas las personas entrevistadas y no solo de la gerencia la cual puede opacar la opinión de los demás.

3.2. Población y muestra.

Haciendo uso de lo que indica (Corbetta, 2007) “El muestreo es el procedimiento por el cual, de un conjunto de unidades que forman parte del objeto de estudio (Población), se elige un número reducido de unidades (Muestra) aplicando criterios que permitan generalizar los resultados obtenidos del estudio de la muestra de toda la población.” (pág. 272).

Dado que la presente investigación se ha decidido realizar integrando componentes cualitativos y cuantitativos, es necesario extraer una muestra que nos permita representar la población con la finalidad de reducir toda la recolección de datos que se realizará para llevar a cabo los análisis de las soluciones aplicables.

De esta forma determinamos que la población objetivo de esta investigación está compuesta por todas aquellas personas que están relacionadas de alguna manera con Grupo Treming S.A. de C.V. aquí incluiremos a dos departamentos de la empresa, los cuales son el departamento de implementación de sistemas y el departamento de desarrollo, el cual asciende a una cantidad de 40 empleados, así como el banco de clientes que cuenta con aproximadamente 15 empresas.

De los datos anteriores, se eligieron parte de los empleados de ambos departamentos, así como al CEO de la empresa, totalizando una cantidad de 14 personas. Por parte de los clientes se escogieron a 5 los cuales forman parte del grupo de clientes a los cuales Grupo Treming les ofrece un servicio de nube publica, para el contacto con dichos clientes seremos apoyados por el jefe del departamento de IT. Se eligió al CEO de la empresa y al encargado de levantamiento de servicios para realizar las entrevistas ya que son los principales tomadores de decisiones de la empresa. Se seleccionaron 7 del departamento de

implementación de sistemas para realizar un cuestionario ya que ellos serían los encargados de mantener e implementar la solución propuesta en este documento y 5 del departamento de desarrollo para evaluar el tipo de entornos de desarrollo y pruebas que utilizan y como estos se integran con la nueva solución.

La selección de los empleados fue determinada haciendo uso de un muestreo aleatorio simple, con el cual se pretende que las respuestas no presenten un sesgo en la investigación.

3.3. Instrumentos.

Partiendo del hecho, de que la presente investigación contempla un enfoque cualitativo y cuantitativo es necesario también contemplar instrumentos para la recolección de datos que vayan acorde a cada enfoque, es por ello, que se han considerado los siguientes instrumentos.

4.3.1. Enfoque cualitativo.

Debido a que se considera pertinente que las personas entrevistadas tengan conocimiento sobre el contenido de dicha entrevista, se ha optado por realizar una entrevista como técnica para la recolección de datos en esta investigación. Esta entrevista será realizada mediante una plataforma de videoconferencias, esto como medida de seguridad ante el rebrote de Covid-19 que vive actualmente nuestro país.

Esta entrevista estará dirigida al CEO de Grupo Treming S.A. de C.V. y al encargado de levantamiento de servicios del departamento de TI, el objetivo de esta es conocer su opinión y percepción sobre la posibilidad de ofertar servicios de infraestructura en su empresa. La elección de estas personas se debió al cargo que ocupan dentro de la empresa y la importancia de incluirlos es la siguiente:

- **Encargado de levantamiento de servicios:** resulta de mucha importancia conocer la postura de la persona que hace el levantamiento de servicios para los clientes ya que este es el encargado de administrar los servicios de los clientes, esto lo obliga a estar familiarizado con las necesidades que se tienen en esta rama, por lo que su opinión puede marcar pautas que lleven al equipo de trabajo a encontrar la solución que mejor se adapte a las condiciones y recursos de la institución.
- **Director de la empresa:** resulta de mucho interés conocer la opinión del alto mando de Grupo Treming S.A. de C.V., con respecto a cómo consideran que administrar los servicios de la forma propuesta genera un añadido a los servicios que ofrece y por consiguiente lograr la atracción de más clientes.

5.3.2. Enfoque Cuantitativo.

A diferencia del enfoque cuantitativo, el enfoque cuantitativo busca recolectar datos utilizando como instrumento cuestionarios con preguntas cerradas, el objetivo de estos cuestionarios está encaminado a recolectar información en las siguientes áreas:

- **Empleados del departamento de desarrollo:** se han considerado debido a que usan continuamente los servicios de infraestructura brindados por la empresa para configurar los entornos de prueba y en el ciclo de desarrollo de los sistemas.
- **Empleados del departamento de implementación:** también resulta importante incluir a estos empleados debido a que ellos poseen ambientes y necesitan en ciertas ocasiones reiniciar y tener

interacción directamente con el servicio del cliente al cual tienen asignado como proyecto de implementación

- **Cientes de la empresa:** aunque la utilización de la nube privada en un principio está planeada para uso interno, la empresa también tiene planeado ofrecer posteriormente estos servicios a sus clientes, por lo que resulta importante conocer las preferencias de estos clientes en los servicios contratan y así tomar en cuenta estos resultados al momento de plantear la solución.

4. RESULTADOS Y DISCUSIÓN.

Para la obtención de información influyente en la solución se harán uso de herramientas de enfoque cualitativo con el cual se obtendrán las opiniones de los tomadores de decisiones de Grupo TREMING S.A. de C.V. respecto a contar con su propia nube privada y herramientas de enfoque cuantitativo hacia los departamentos de la empresa y clientes, los cuales tendrán consecuencias directas o indirectas al implementar una nube privada.

4.1. Análisis de datos cualitativos.

Para la recolección de los datos cualitativos, se hizo por medio de entrevistas al CEO de grupo TREMING S.A. de C.V. (Ver Anexo A) y al encargado de levantamiento de servicios (Ver Anexo B), ambos a través de la herramienta Google Meet en la semana del 22 de agosto de 2022 al 28 de agosto de 2022. Luego de realizarse las entrevistas se procedió con la transcripción de la entrevista al CEO de grupo TREMING S.A. de C.V. (Ver anexo F) y la entrevista al encargado de levantamiento de servicios (Ver Anexo G). Con estas respuestas se creó el siguiente diagrama FODA.



A continuación, se detalla la matriz FODA.

- **Amenazas**
 - **Poca aceptación de clientes:** Ya que los clientes pueden no estar satisfechos o no estén interesados por adquirir un servicio de nube privada por parte de la empresa.
- **Fortalezas**
 - **Apoyo de tomadores de decisión:** Están a favor de la implementación de una nube privada para la empresa.
 - **Presupuesto para ejecución:** La empresa se encuentra invirtiendo en diferentes tecnologías y hay disposición en invertir en una nube privada.
- **Debilidades**

- **Resistencia al cambio:** Implementar una nube privada significara cambios en procesos o hasta reestructuración de departamentos, y adaptarse a esto puede ser lento e improductivo.
- **Oportunidades**
 - **Valor agregado:** Se brindaría un nuevo servicio a los clientes, dando ventaja competitiva a la empresa.
 - **Seguridad y Control:** Al ser una nube privada, la empresa puede controlar por completo los temas de seguridad y control más convenientes.

4.2. Análisis de datos cuantitativos.

Los datos cuantitativos fueron obtenidos a través de la plataforma Google Forms, donde se crearon tres cuestionarios a los departamentos de implementación de sistemas (Ver anexo C), Desarrollo de sistemas (Ver anexo D) y a los clientes del Grupo Treming S.A. de C.V. (Ver anexo E). Los cuestionarios fueron contestados en la semana del 22 de agosto de 2022 al 28 de agosto de 2022.

Con los resultados del cuestionario al departamento de implementación de sistemas (Ver anexo H) se puede observar lo siguiente:

1. El 30% de los empleados del departamento de implementación tienen conocimientos básicos sobre infraestructura en la nube, un 50% de los empleados tienen conocimientos intermedios y un 20% de los empleados tienen conocimientos avanzados.
2. El 40% de los empleados está dispuesto a dedicar tiempo extra para mejorar sus conocimientos en infraestructura en la nube en caso desconozca algunos aspectos de este, uno 10% de los

empleados no está dispuesto a dedicar tiempo extra, un 40% de los empleados está dispuesto a dedicar tiempo extra, solo si la empresa los capacita y un 10% de los empleados está dispuesto a dedicar tiempo extra, solo si este fuera renumerado.

3. El 40% de los empleados considera que PaaS (Plataforma como Servicios) ayuda en la implementación de servicios a clientes y un 60% de los empleados considera que IaaS (Infraestructura como servicio) ayudaría en la implementación de servicios a clientes.
4. El 30% de los empleados consideran que no es necesario que la empresa posea su nube privada, un 10% de los empleados lo consideran poco necesario, un 30% de los empleados lo consideran necesario y un 30% de los empleados lo consideran muy necesario.
5. El 50% de los empleados consideran que el mayor inconveniente como departamento de implementación al implementar una solución de servicio IaaS sería tener personal poco capacitado, el 20% de los empleados considera que el mayor inconveniente es infraestructura interna actual insuficiente y al 30% de los empleados considera que el mayor inconveniente sería el poco interés por parte de los tomadores de decisiones.
6. El 10% de las aplicaciones se despliegan en Windows, un 30% de las aplicaciones se despliegan en Debian y un 60% de las aplicaciones se despliegan en Ubuntu Server.
7. El 40% de las aplicaciones usan entre 20 – 30 GB de disco duro, un 40% usan entre 30 – 50 GB de disco duro, un 10% usan entre 50-100 GB de disco duro y un 10% usan entre 100 – 200 GB de disco duro.
8. El 60% de las aplicaciones consumen entre 2 – 4 GB de RAM con baja demanda, un 30% consumen entre 4 – 6 GB de RAM y un 10% consumen entre 6 – 8 GB de RAM.

9. El 50% de las aplicaciones consumen entre 4 – 8 GB de RAM con alta demanda, un 40% consume entre 8 – 12 GB de RAM y un 10% consume entre 12 – 16 GB de RAM.

Con los resultados del cuestionario al departamento de desarrollo de sistemas (Ver anexo I) se puede observar lo siguiente:

1. El 25% de los empleados utiliza los servicios que contrata la empresa en la nube para configurar entornos de prueba y un 75% de empleados los utiliza para despliegue de aplicaciones
2. El 25% de los empleados experimenta fallas en los servicios contratados de forma poco frecuente y un 75% de los empleados los experimenta de forma frecuente.
3. El 25% de los empleados al contar con una nube privada de la empresa se beneficiaría con la creación de entornos de pruebas parecidos al entorno de producción, un 25% de los empleados se beneficiaría con una solución más rápida a problemas y un 50% de los empleados contaría con los dos beneficiados mencionados anteriormente.
4. El 50% de los empleados consideran que mejorara su desempeño con una nube privada, un 25% considera que mejorara su desempeño con una nube publica y un 25% le es indiferente el tipo de nube.
5. El 75% de los empleados desarrolla con el lenguaje Python y un 25% de los empleados desarrolla con el lenguaje Java.
6. El 25% de los empleados utiliza MariaDB como gestor de base de datos y un 75% usa PostgreSQL como gestor de base de datos.

7. El 25% de los empleados obtienen un entorno de pruebas al iniciar un nuevo proyecto entre 2-6 días, un 25% obtienen el entorno de prueba entre 1 – 2 semanas y un 50% obtienen el entorno de prueba entre 2 o más semanas.

Con los resultados del cuestionario a los clientes del Grupo Treming S.A. de C.V. (Ver anexo G)

se puede observar lo siguiente:

1. El 20% de los clientes tiene 2 aplicaciones funcionando actualmente, 40% de los clientes tiene 3 aplicaciones funcionando, 20% de los clientes tienen 4 aplicaciones funcionando y 20% de los clientes tiene 5 o más aplicaciones funcionando.
2. El 60% de los clientes casi nunca experimenta fallas en sus aplicaciones, un 20% experimenta fallas frecuentemente y un 20% de clientes experimenta fallas muy frecuentemente.
3. El 40% de los clientes considera justo el precio por el servicio recibido actualmente, un 40% de los clientes considera que actualmente está recibiendo un buen servicio, pero con costos altos y un 20% de los clientes considera que los costos están bien, pero el servicio es de mala calidad.
4. El 40% de los clientes está dispuesto a cambiar el servicio en donde corre sus aplicaciones mientras funciones bien y un 40% de clientes lo haría solo por otro mejor en razón calidad/ precio.
5. El 20% de los clientes considera que sus aplicaciones consumen muchos recursos, un 20% de los clientes considera que consumen una cantidad considerable, un 20% de los clientes considera que consumen pocos recursos y un 40% no está seguro sobre esto.
6. El 60% de los clientes están dispuestos de adquirir el servicio de nube privada de Grupo Treming S.A. de C.V. solo si es más barato que su proveedor actual, un 20% de los clientes se cambiaría

solo si no da tantos errores como su proveedor actual y un 20% de los clientes se cambiaría solo si hay una buena comunicación para el soporte del servicio.

7. El 20% de los clientes considera que necesita conocer los detalles técnicos de su servicio de nube contratado y un 80% considera que no es necesario conocer todos los detalles.

5. DESARROLLO DE LA SOLUCIÓN PROPUESTA

5.1. Caso de estudio Prototipo para una nube privada para la empresa Grupo Treming S.A. de C.V.

Se construirá un prototipo compuesto de una nube privada capaz de brindar infraestructura y plataforma como servicio con el fin de ayudar o facilitar el despliegue de diferentes aplicaciones web, instancias de bases de datos, así como también diferentes herramientas por medio de un orquestador de contenedores. La solución planteada contara con herramientas de monitoreo en tiempo real para asegurar la disponibilidad de la solución el mayor tiempo posible sin errores.

5.2. Herramientas utilizadas.

5.2.1. OpenStack

Openstack es la plataforma de cómputo en la nube de software libre más importante y con mayor crecimiento en los últimos años. Cuenta con diferentes componentes instalados en conjunto o separados los cuales cubren funciones específicas, capaces de satisfacer las necesidades requeridas.

5.2.2. Kubernetes

Kubernetes es una plataforma de software libre con una comunidad muy grande la cual cuenta con un gran soporte. Permite administrar y escalar aplicaciones y servicios contenerizados.

5.2.3. Rancher

Permite la administración de cluster de kubernetes, administración de usuario y políticas, además facilita la integración con software de métricas y notificaciones.

5.3. Diseño del prototipo

El prototipo planteado será implementado a través de máquinas virtuales corriendo sobre el hipervisor KVM en un sistema operativo Linux Ubuntu Server 20.04. En las maquinas virtuales se encuentra instalado y configurado OpenStack el cual nos permitirá gestionar nubes privadas y brindar un modelo de infraestructura como servicio, y sobre OpenStack se encuentra la herramienta Rancher, la cual nos permitirá crear y gestionar clusters de kubernetes con el cual brindaremos un modelo de plataforma como servicio.

La siguiente imagen muestra los componentes de OpenStack y su comunicación a través de APIs.

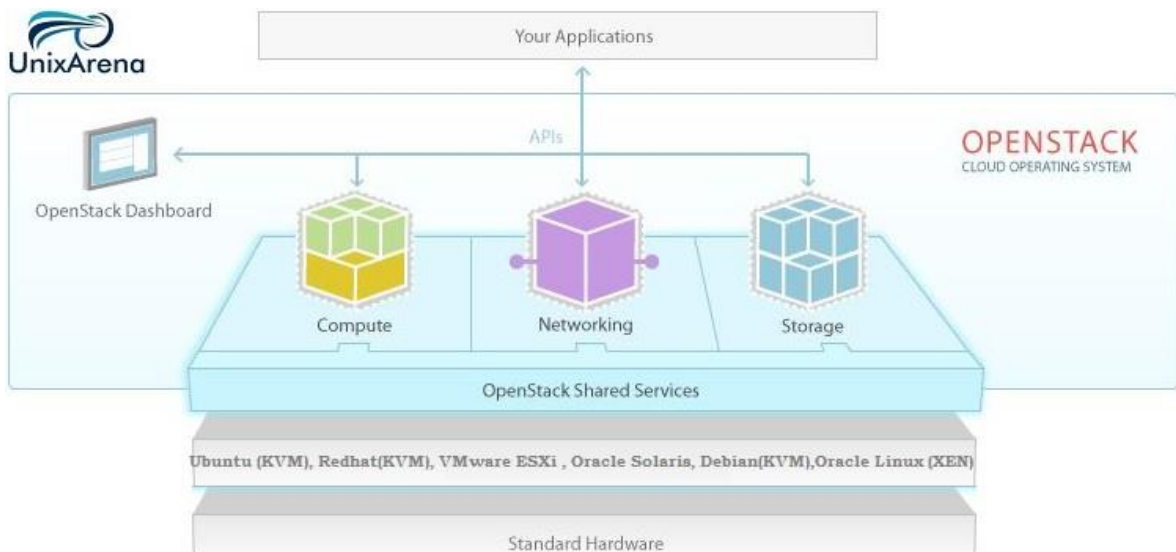


Ilustración 9 Componentes de OpenStack recuperado de <https://www.unixarena.com/2015/08/openstack-architecture-and-components-overview.html/>

A continuación, se muestra la Arquitectura de Rancher:

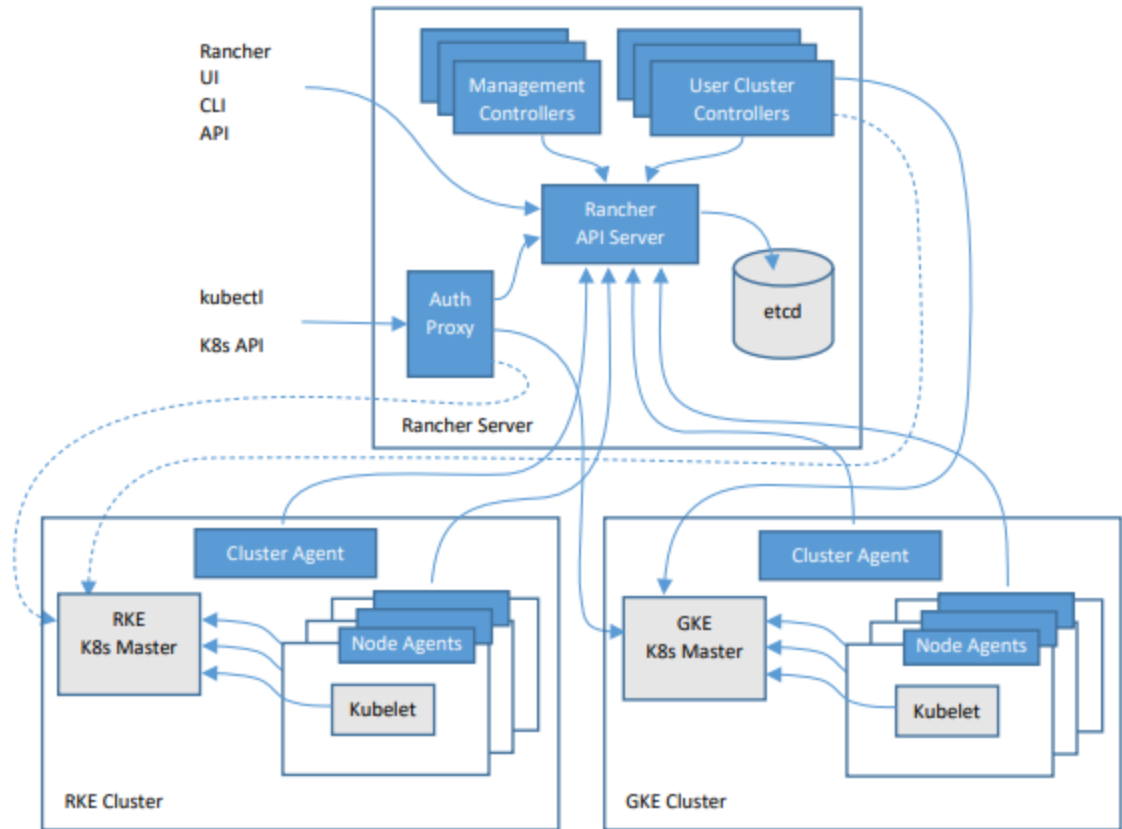


Ilustración 10 Arquitectura Rancher

5.4. Construcción del prototipo.

Para la construcción del prototipo se hará uso de cuatro máquinas virtuales.

1. **Controller Node:** ejecuta los principales servicios que ofrece openstack, panel grafico de administración y acceso a las APIS.
2. **Compute Node:** Provee los recursos de cómputo en la nube (Procesamiento, RAM).
3. **Block Storage Node:** Se encarga de los servicios de almacenamiento en bloques y creación de volúmenes para las máquinas virtuales o instancias.

4. **Network Node:** Administra las redes de la nube (Red de proveedores y de autoservicio).

Los recursos de hardware para las máquinas virtuales son los siguientes:

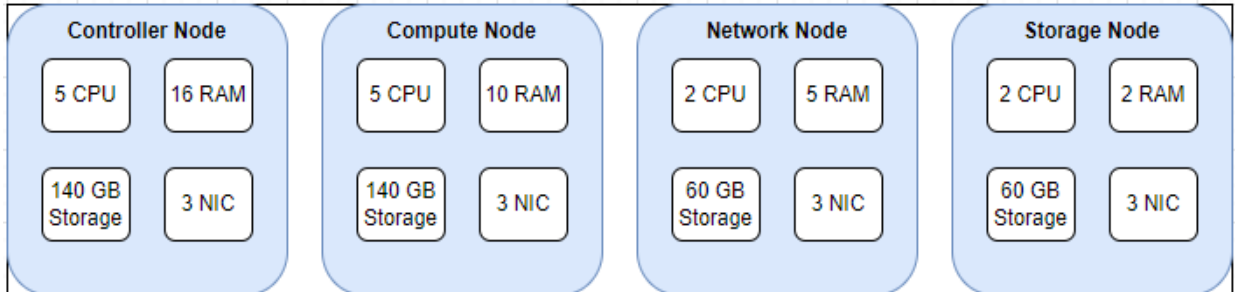


Ilustración 11 Recursos de hardware a utilizar para la construcción del prototipo.

Los servicios instalados necesarios en cada una de las máquinas virtuales para el correcto funcionamiento del prototipo son los siguientes:

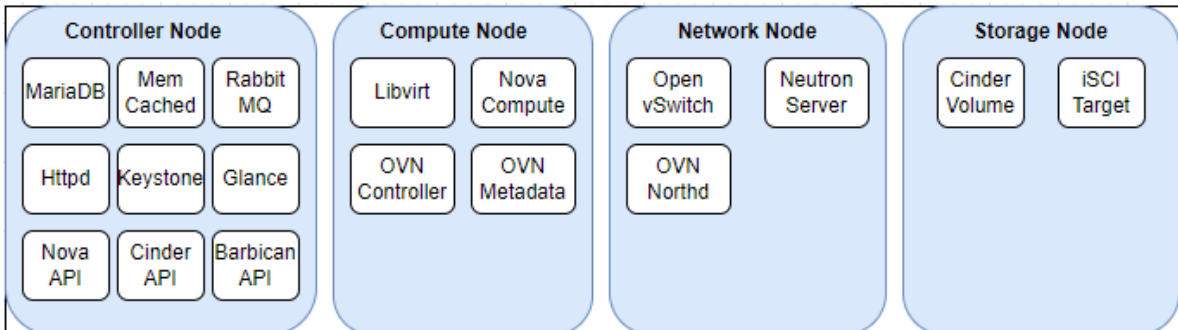


Ilustración 12 Servicios necesarios por máquina virtual

A continuación, se definirán las redes utilizadas para el funcionamiento del prototipo:

1. **Red proveedora para internet:** Es una red tipo NAT la cual se encarga de brindar acceso a internet a los diferentes nodos de la nube. El rango de direcciones IP para esta red es 192.168.122.0 /24

2. **Red de comunicación entre máquinas virtuales:** Red para la comunicación entre los nodos de OpenStack. El rango de direcciones IP para esta red es 10.0.0.0 /24
3. **Red internal OVN:** Red para conectar todas las maquinas a través de Openvswitch. El rango de direcciones IP para esta red es 192.168.100.0 /24

Para la instalación de Openstack se siguió la guía oficial (OpenStack Installation Guide — Installation Guide documentation. (s. f.). <https://docs.openstack.org/install-guide/>).

Para la instalación de Rancher (Ver Anexo L)

5.5. Pruebas de funcionamiento.

Las pruebas de funcionamiento del prototipo de nube se realizarán tomando como base algunos de los resultados obtenidos en las encuestas y entrevistas realizadas, se logró constatar que la distribución más utilizada para el levantamiento de servicios y aplicaciones es Ubuntu server, por lo que se hará el levantamiento de una instancia que contenga dicha distribución para verificar el funcionamiento del prototipo.

Partiendo de lo anterior, se realizarán las siguientes pruebas:

Lanzamiento de instancias

Ahora se procede a lanzar una instancia de Ubuntu Server 20.04 con las siguientes especificaciones de hardware:

VCPU	RAM	Almacenamiento	Red
1	2048 MB	10GB	Private

```

root@controller tremng(keystone)# openstack server create --flavor m1.small --image="Ubuntu2004" --network private --key-name mykey2 --security-group rke1 prueba
/usr/lib/python3/dist-packages/secretsstorage/dhcrypto.py:15: CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes instead
  from cryptography.utils import int_from_bytes
/usr/lib/python3/dist-packages/secretsstorage/util.py:19: CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes instead
  from cryptography.utils import int_from_bytes
-----
| Field | Value |
-----
| OS-DCF:diskConfig | MANUAL |
| OS-EXT-AZ:availability_zone | None |
| OS-EXT-SRV-ATTR:host | None |
| OS-EXT-SRV-ATTR:hypervisor_hostname | None |
| OS-EXT-SRV-ATTR:instance_name | None |
| OS-EXT-SYS:power_state | NOSTATE |
| OS-EXT-SYS:task_state | scheduling |
| OS-EXT-SYS:vm_state | building |
| OS-SRV-USG:launched_at | None |
| OS-SRV-USG:terminated_at | None |
| accessIPv4 | |
| accessIPv6 | |
| addresses | |
| adminPass | ecuzU32tbyHt |
| config_drive | |
| created | 2022-10-31T04:06:33Z |
| flavor | m1.small (0) |
| hostId | |
| id | d48eb685-0129-4f80-89ae-f0a173d2289d |
| image | Ubuntu2004 (ac7f1de6-057d-4e2f-9aa7-5c9c6edf8994) |
| key_name | mykey2 |
| name | prueba |
| progress | 0 |
| project_id | 3fbad2d2690744bd9a7ef44fd3808e78 |
| properties | |
| security_groups | name='0980d10a-8dcf-4aa8-97b2-3881a78c5658' |
| status | BUILD |
| updated | 2022-10-31T04:06:33Z |
| user_id | b410e3ee2cbd4c09825cd5a25d09dced |
| volumes_attached | |
-----

```

Ilustración 13 Creación máquina virtual

Una vez enviado este comando en el nodo controller, podremos visualizar la instancia creada dentro del Dashboard Horizon. En donde podemos asignarle una IP flotante para brindarle acceso a dicho servidor.

Displaying 4 items

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
<input type="checkbox"/>	prueba	Ubuntu2004	192.168.100.115	m1.small	mykey2	Active	nova	None	Running	0 minutes	Create Snapshot
<input type="checkbox"/>	rke1-launcher2	Ubuntu2004	192.168.100.25, 192.168.122.77	m1.medium	mykey2	Active	nova	None	Running	1 day, 5 hours	Create Snapshot
<input type="checkbox"/>	rke1-2	Ubuntu2004	192.168.100.58	m1.medium	mykey2	Active	nova	None	Running	1 day, 5 hours	Create Snapshot
<input type="checkbox"/>	rke1-1	Ubuntu2004	192.168.100.79	m1.medium	mykey2	Active	nova	None	Running	1 day, 5 hours	Create Snapshot

Displaying 4 items

Ilustración 14 Instancia creada y visualizada desde Horizon

Manage Floating IP Associations ✕

IP Address *

▼
+

Select the IP address you wish to associate with the selected instance or port.

Port to be associated *

Cancel
Associate

Ilustración 15 Creación de IP flotante para instancia creada

En la siguiente imagen se puede apreciar la IP flotante que fue asignada es la **192.168.122.128**

Name	prueba
ID	d48eb685-0129-4f80-89ae-f0a173d2289d
Description	prueba
Project ID	3fbad2d2690744bd9a7ef44fd3868e78
Status	Active
Locked	False
Availability Zone	nova
Created	Oct. 31, 2022, 4:06 a.m.
Age	3 minutes
Host	compute
Instance Name	instance-00000037
Reservation ID	r-0i80jl6y
Launch Index	-
Hostname	prueba
Kernel ID	-
Ramdisk ID	-
Device Name	/dev/vda
User Data	-

Specs

Flavor	Not available
---------------	---------------

IP Addresses

private	192.168.100.115, 192.168.122.128
----------------	----------------------------------

Ilustración 16 Ip flotante asignada a instancia

Procedemos a verificar las reglas del grupo de seguridad asignado a la instancia en cuestión

Security Groups

```
rke1 ALLOW IPv4 30000-32767/tcp from 0.0.0.0/0
ALLOW IPv4 6443/tcp from 0.0.0.0/0
ALLOW IPv4 8472/udp from 0.0.0.0/0
ALLOW IPv4 8472/tcp from 0.0.0.0/0
ALLOW IPv4 22/tcp from 0.0.0.0/0
ALLOW IPv4 to 0.0.0.0/0
ALLOW IPv4 icmp from 0.0.0.0/0
ALLOW IPv4 7080/tcp from 0.0.0.0/0
ALLOW IPv4 53/tcp from 0.0.0.0/0
ALLOW IPv4 9100/tcp from 0.0.0.0/0
ALLOW IPv4 179/tcp from 0.0.0.0/0
ALLOW IPv4 80/tcp from 0.0.0.0/0
ALLOW IPv4 2380/tcp from 0.0.0.0/0
ALLOW IPv4 2379/tcp from 0.0.0.0/0
ALLOW IPv4 8080/tcp from 0.0.0.0/0
ALLOW IPv4 10250/tcp from 0.0.0.0/0
ALLOW IPv6 to ::/0
```

Ilustración 17 Grupo de seguridad asignado a la instancia

Constatamos que este grupo de seguridad posee la regla de SSH para poder acceder a la instancia creada desde el exterior de la nube privada.

```
klmba@klmba-H410M-H-V3:~$ sudo ssh -i mykey2.pem ubuntu@192.168.122.128
[sudo] contraseña para klmba:
The authenticity of host '192.168.122.128 (192.168.122.128)' can't be established.
ECDSA key fingerprint is SHA256:W0cYczWAgc4VsVRa4oqFGQt0UiarBZBjInbQLQF9hw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.122.128' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-131-generic x86_64)

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

System information as of Mon Oct 31 04:12:17 UTC 2022

System load:  0.0          Processes:    107
Usage of /:   14.1% of 9.51GB   Users logged in:  0
Memory usage: 10%          IPv4 address for ens3: 192.168.100.115
Swap usage:  0%

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

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

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

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

ubuntu@prueba:~$
```

Ilustración 18 Acceso SSH a la instancia

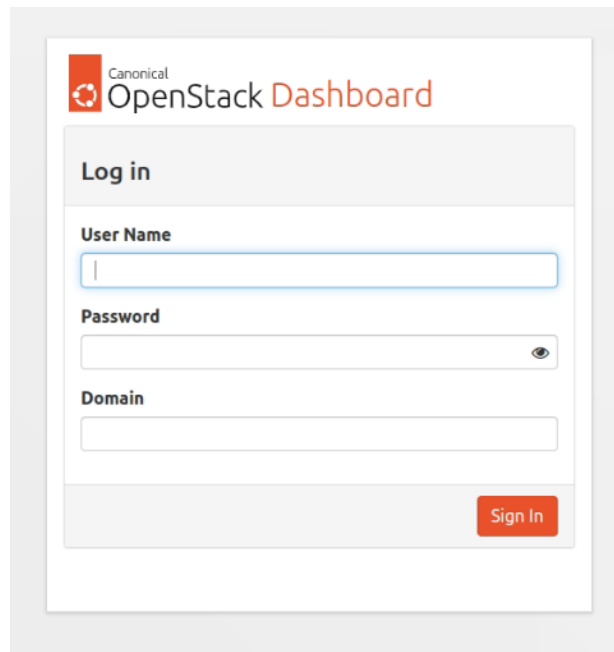
Una vez dentro de la maquina hacemos ping a Google.com para verificar el acceso a internet de la instancia.

```
ubuntu@prueba:~$ ping google.com
PING google.com (142.250.217.174) 56(84) bytes of data:
64 bytes from mia07s60-in-f14.1e100.net (142.250.217.174): icmp_seq=1 ttl=112 time=49.9 ms
64 bytes from mia07s60-in-f14.1e100.net (142.250.217.174): icmp_seq=2 ttl=112 time=45.7 ms
64 bytes from mia07s60-in-f14.1e100.net (142.250.217.174): icmp_seq=3 ttl=112 time=42.6 ms
64 bytes from mia07s60-in-f14.1e100.net (142.250.217.174): icmp_seq=4 ttl=112 time=42.4 ms
64 bytes from mia07s60-in-f14.1e100.net (142.250.217.174): icmp_seq=5 ttl=112 time=55.5 ms
64 bytes from mia07s60-in-f14.1e100.net (142.250.217.174): icmp_seq=6 ttl=112 time=43.5 ms
64 bytes from mia07s60-in-f14.1e100.net (142.250.217.174): icmp_seq=7 ttl=112 time=54.2 ms
^C
--- google.com ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6010ms
rtt min/avg/max/mdev = 42.377/47.657/55.464/5.111 ms
ubuntu@prueba:~$
```

Ilustración 19 Ping a Google.com

5.5.1. Verificación de la instalación de openstack

Login de Dashboard Horizon para openstack



The image shows a screenshot of the OpenStack Dashboard login interface. At the top left, there is the Canonical logo and the text "OpenStack Dashboard". Below this, the heading "Log in" is displayed. The form contains three input fields: "User Name", "Password", and "Domain". The "Password" field includes a toggle icon for visibility. A red "Sign In" button is located at the bottom right of the form area.

Ilustración 20 Login Dashboard Horizon

Endpoints de los servicios

Displaying 11 items

Name	Service	Region	Endpoints
keystone	identity	RegionOne	Admin http://10.0.0.30:5000/v3/ Internal http://10.0.0.30:5000/v3/ Public http://10.0.0.30:5000/v3/
barbican	key-manager	RegionOne	Admin http://10.0.0.30:9311 Internal http://10.0.0.30:9311 Public http://10.0.0.30:9311
cinderv3	volumev3	RegionOne	Admin http://10.0.0.30:8776/v3/3fbad2d2690744bd9a7ef44fd3868e78 Internal http://10.0.0.30:8776/v3/3fbad2d2690744bd9a7ef44fd3868e78 Public http://10.0.0.30:8776/v3/3fbad2d2690744bd9a7ef44fd3868e78
heat-cfn	cloudformation	RegionOne	Admin http://192.168.122.22:8000/v1 Internal http://192.168.122.22:8000/v1 Public http://192.168.122.22:8000/v1
octavia	load-balancer	RegionOne	Admin http://192.168.122.22:9876 Internal http://192.168.122.22:9876 Public http://192.168.122.22:9876
glance	image	RegionOne	Admin http://10.0.0.30:9292 Internal http://10.0.0.30:9292 Public http://10.0.0.30:9292
placement	placement	RegionOne	Admin http://10.0.0.30:8778 Internal http://10.0.0.30:8778 Public http://10.0.0.30:8778
neutron	network	RegionOne	Admin http://192.168.122.22:9696 Internal http://192.168.122.22:9696 Public http://192.168.122.22:9696
nova	compute	RegionOne	Admin http://192.168.122.20:8774/v2.1/3fbad2d2690744bd9a7ef44fd3868e78 Internal http://192.168.122.20:8774/v2.1/3fbad2d2690744bd9a7ef44fd3868e78 Public http://192.168.122.20:8774/v2.1/3fbad2d2690744bd9a7ef44fd3868e78
heat	orchestration	RegionOne	Admin http://192.168.122.22:8004/v1/3fbad2d2690744bd9a7ef44fd3868e78 Internal http://192.168.122.22:8004/v1/3fbad2d2690744bd9a7ef44fd3868e78 Public http://192.168.122.22:8004/v1/3fbad2d2690744bd9a7ef44fd3868e78

Displaying 11 items

Ilustración 21 Endpoints de servicios

Verificación de los contenedores creados por RKE1 (Rancher).

```
ubuntu@rke1-launcher2:~$ k get pods -A -o wide
```

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
default	hostname-server	1/1	Running	0	60m	10.42.1.14	192.168.100.58	<none>	<none>
Ingress-nginx	ingress-nginx-admission-create-97k7v	0/1	Completed	0	29h	10.42.1.4	192.168.100.58	<none>	<none>
Ingress-nginx	ingress-nginx-admission-patch-nlwq	0/1	Completed	2	29h	10.42.0.4	192.168.100.79	<none>	<none>
Ingress-nginx	nginx-ingress-controller-5gjh	1/1	Running	1 (9h ago)	29h	10.42.1.12	192.168.100.58	<none>	<none>
Ingress-nginx	nginx-ingress-controller-xpxcl	1/1	Running	1 (9h ago)	29h	10.42.0.15	192.168.100.79	<none>	<none>
kube-system	calico-kube-controllers-74df54cbb7-r9n9c	1/1	Running	1 (9h ago)	29h	10.42.1.11	192.168.100.58	<none>	<none>
kube-system	canal-qcn6z	2/2	Running	2 (9h ago)	29h	192.168.100.79	192.168.100.79	<none>	<none>
kube-system	canal-qfp5b	2/2	Running	2 (9h ago)	29h	192.168.100.58	192.168.100.58	<none>	<none>
kube-system	coredns-59699769fb-7tclz	1/1	Running	1 (9h ago)	29h	10.42.0.12	192.168.100.79	<none>	<none>
kube-system	coredns-59699769fb-14hxg	1/1	Running	1 (9h ago)	29h	10.42.1.9	192.168.100.58	<none>	<none>
kube-system	coredns-autoscaler-67cbd4599c-bxv9z	1/1	Running	1 (9h ago)	29h	10.42.0.14	192.168.100.79	<none>	<none>
kube-system	metrics-server-585b7cc746-zjcmv	1/1	Running	1 (9h ago)	29h	10.42.0.11	192.168.100.79	<none>	<none>
kube-system	openstack-cloud-controller-manager-5hkmb	1/1	Running	0	46m	192.168.100.58	192.168.100.58	<none>	<none>
kube-system	openstack-cloud-controller-manager-zcm7g	1/1	Running	0	46m	192.168.100.79	192.168.100.79	<none>	<none>
kube-system	rke-coredns-addon-deploy-job-g9q88	0/1	Completed	0	29h	192.168.100.79	192.168.100.79	<none>	<none>
kube-system	rke-ingress-controller-deploy-job-glqrx	0/1	Completed	0	29h	192.168.100.79	192.168.100.79	<none>	<none>
kube-system	rke-metrics-addon-deploy-job-4d5jb	0/1	Completed	0	29h	192.168.100.79	192.168.100.79	<none>	<none>
kube-system	rke-network-plugin-deploy-job-pggdk	0/1	Completed	0	29h	192.168.100.79	192.168.100.79	<none>	<none>

Ilustración 22 Verificación de contenedores

Acceso a la herramienta instalada en contenedor de cluster kubernetes

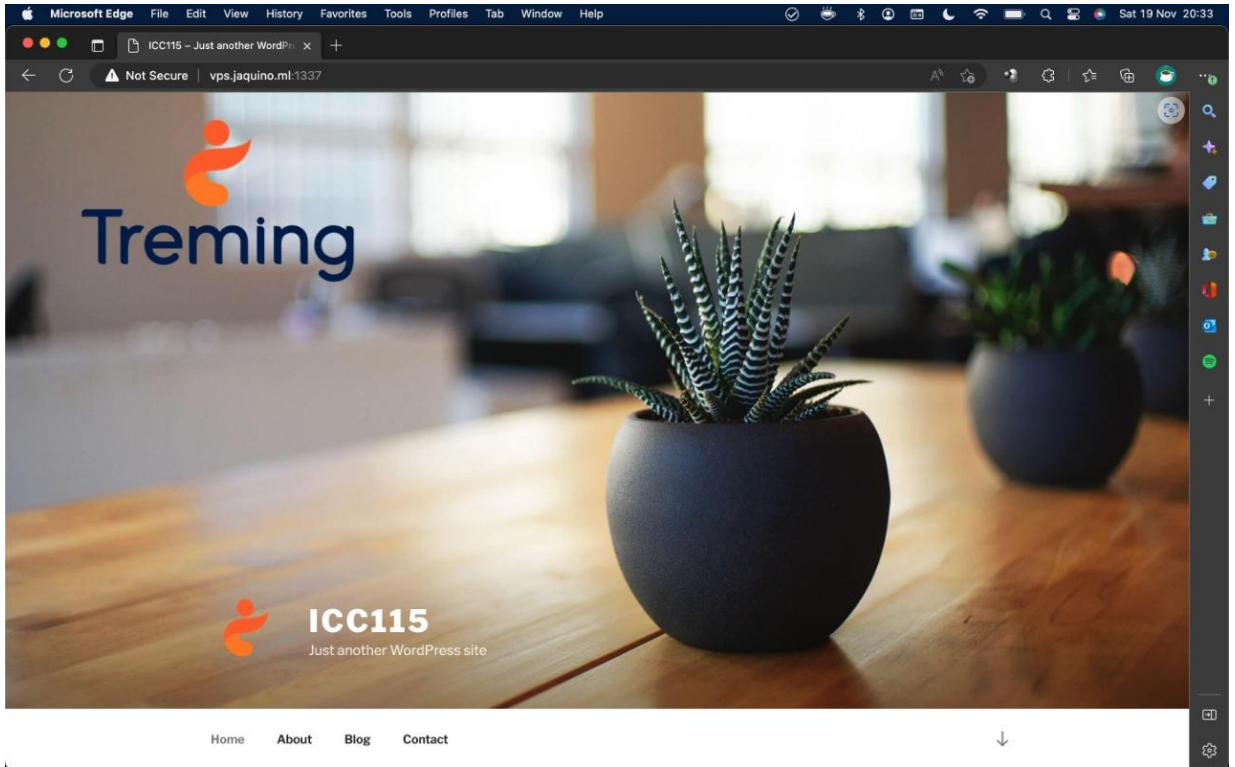


Ilustración 23 Herramienta instalada en contenedor de cluster kubernetes

6. FACTIBILIDADES

Al realizar dicho análisis se ha tenido en cuenta las opciones de nubes públicas, comparando sus precios junto a armar una nube privada. Para las nubes públicas de AWS, Azure y GCP se hicieron uso de sus respectivas herramientas para calcular el costo. Asimismo, se estableció que tanto las nubes públicas tienen que cumplir los siguientes requisitos:

- Networking
- 5 load Balancers
- 2 Clusters de Kubernetes (Dev y UAT)
 - 3 nodos en cada clúster de 8GB RAM, 4 vCPU y 50 GB de Almacenamiento.

En el caso de los primeros dos, al desplegar la nube privada con openstack se puede configurar la red. Los load balancers se pueden configurar como IPs flotantes así que no se incurre en gastos extra para la nube privada. Dichos cálculos se pueden apreciar en el *Anexo K: Cálculo de precios para nubes públicas y privada*.

Una vez realizados los cálculos para las distintas soluciones planteadas, se puede también realizar un cálculo de los gastos acumulados si se llegara a utilizar cualquiera de las soluciones en un plazo de 5 años:

Proveedor	Primer Año	Segundo Año	Tercer Año	Cuarto Año	Quinto Año
Amazon Web Services	\$ 11,557.56	\$ 23,115.12	\$ 34,672.68	\$ 46,230.24	\$ 57,787.80
Microsoft Azure	\$ 12,940.08	\$ 25,880.16	\$ 38,820.24	\$ 51,760.32	\$ 64,700.40
Google Cloud Platform	\$ 12,451.08	\$ 24,902.16	\$ 37,353.24	\$ 49,804.32	\$ 62,255.40
Nube privada	\$ 22,820.73	\$ 23,120.73	\$ 23,420.73	\$ 23,720.73	\$ 24,020.73

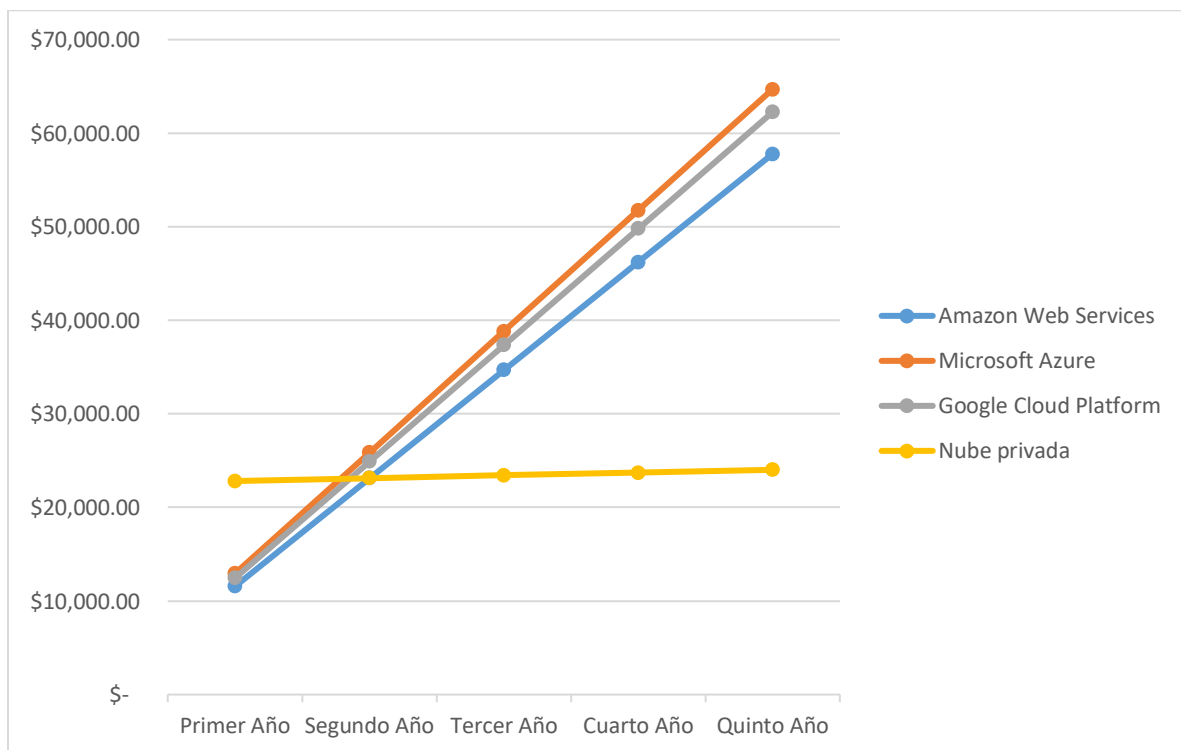


Ilustración 24 Gráfica de gastos acumulados de la solución por proveedores de servicios en la nube

Con los cálculos realizados y al observar la gráfica, se puede apreciar que mientras el costo inicial de la nube privada sobrepasa los de las otras soluciones en nube pública, su costo no sube de la misma manera que los proveedores de nubes públicas. Esto debido a que a largo plazo las soluciones de nube pública tienen una tarifa establecida y el gasto acumulado será mucho mayor y sin tener en consideración costes por sobrepasar la cuota establecida en el ancho de banda, o si se pretende escalar la solución.

Es por esto por lo que a largo plazo la solución que se recomienda para la empresa Treming, S.A. de C.V. es la de nube privada. Con esta solución no solo se podrá observar un costo acumulado al pasar de los años, sino que también se puede escalar a medida que se necesite.

Si bien no se han incluido todos los costes y gastos que se incurren en cualquier solución, se ha incluido los costes más importantes y los que significaran a largo plazo un punto de inflexión en la decisión.

7. CONCLUSIONES

- La computación en la nube es cada vez más utilizada para las soluciones tecnológicas de las empresas, esto debido a los diferentes beneficios que ofrece, como escalabilidad, seguridad, opciones de control, etc. A través de la investigación realizada se puede concluir que implementar una nube privada es una opción factible y con beneficios, para, prestar servicios a los clientes y servicios internos.
- Una nube privada conlleva una inversión inicial mayor que la contratación de una nube pública debido a que se debe tener la infraestructura inicial para montar la nube y capacitar al personal correspondiente para la puesta en marcha, pero a medio plazo la inversión esta se ha recuperado, brindando beneficios de control y seguridad una vez estos sean configurados adecuadamente.
- Los servicios de nube privada planteadas en el prototipo para grupo Treming ayudaran a agilizar los desarrollos de sistemas a sus clientes a través de entornos de prueba y desarrollo generados rápidamente con los contenedores de kubernetes, y les otorgara una ventaja competitiva con sus competidores al brindarles a sus clientes servicio como infraestructura para las soluciones de software que estos posean.

8. REFERENCIAS BIBLIOGRÁFICAS.

Education, I. C. (2021, mayo 5). *IaaS (Infrastructure-as-a-Service)*. IBM Cloud Learn Hub.

<https://www.ibm.com/cloud/learn/iaas>

Gracia, L. (2020, enero 7). *¿Qué es Kubernetes? Un poco de Java*.

<https://unpocodejava.com/2020/01/07/que-es-kubernetes/amp/>

Hall, T. (2020). *Cultura de DevOps*. Atlassian. [https://www.atlassian.com/es/devops/what-is-](https://www.atlassian.com/es/devops/what-is-devops/devops-culture)

[devops/devops-culture](https://www.atlassian.com/es/devops/what-is-devops/devops-culture)

Harris, R. (2021, enero 25). *Aspectos básicos de los contenedores: ¿qué es la tecnología de contenedores, qué es Kubernetes y por qué lo necesita?* Rackspace.

[https://www.rackspace.com/es/blog/containers-](https://www.rackspace.com/es/blog/containers-101#:~:text=%C2%BFQu%C3%A9%20es%20la%20contenerizaci%C3%B3n%3F,y)

[101#:~:text=%C2%BFQu%C3%A9%20es%20la%20contenerizaci%C3%B3n%3F,y](https://www.rackspace.com/es/blog/containers-101#:~:text=%C2%BFQu%C3%A9%20es%20la%20contenerizaci%C3%B3n%3F,y)

[%20de%20la%20infraestructura%20subyacente](https://www.rackspace.com/es/blog/containers-101#:~:text=%C2%BFQu%C3%A9%20es%20la%20contenerizaci%C3%B3n%3F,y)

Install, L. P. (2019, enero 14). *Virtualización: Tecnologías disponibles para el 2019*. Desde Linux.

<https://blog.desdelinux.net/virtualizacion-tecnologias-disponibles-2019/>

Jamsa, K. (2022). *Cloud Computing* (2nd ed.) [E-book]. Jones & Bartlett Learning.

Lisdorf, A. (2021). *Cloud Computing Basics: A Non-Technical Introduction* (1st ed.) [E-book].

Apress.

Manvi, S., & Shyam, G. K. (2021). *Cloud Computing* (1st ed.). CRC Press.

Marinescu, D. C. (2022). *Cloud Computing: Theory and Practice* (3rd ed.) [E-book]. Morgan Kaufmann.

Mell, P., Grance, T., & Information Technology Laboratory (National Institute of Standards and Technology). Computer Security Division. (2011). *The NIST definition of cloud computing*. Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology.

Microsoft. (2022a). *¿Qué es DevOps? Explicación de DevOps / Microsoft Azure*. Microsoft Azure. <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-devops/#culture>

Microsoft. (2022). *Qué es virtualización - Definición*. Microsoft Azure. <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-virtualization/>

Redhat. (2018, marzo 19). *Virtualización*. <https://www.redhat.com/es/topics/virtualization#:~:text=La%20m%C3%A1quina%20f%C3%ADsica%20original%20en,medios%20que%20pueden%20redistribuirse%20f%C3%A1cilmente>

Redhat. (2018b, abril 19). *El concepto de DevOps*. <https://www.redhat.com/es/topics/devops>

Redhat. (2020, marzo 27). *¿Qué es Kubernetes?* <https://www.redhat.com/es/topics/containers/what-is-kubernetes>

Sattler, T. (2022, abril 25). *3 Types of Container Runtime and the Kubernetes Connection*. Aqua.

<https://www.aquasec.com/cloud-native-academy/container-security/container-runtime/>

Verdino, X. (2017, octubre 10). *Instanciamos DevOps*. Medium.

<https://medium.com/pyxistechblog/instanciamos-devops-e75bf0bad3b4>

Veritas. (2022). *¿Qué es la contenerización? ¿Cuáles son los beneficios?*

<https://www.veritas.com/es/mx/information-center/containerization>

Red Hat. (27 de 06 de 2021). *Obtenido de Tecnología de virtualización (KVM):*

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

[isKVM#:~:text=KVM%20convierte%20a%20Linux%20en,1%20\(sin%20sistema%20op](https://www.redhat.com/es/topics/virtualization/what-is-kvm#:~:text=KVM%20convierte%20a%20Linux%20en,1%20(sin%20sistema%20op)

[erativo\).&text=Cada%20m%C3%A1quina%20virtual%20se%20implementa,%20C%20C](https://www.redhat.com/es/topics/virtualization/what-is-kvm#:~:text=KVM%20convierte%20a%20Linux%20en,1%20(sin%20sistema%20op)

[P U%20%20memoria%20y%20discos](https://www.redhat.com/es/topics/virtualization/what-is-kvm#:~:text=KVM%20convierte%20a%20Linux%20en,1%20(sin%20sistema%20op)

Zun. (22 de 09 de 2017), *Obtenido de Bugs launchpad Zun:*

<https://bugs.launchpad.net/zun/+bug/1718865>

Madhuri Kumari, Shubham Kumar Sharma (2016). *Who Carries your containers? Zun or Magnum?*.

<https://dokumen.tips/technology/who-carries-your-container-zun-or->

[magnum.html?page=1](https://dokumen.tips/technology/who-carries-your-container-zun-or-magnum.html?page=1)

Server World (4 de 04 de 2022). *Ubuntu 22.04: Openstack Yoga*

<https://www.server->

[world.info/en/note?os=Ubuntu_20.04&p=openstack_yoga&f=1](https://www.server-world.info/en/note?os=Ubuntu_20.04&p=openstack_yoga&f=1)

Ubuntu (2022). *Charmed Kubernetes on OpenStack*

<https://ubuntu.com/kubernetes/docs/openstack-integration>

Openmetal (2022) *Installing a Rancher managed cluster on OpenStack*

<https://openmetal.io/docs/manuals/kubernetes-guides/installing-a-rancher-cluster-on-openstack>

Kavis, M. J. (2014). *Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS)*. Wiley.

9. ANEXOS.

Anexo A Entrevista a CEO grupo TREMING S.A. de C.V.



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.

DISEÑO DE PROTOTIPO DE UNA INFRAESTRUCTURA EN LA NUBE PARA MEJORAR EL
MONITOREO DE SERVICIOS Y ENTREGAS DE FUNCIONALIDADES A LOS CLIENTES DE GRUPO
TREMING S.A. DE C.V.

Guía de Entrevista N° 1. Dirigida al CEO de la institución.

1. **Objetivo.**

Obtener información importante de los tomadores de decisiones de la institución para comprender las consecuencias, sean estas positivas o negativas, que la implementación de una nube privada dentro de la institución podría causar.

2. **Preguntas.**

- 1- ¿Conoce los beneficios de tener una nube privada respecto a tener una nube pública?
- 2- ¿Considera que para su empresa brindar un servicio de nube privada obtendría un valor agregado para sus servicios comparándolos con la competencia?
- 3- ¿Actualmente, existen planes para invertir en tecnología?
- 4- Una nube privada de tipo IaaS, le permitiría publicar sus propias aplicaciones y ofrecer estos servicios a sus clientes y una nube privada de tipo PaaS, le permitiría a su equipo de implementaciones y desarrollo facilitar el despliegue de nuevos módulos para su testeo. ¿Cómo cree que la falta de esta está afectando a su negocio?
- 5- ¿Planea capacitar al personal de la empresa o contratar nuevo personal para la implementación del proyecto?
- 6- ¿Qué servicios quiere brindar a los clientes?

Anexo B Entrevista al encargado de levantamientos de servicios del departamento de IT



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.

DISEÑO DE PROTOTIPO DE UNA INFRAESTRUCTURA EN LA NUBE PARA MEJORAR EL
MONITOREO DE SERVICIOS Y ENTREGAS DE FUNCIONALIDADES A LOS CLIENTES DE GRUPO
TREMING S.A. DE C.V.

Guía de Entrevista Nº 2. Dirigida al encargado de levantamiento de servicios del departamento de TI.

1. Objetivo.

Obtener información para comprender las consecuencias, sean estas positivas o negativas, que una nube privada afectara en el proceso de levantamiento de servicios a clientes.

2. Preguntas.

1. ¿Cuáles son las funciones que realiza en el departamento de implementación de servicios y cuánto tiempo lleva desempeñando dichas funciones?

2. ¿Qué tipo de servicio en la nube es el más demandado actualmente dentro de su empresa?

3. ¿Cuál cree usted que sería el mayor beneficio para su empresa contar con su propia nube privada?

4. ¿Cree usted que la empresa cuenta con el personal capacitado para implementar una solución de servicio tipo IaaS y PaaS, así como también su mantenimiento a lo largo del tiempo?

5. ¿Considera que se facilitara el levantamiento de servicios a clientes con una nube privada?

Anexo C Cuestionario a departamento de implementación de Sistemas



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.

**DISEÑO DE PROTOTIPO DE UNA INFRAESTRUCTURA EN LA NUBE PARA MEJORAR EL
MONITOREO DE SERVICIOS Y ENTREGAS DE FUNCIONALIDADES A LOS CLIENTES DE GRUPO
TREMING S.A. DE C.V.**

Guía de Cuestionario Nº 1. Dirigida a los empleados del departamento de Implementación de
Sistemas.

1. Objetivo.

Conocer la opinión de los empleados del departamento de Implementación de sistemas respecto a la implementación de una nube privada en la empresa.

2. Preguntas.

1) ¿En qué nivel están sus conocimientos en el área de infraestructura en la nube?

a. Básico. b. Intermedio. c. Avanzado.

2) ¿En caso de que desconozca algunos aspectos de infraestructura en la nube, estaría usted dispuesto a dedicar tiempo extra para mejorar su conocimiento?

a. Sí. b. No. c. Si, si la empresa nos capacita e. Solo si fuera remunerado.

3) ¿Cuál es el tipo de servicio en la nube que considera ayude en la implementación de servicios a clientes?

a. IaaS (Infraestructura como Servicio). b. PaaS (Plataforma como Servicio).

c. SaaS (Software como Servicio). e. Ninguna

4) ¿Qué tan necesario considera usted que la empresa posea su propia nube?

a. No es necesario. b. Poco necesario. c. Necesario. d. Muy necesario.

5) ¿Cuál cree usted que podría ser el mayor inconveniente que podrían tener como departamento de implementación al implementar una solución de servicio IaaS?

a. Personal poco capacitado. b. Infraestructura interna actual insuficiente.

c. Poco interés por parte de los tomadores de decisiones.

6) Para desplegar las aplicaciones de los clientes que manejan, ¿Qué sistema operativo es el que utiliza?

- a. Windows b. Ubuntu Server c. Debian d. Centos e. Fedora

7) ¿Cuánto espacio en disco duro utilizan en promedio las aplicaciones implementadas?

- a. 20 - 30 GB b. 30 - 50 GB c. 50 GB - 100 GB d. 100 GB - 200 GB
e. 200 GB – 1TB f. 1 TB - 5 TB g. 5 TB - 20 TB

8) ¿Cuánto es el consumo de RAM de las aplicaciones con baja demanda de los clientes?

- a. 512 MB - 1GB b. 2 GB - 4 GB c. 4 GB - 6 GB
d. 6 GB - 8 GB e. 8 GB - 10 GB f. Mas de 10 GB

9) ¿Cuánto es el consumo de RAM de las aplicaciones con alta demanda de los clientes?

- a. 4 GB - 8 GB b. 8 GB - 12 GB c. 12 GB – 16 GB
d. 16 GB – 20 GB e. Mas de 20 GB

Anexo D Cuestionario a departamento de desarrollo de sistemas



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.

**DISEÑO DE PROTOTIPO DE UNA INFRAESTRUCTURA EN LA NUBE PARA MEJORAR EL
MONITOREO DE SERVICIOS Y ENTREGAS DE FUNCIONALIDADES A LOS CLIENTES DE GRUPO
TREMING S.A. DE C.V.**

Guía de Cuestionario N° 2. Dirigida a los empleados del departamento desarrollo.

1. Objetivo.

Recopilar información importante de los empleados del departamento de desarrollo respecto a las aplicaciones actuales.

2. Preguntas.

1) ¿Generalmente para qué utiliza usted los servicios contratados en la nube?

a. Configurar entornos de prueba.

b. Despliegue de aplicaciones.

c. Alojamiento de bases de datos.

d. Alojamiento de archivos.

2) ¿Con qué frecuencia experimenta fallas en los servicios contratados?

a. Nunca.

b. Poco frecuente.

c. Frecuentemente.

d. Muy frecuente.

3) ¿Cuál sería el beneficio para usted si la empresa administrara su propia nube privada?

a. Creación de entornos de pruebas parecidos al entorno de producción.

b. Solución más rápida a problemas. c. Ambos.

4) ¿Con cuál tipo de nube mejora el desempeño de su trabajo?

a. Nube privada.

b. Nube pública.

c. Indiferente

5) ¿Lenguajes de programación en los que se desarrolla?

a. PHP b. Python

c. Javascript

d. Java

e. C

f. Dart

g. Otro _____

6) ¿Cuál es el gestor de bases de datos que más utiliza?

a. MySQL

b. MongoDB

c. PostgreSQL

d. Oracle

e. SQLite

f. MariaDB

g. SQLServer

7) ¿Cuál es el rango de tiempo en que consiguen un entorno de pruebas al iniciar un nuevo proyecto de desarrollo?

- a. Mas de 2 semanas b. 1-2 semanas c. 2-6 días d. 1 día

Anexo E Cuestionario a clientes Grupo TREMING S.A. de C.V.



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.

**DISEÑO DE PROTOTIPO DE UNA INFRAESTRUCTURA EN LA NUBE PARA MEJORAR EL
MONITOREO DE SERVICIOS Y ENTREGAS DE FUNCIONALIDADES A LOS CLIENTES DE GRUPO
TREMING S.A. DE C.V.**

Guía de Cuestionario N° 3. Dirigida a los Clientes.

1. Objetivo.

Recopilar información de los clientes sobre los servicios en nube que contratan en la actualidad y su opinión respecto a la posibilidad de que la institución que les desarrolla sus aplicaciones ofrezca estos servicios.

2. Preguntas.

1) ¿Cuántas aplicaciones tiene funcionando actualmente?

- a. 1. b. 2. c. 3. d. 4. e. 5 o más

- 2) ¿Experimenta fallas en sus aplicaciones frecuentemente?
- a. Nunca.
 - b. Casi nunca.
 - c. Frecuentemente.
 - d. Muy frecuente
- 3) ¿Considera usted justo el precio que paga por el servicio recibido?
- a. Sí, estoy conforme con ello.
 - b. Es un buen servicio, pero con costos altos.
 - c. Los costos están bien, pero el servicio es de mala calidad.
 - d. No, no estoy conforme con el servicio.
- 4) ¿Estaría dispuesto a cambiar el servicio donde corren sus aplicaciones?
- a. Sí, mientras funcione bien.
 - b. Si, por otro mejor en razón calidad/precio.
 - c. No, no lo cambiaria.
- 5) ¿Las aplicaciones que utiliza consumen muchos recursos?
- a. Sí, consumen muchos recursos.
 - b. Consumen una cantidad considerable.
 - c. No, consumen pocos recursos.
 - d. No estoy seguro.
- 6) ¿Apoyaría a la empresa Treming S.A. de C.V. en caso ofrecieran un servicio de nube privada?
- a. Si es más barato que mi proveedor actual
 - b. Si no da tantos errores como el proveedor actual

c. Si hay una buena comunicación para el soporte del servicio.

d. No lo apoyaría

7) Cuando contrata un servicio de computación en la nube ¿Necesita conocer los detalles técnicos de este servicio?

a. Si, Es necesario conocer todos los detalles b. No, Solo necesito saber el precio

Anexo F Resultado Entrevista a CEO grupo TREMING S.A. de C.V.

Pregunta Realizada	Respuesta
<p>¿Conoce los beneficios de tener una nube privada respecto a tener una nube publica?</p>	<p>No tengo mucho conocimiento entre la diferencia entre una nube publica respecto a la privada, pero tal como su nombre dice considero que el termino privado es algo alusivo a que yo tendré la capacidad de velar por mis recursos, tanto en sus asignaciones, como en su mantenimiento, a su vez tendría la capacidad de agregar toda la seguridad que sea pertinente a razón de mantener seguros los servicios de mis clientes.</p>
<p>¿Considera que para su empresa brindar un servicio de nube privada obtendría un valor agregado para sus servicios comparándolos con la competencia?</p>	<p>Sí, ya que tengo el conocimiento que la competencia hace uso de servicios pobres de nube, por lo que tener la capacidad de ofrecer una nube con mucha seguridad presentaría una brecha muy amplia entre mis servicios y los de las demás empresas.</p>

<p>¿Actualmente, existen planes para invertir en tecnología?</p>	<p>Si, actualmente nos encontramos haciendo una inversión fuerte en temas de tecnología ya que sufrimos un pequeño ataque en el cual desviaron llamada a través de nuestra planta telefónica. Por lo que nos encontramos en plan de invertir todo aquello que sea posible en temas de tecnología.</p>
<p>Una nube privada de tipo IaaS, le permitiría publicar sus propias aplicaciones y ofrecer estos servicios a sus clientes y una nube privada de tipo PaaS, le permitiría a su equipo de implementaciones y desarrollo facilitar el despliegue de nuevos módulos para su testeó. ¿Como cree que la falta de esta está afectando a su negocio?</p>	<p>Nos afecta fuertemente, ya que actualmente no poseemos ambientes de desarrollo como tal. Ya que nuestras pruebas las hacemos en un entorno compartido con producción de cada cliente. Por lo que una plataforma como servicio para pruebas internas nos caería como anillo al dedo.</p>
<p>¿Planea capacitar al personal de la empresa o contratar nuevo personal para la implementación del proyecto?</p>	<p>Considero que tenemos un personal capaz de implementar la tecnología que se nos proponga, realizamos un proceso de selección exhaustivo en cada plaza por lo que contamos con el mejor personal posible.</p>

<p>¿Qué servicios quiere brindar a los clientes?</p>	<p>Queremos brindar servicios web confiables, seguros y persistentes. Ofrecer tranquilidad al cliente de que todos sus sistemas está perfectamente resguardado y protegido con nosotros.</p>
---	--

Anexo G Resultado Entrevista al encargado de levantamientos de servicios del departamento de IT

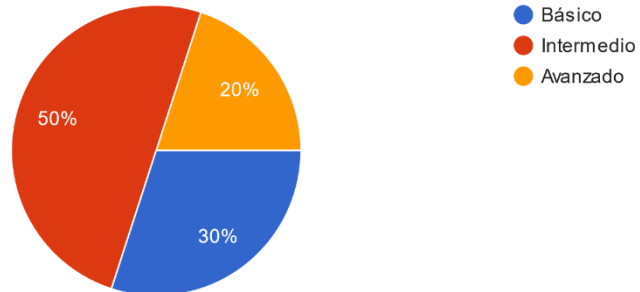
Pregunta Realizada	Respuesta
<p>¿Cuáles son las funciones que realiza en el departamento de implementación de servicios y cuánto tiempo lleva desempeñando dichas funciones?</p>	<p>Yo me encargo de realizar el lanzamiento de instancias, a su vez evaluar que módulos serán necesarias en la implementación para poder cargar todo aquel modulo que sea necesario para el funcionamiento de la vertical solicitada por el cliente. A su vez, me encargo de realizar la instalación del servicio de base de datos y verificar la comunicación entre el servicio web lanzado y su respectiva base de datos. Levo aproximadamente 5 años siendo el encargo de este tema en la empresa.</p>
<p>¿Qué tipo de servicio en la nube es el más demandado actualmente dentro de su empresa?</p>	<p>El servicio más demandado es el ERP llamado Odoo, aunque siempre hay algunas implementaciones asiladas de sitios web hechos a la medida.</p>

<p>¿Cuál cree usted que sería el mayor beneficio para su empresa contar con su propia nube privada?</p>	<p>La administración de la seguridad de los servicios.</p>
<p>¿Cree usted que la empresa cuenta con el personal capacitado para implementar una solución de servicio tipo IaaS y PaaS, así como también su mantenimiento a lo largo del tiempo?</p>	<p>Si, considero que todos somos capaces de llevar a cabo dicha implementación. Porque tenemos la suficiente experiencia en manejo de servicios.</p>
<p>¿Considera que se facilitara el levantamiento de servicios a clientes con una nube privada?</p>	<p>Si, porque podríamos tener sabores de instancias ya creados en imágenes de disco y solo sería necesario lanzar el servidor con dicha imagen las veces que sea necesario</p>

Anexo H Resultados cuestionario a departamento de implementación de Sistemas

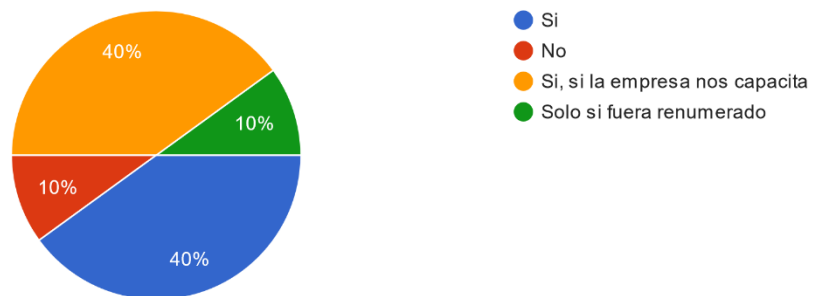
¿En qué nivel están sus conocimientos en el área de infraestructura en la nube?

10 respuestas



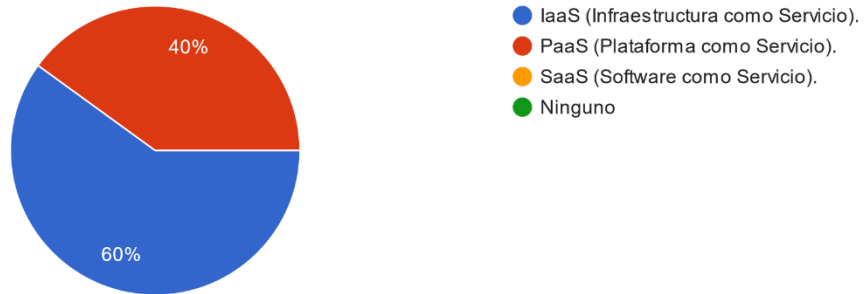
¿En caso de que desconozca algunos aspectos de infraestructura en la nube, estaría usted dispuesto a dedicar tiempo extra para mejorar su conocimiento?

10 respuestas



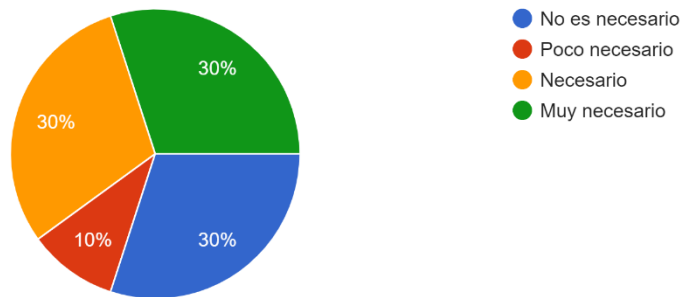
¿Cuál es el tipo de servicio en la nube que considera ayude en la implementación de servicios a clientes?

10 respuestas



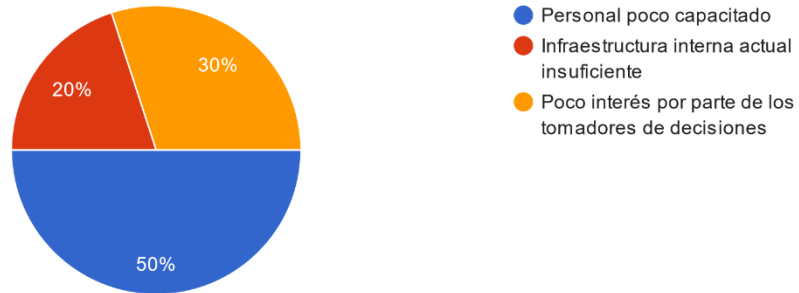
¿Qué tan necesario considera usted que la empresa posea su propia nube?

10 respuestas



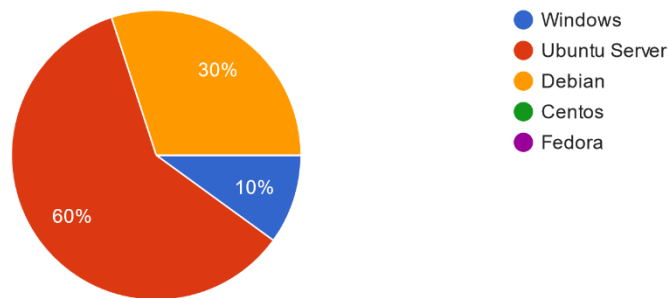
¿Cuál cree usted que podría ser el mayor inconveniente que podrían tener como departamento de implementación al implementar una solución de servicio IaaS?

10 respuestas



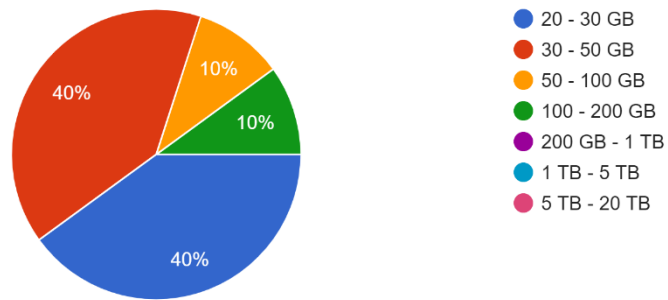
Para desplegar las aplicaciones de los clientes que manejan, ¿Qué sistema operativo es el que utiliza?

10 respuestas



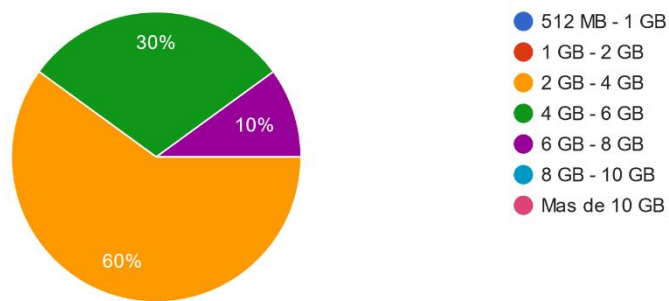
¿Cuánto espacio en disco duro utilizan en promedio las aplicaciones implementadas?

10 respuestas



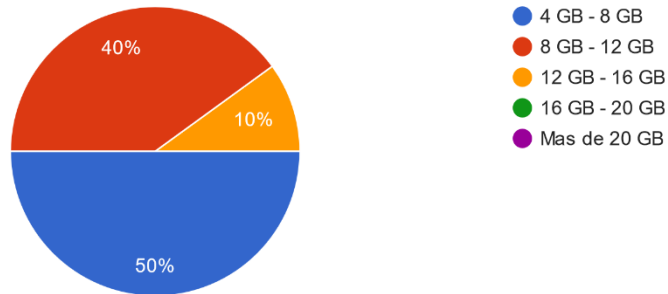
¿Cuánto es el consumo de RAM de las aplicaciones con baja demanda de los clientes?

10 respuestas



¿Cuánto es el consumo de RAM de las aplicaciones con alta demanda de los clientes?

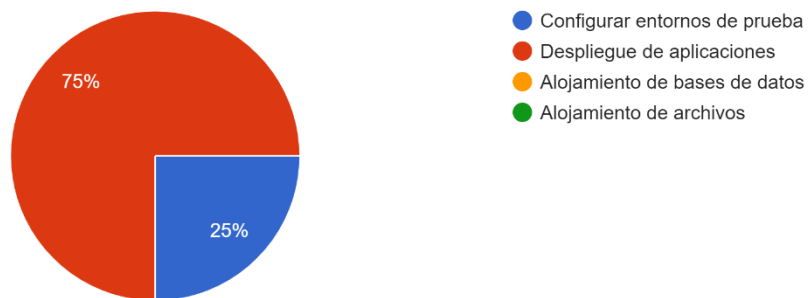
10 respuestas



Anexo I Resultados cuestionario a departamento de desarrollo de sistemas

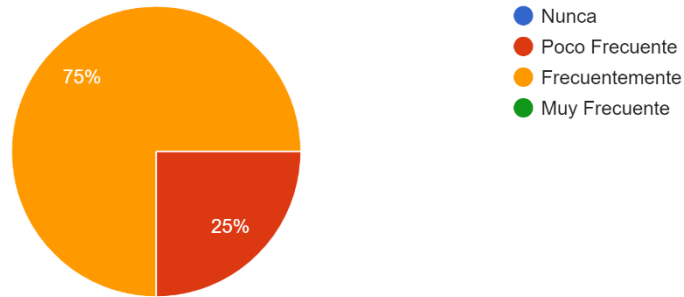
¿Generalmente para qué utiliza usted los servicios contratados en la nube?

4 respuestas



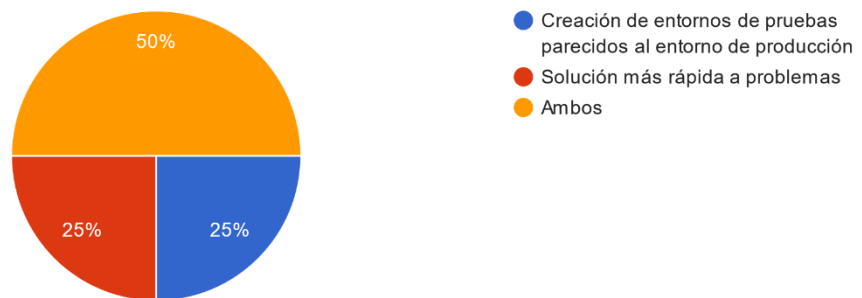
¿Con qué frecuencia experimenta fallas en los servicios contratados?

4 respuestas



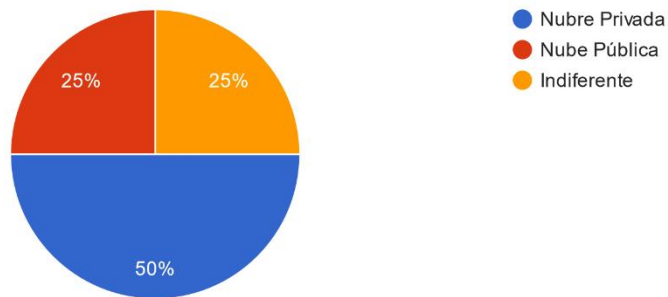
¿Cuál sería el beneficio para usted si la empresa administrara su propia nube privada?

4 respuestas



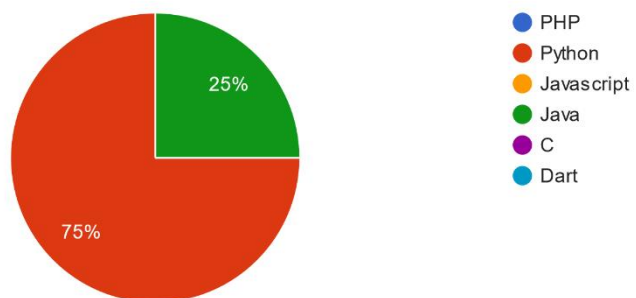
¿Con cuál tipo de nube mejora el desempeño de su trabajo?

4 respuestas



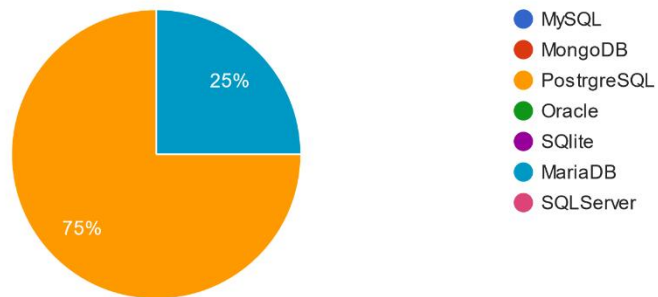
¿Lenguajes de programación en los que se desarrolla?

4 respuestas



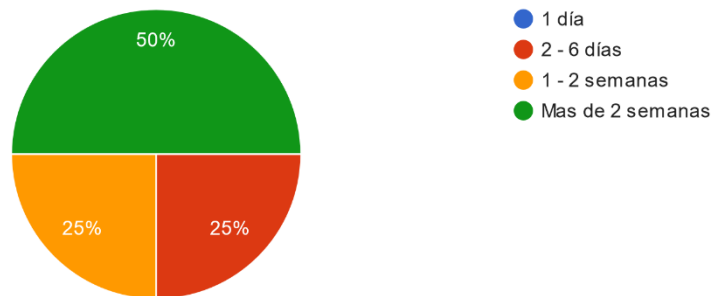
¿Cuál es el gestor de bases de datos que más utiliza?

4 respuestas



¿Cuál es el rango de tiempo en que consiguen un entorno de pruebas al iniciar un nuevo proyecto de desarrollo?

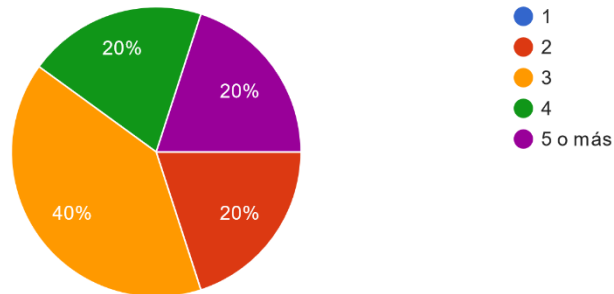
4 respuestas



Anexo J Resultados cuestionario a clientes Grupo TREMING S.A. de C.V.

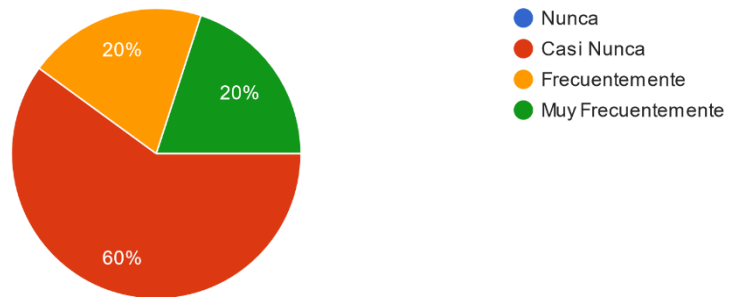
¿Cuántas aplicaciones tiene funcionando actualmente?

5 respuestas



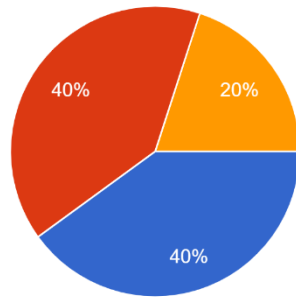
¿Experimenta fallas en sus aplicaciones frecuentemente?

5 respuestas



¿Considera usted justo el precio que paga por el servicio recibido?

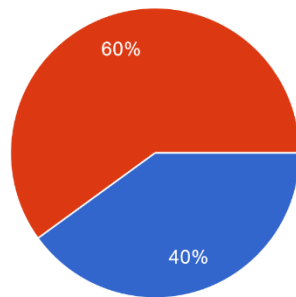
5 respuestas



- Sí, estoy conforme con ello
- Es un buen servicio, pero con costos altos
- Los costos están bien, pero el servicio es de mala calidad.
- No, no estoy conforme con el servicio

¿estaría dispuesto a cambiar el servicio donde corren sus aplicaciones?

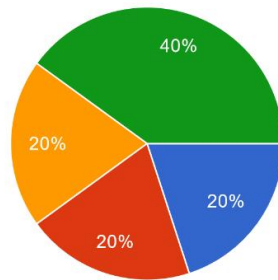
5 respuestas



- Sí, mientras funcione bien
- Sí, por otro mejor en razón calidad/ precio.
- No, no lo cambiaría

¿Las aplicaciones que utiliza consumen muchos recursos?

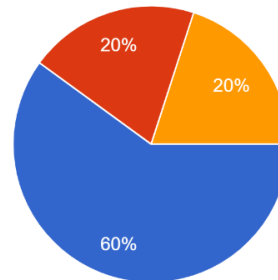
5 respuestas



- Si, consumen muchos recursos.
- Consumen una cantidad considerable.
- No, consumen pocos recursos.
- No estoy seguro.

¿Apoyaría a la empresa Treming S.A. de C.V. en caso ofrecieran un servicio de nube privada?

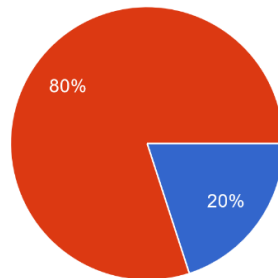
5 respuestas



- Si es más barato que mi proveedor actual
- Si no da tantos errores como el proveedor actual
- Si hay una buena comunicación para el soporte del servicio.
- No lo apoyaría

Quando contrata un servicio de computación en la nube ¿Necesita conocer los detalles técnicos de este servicio?

5 respuestas



- Si, Es necesario conocer todos los detalles
- No, Solo necesito saber el precio

Anexo K Cálculo de precios para nubes públicas y privada

Amazon Web Services (AWS)

Servicio	Region	Costo Mensual (\$)	Costo 12 Meses (\$)	Configuración
Amazon VPC (Networking)	us-east-2 (Ohio)	229.04	2748.48	2 NAT Gateways, 2 Internet gateways, 8 subnets 2 VPC
Amazon EKS (Kuberntes)	us-east-2 (Ohio)	146.00	1752.00	2 Clusters EKS
Amazon EC2 (Nodos para Kubernetes)	us-east-2 (Ohio)	476.76	5721.12	6 Instancias a1.xlarge EC2, 50GB de almacenamiento.
Amazon ALB (Application Load Balancer)	us-east-2 (Ohio)	111.33	1335.96	5 Application Load Balancers
TOTAL		963.13	7057.08	

Microsoft Azure

Servicio	Region	Costo Mensual (\$)	Costo 12 Meses (\$)	Configuración
Azure Kubernetes Service	West US	1051.20	12614.40	2 AKS Clusters. El precio ya incluye 3 Instancias tipo A3 por cada clúster.
Virtual Network	West US y West US 2	14.00	168.00	Virtual Network 1 en West US, Virtual Network 2 en West US. Precio por cada 100 GB utilizados en cada Virtual Network.
Load Balancer	West US	0.00	0.00	5 Basic Load Balancers.
Direcciones IP Estáticas (para los Load Balancer)	West US	13.14	157.68	5 direcciones IP Elásticas.
TOTAL		1078.34	12940.08	

Google Cloud Platform

Servicio	Region	Costo Mensual (\$)	Costo 12 Meses (\$)	Configuración
Google Kubernetes Engine	us-east1 (South Carolina)	862.00	10344.00	2 Cluster GKE. El precio ya incluye 3 instancias tipo n1-standard-4 (con 50 GB de almacenamiento por instancia) por cada cluster.
Cloud Load Balancing	us-east1 (South Carolina)	128.55	1542.60	5 Load Balancers.
Cloud NAT (Networking)	us-east1	47.04	564.48	1000 GB de tráfico a través de este NAT. Networking viene sin costo adicional ya que GCP cobra por el servicio de NAT.
TOTAL		1037.59	12451.08	

Nube Privada

Debido a que las consideraciones a tomar para poder armar y desplegar una nube privada van más allá de solo tener en cuenta el hardware y software a utilizar, sino que también se toma en cuenta la inversión del espacio a utilizar y las normativas que se planean seguir. Para objetivos de este trabajo de investigación se plantea utilizar un Módulo Autocontenido, de esta manera los requisitos de espacio para el gabinete y el equipo físico a utilizar baja considerablemente.

En la implementación de la nube privada se ha considerado un módulo compute, un módulo de block storage con redundancia y un módulo controlador para la solución con openstack. Para cumplir con el planteamiento se presentan a continuación los elementos principales de la solución de nube privada:

NetShelter CX 38U



Precio: \$5,919.54

Características principales:

- Refrigeración integrada.
- Suspensión de ruido
- Sistema de distribución de energía.
- Almacenamiento de montaje en rack de 19 pulgadas mediante estándar EIA-310

Netgear GS748T



Precio: \$479.99

Características principales:

- 48 puertos
- Velocidad de transmisión 10Gb
- Diseño de eficiencia energética que cumple con IEEE802.3az

Router y firewall

Netgate 7100U



Precio: \$1199.00

Características principales:

- 2 Puertos SFP 10Gb
- Almacenamiento 32 GB eMMC.
- Memoria RAM 8GB DDR4
- CPU Intel Atom C33558 Quad Core 2.2GHz
- Firewall pfSense
- Expandible hasta 14 puertos

Nodo Compute

Dell PowerEdge R940 Rack Server



Precio: \$12,798.20

- Capacidad de hasta 8 Discos duros de 2.5"
- 1 x SSD 480 SATA 6Gbps.
- 2 x Intel Xeon Gold 5215 2.5Ghz 10 núcleos/20 hilos.
- 2 x 64 GFB 3200MT/s

Nodo Controller

HP ProLiant DL360 G7 1U



Precio: \$217.00

- 8 discos SAS de 146GB
- 2x Intel Xeon X5650 6 Core 2.66GHz
- Ram 32GB
- 4x Gigabit NIC

Nodo Block Storage

HP ProLiant DL360 G7 1U



Precio: \$217.00

- 8 discos SAS de 146GB
- 2x Intel Xeon X5650 6 Core 2.66GHz
- Ram 32GB
- 4x Gigabit NIC

SSD Block Storage

Samsung 860 Pro



Precio: \$845.00

- Capacidad: 4TB
- Conectividad SATA
- Velocidad de escritura secuencial de hasta 530 MB/s
- Velocidad de lectura secuencial de hasta 560 MB/s.

Elemento	Precio Unitario	Unidades	Total
NetShelter CX 38U	\$5,919.54	1	\$5,919.54
Ethernet Switch (NetShelter CX 38U)	\$479.99	1	\$479.99
Router/Firewall (Netgear GS748T)	\$1199.00	1	\$1199.00
Nodo compute	\$12,798.20	1	\$12,798.20
HP ProLiant DL360 G7 1U (Nodo controller y nodo block storage)	\$217.00	2	\$434.00
SSD Block Storage	\$845.00	2	\$1690.00
TOTAL			\$22,520.73

Asimismo, se deberán agregar gastos de mantenimiento estimados en \$150 de forma semestral, lo cual añaden \$300 a los gastos anuales de la solución de nube privada.

Anexo L Guía de instalación Rancher en OpenStack

Esta guía sirve para correr un cluster de kubernetes con Rancher en openstack.

Prerrequisitos:

- Tener acceso a la línea de comandos de OpenStack (OpenStack CLI)

1. Creación de nodos

Se necesita crear tres nodos para el cluster, Antes de crear estos nodos, se debe crear una red, una subred y un router. También se necesita crear un grupo de seguridad para que el tráfico de red puede llegar al cluster.

- Crear un proyecto

Crear un proyecto separará los recursos y hará más fácil la limpieza posterior. Correr los siguientes comandos:

```
openstack project create --domain default --description "RKE1  
Cluster" rke1
```

```
openstack role add --project rke1 --user admin admin
```

Actualizar las siguientes variables de entorno. Nota: Reemplazar <project_id> con el id de proyecto del paso anterior.

```
Export OS_PROJECT_ID=<project_id>
```

```
export OS_PROJECT_NAME=rke1
```

- Redes

Necesitaremos crear algunos recursos de red para los nodos.

- Crear red

```
openstack network create --project rke1 rke1
```

- Crear subred

```
openstack subnet create rke1-subnet --project rke1 --network rke1 \  
--subnet-range 172.31.0.0/28
```

- Crear router

```
openstack router create rke1-router --project rke1
```

- Agregar subnet al router

```
openstack router add subnet rke1-router rke1-subnet
```

- Designar gateway externo a router

```
openstack router set --external-gateway \  
$(openstack network list --external -f value -c ID) \  
rke1-router
```

- Crear grupo de seguridad

Se creará un grupo de seguridad que permita el tráfico en puertos comunes requerido por el sistema de despliegues de Kubernetes. Nota: La siguiente lista de grupos no es definitiva. En despliegues en producción se debe asegurar que los grupos de seguridad solo permitan el tráfico de red a los nodos y puertos que se necesiten.

```
openstack security group create rke1 --project rke1
openstack security group rule create --protocol icmp --dst-port 1:65535
rke1
openstack security group rule create --protocol tcp --dst-port 22:22 rke1
openstack security group rule create --protocol tcp --dst-port 53:53 rke1
openstack security group rule create --protocol tcp --dst-port 179:179
rke1
openstack security group rule create --protocol tcp --dst-port 6443:6443
rke1
openstack security group rule create --protocol tcp --dst-port 2380:2380
rke1
openstack security group rule create --protocol tcp --dst-port 7080:7080
rke1
openstack security group rule create --protocol tcp --dst-port 8472:8472
rke1
```

```
openstack security group rule create --protocol tcp --dst-port 8080:8080  
rke1
```

```
openstack security group rule create --protocol tcp --dst-port 9100:9100  
rke1
```

```
openstack security group rule create --protocol tcp --dst-port 10250:10250  
rke1
```

```
openstack security group rule create --protocol udp --dst-port 8472:8472  
rke1
```

```
openstack security group rule create --protocol tcp --dst-port 30000:32767  
rke1
```

- Crear par de llaves

Estas llaves serán utilizadas en el entorno de despliegue para acceder a los nodos.

- Generar par de llaves

```
ssh-keygen -t ed25519 -N '' -f /root/.ssh/id_rke1
```

- Subir llave pública a openstack

```
openstack keypair create --public-key /root/.ssh/id_rke1.pub rke1-  
key
```

- Crear instancias
 - Preparar Script instalación de Docker

Cada uno de los nodos necesita la instalación de Docker en ellos. Este script instalará cualquier dependencia después de que la máquina virtual es creada.

```
cat <<EOF > ./install_docker.sh
#!/bin/bash

curl https://releases.rancher.com/install-docker/20.10.sh | sh

sudo usermod -aG docker ubuntu

EOF
```

- Crear servidores

Estos nodos servirán como 3 nodos del cluster de kubernetes. Usaremos la bandera “--user data” para pasar al script que creamos en el paso anterior en los nodos. Esto instalará Docker en los nodos. También usaremos la bandera “-- max” para crear 3 nodos al mismo tiempo.

```
openstack server create --flavor m1.medium \
--image="Ubuntu 20.04 (focal-amd64)" \
--network rke1 \
--key-name rke1-key \
--security-group rke1 \
--user-data ./install_docker.sh \
--max 3 \
rke1
```


- Nodo de despliegue

Desplegaremos un nodo que usaremos para desplegar el cluster. Este nodo será usado para correr el instalador de RKE. No es necesario, pero es una buena idea mantener el ambiente de despliegue separado del cluster. Esto nos permitirá destruir fácilmente el ambiente de despliegue sin afectar el cluster.

```
openstack server create --flavor m1.medium \  
--image="Ubuntu 20.04 (focal-amd64)" \  
--network rke1 \  
--key-name rke1-key \  
--security-group rke1 \  
rke1-launcher2
```

- Agregar IP flotante al nodo de despliegue

En este punto debemos tener 4 nodos corriendo. Cada nodo actualmente es solo accesible desde la red privada. Necesitaremos agregar una IP flotante al nodo de despliegue para poder accederlo. Nota: deberá reemplazar <floating-ip> con la IP que creo para esto en los siguientes comandos.

```
openstack floating ip create \  
--description "RKE1 Cluster - Deployment Node" \  
--project rke1 \  
External
```

```
openstack server add floating ip rke1-launcher2 <floating-ip>
```

- Copiar llave SSH al nodo de despliegue

Debemos copiar la llave SSH creada anteriormente en el nodo de despliegue. Esto permitirá a RKE desplegar Kubernetes en los 3 nodos del cluster.

```
scp -i ~/.ssh/id_rke1 ~/.ssh/id_rke1 ubuntu@<floating-  
ip>:/home/ubuntu/.ssh/id_rsa
```

- SSH al nodo de despliegue

```
ssh -i ~/.ssh/id_rke1 ubuntu@<floating-ip>
```

2. Despliegue del cluster

- Instalar RKE

```
curl -OL
```

```
https://github.com/rancher/rke/releases/download/v1.3.14/rke\_linux-amd64
```

```
chmod +x rke_linux-amd64 && sudo mv rke_linux-amd64 /usr/local/bin/rke
```

- Instalar Kubectl

```
curl -LO https://dl.k8s.io/release/\$\(curl -L -s
```

```
https://dl.k8s.io/release/stable.txt\)/bin/linux/amd64/kubectl
```

```
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
```

- Preparar archivo de configuración

Crearemos un archivo de configuración para el cluster RKE. Cambiar “address” con las direcciones IP creadas para el nodo. Guardar el archivo con el nombre “rancer-cluster.yml”

Puedes ver las direcciones IP de tus nodos con el siguiente comando:

```
openstack server list --project rke1
```

- Archivo de configuración

```
nodes:
```

```
- address: <node_1_IP_address>
```

```
  user: ubuntu
```

```
  role:
```

```
    - controlplane
```

```
    - worker
```

```
    - etcd
```

```
- address: <node_2_IP_address>
```

```
  user: ubuntu
```

```
  role:
```

```
    - controlplane
```

```
    - worker
```

```
    - etcd
```

```
- address: <node_3_IP_address>
```

```
  user: ubuntu
```

```
  role:
```

```
    - controlplane
```

```
    - worker
```

```
    - etcd
```

```
services:
```

```
  etcd:
```

```
    snapshot: true
```

```
    creation: 6h
```

```
    retention: 24h
```

```
ingress:
```

```
  provider: nginx
```

```
  options:
```

```
    use-forwarded-headers: "true"
```

- Correr RKE

Este proceso tarda alrededor de 2 minutos. Después de completado deberíamos tener un cluster de Kubernetes corriendo.

```
rke up --config rancher-cluster.yml
```

Respuesta esperada:

```
...
```

```
INFO[0171] [ingress] Setting up nginx ingress controller
INFO[0171] [ingress] removing admission batch jobs if they exist
INFO[0171] [addons] Saving ConfigMap for addon rke-ingress-controller to
Kubernetes
INFO[0171] [addons] Successfully saved ConfigMap for addon rke-ingress-
controller to Kubernetes
INFO[0171] [addons] Executing deploy job rke-ingress-controller
INFO[0181] [ingress] removing default backend service and deployment if
they exist
INFO[0181] [ingress] ingress controller nginx deployed successfully
INFO[0181] [addons] Setting up user addons
INFO[0181] [addons] no user addons defined
INFO[0181] Finished building Kubernetes cluster successfully
```

3. Verificación de instalación

- Elegir configuración como predeterminado

```
mkdir ~/.kube && cp kube_config_rancher-cluster.yml ~/.kube/config
```

- Buscar recursos

Obtener pods en todos los namespaces.

```
kubectl get pods -A
```

Respuesta esperada.

```
ubuntu@rke1-launcher:~/rancher$ kubectl get pods -A
```

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
ingress-nginx	ingress-nginx-admission-create-9s6t2	0/1	Completed	0	3m55s
ingress-nginx	ingress-nginx-admission-patch-2brpz	0/1	Completed	0	3m55s
ingress-nginx	nginx-ingress-controller-j286h	1/1	Running	0	3m55s
ingress-nginx	nginx-ingress-controller-nm5m7	1/1	Running	0	3m55s
ingress-nginx	nginx-ingress-controller-x65pp	1/1	Running	0	3m55s
kube-system	calico-kube-controllers-74df54cbb7-49xm7	1/1	Running	0	4m25s
kube-system	canal-jkzvb	2/2	Running		

0			4m26s
kube-system	canal-mz67r	2/2	Running
...			

- Credenciales

Guarda y copia los siguientes archivos en una localización segura.

- ❖ rancher-cluster.yml: Archivo de configuración del cluster RKE.
- ❖ kube_config_cluster.yml: El archivo KubeConfig para el cluster, este archivo contiene credenciales para el acceso completo al cluster.
- ❖ rancher-cluster.rkestate: Archivo de estado del cluster de Kubernetes, contiene credenciales para el acceso completo al cluster.