

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



CURSO DE ESPECIALIZACIÓN INFRAESTRUCTURA CLOUD

**PROTOTIPO DE CENTRALIZACIÓN DE TRANSACCIONES EN
NUBE PARA LA EMPRESA VISIÓN Y DESARROLLO**

PRESENTADO POR:

MARTÍNEZ CORNEJO, CARLOS ALBERTO

BARRERA GARCÍA, LUIS SALVADOR

PARA OPTAR AL TÍTULO DE:

INGENIERO DE SISTEMAS INFORMÁTICOS

CIUDAD UNIVERSITARIA, NOVIEMBRE DE 2022

UNIVERSIDAD DE EL SALVADOR

RECTOR:

MSC. ROGER ARMANDO ARIAS ALVARADO

SECRETARIO GENERAL:

MSC. FRANCISCO ANTONIO ALARCON SANDOVAL

FACULTAD DE INGENIERÍA Y ARQUITECTURA

DECANO:

DOCTOR EDGAR ARMANDO PEÑA FIGUEROA

SECRETARIO:

ING. JULIO ALBERTO PORTILLO

ESCUELA DE INGENIERÍA DE SISTEMAS INFORMÁTICOS

DIRECTOR:

ING. RUDY WILFREDO CHICAS VILLEGAS

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

Trabajo de Graduación previo a la opción al Grado de:
INGENIERO DE SISTEMAS INFORMÁTICOS

Título:

**PROTOTIPO DE CENTRALIZACIÓN DE TRANSACCIONES EN
NUBE PARA LA EMPRESA VISIÓN Y DESARROLLO**

Presentado por:

**MARTÍNEZ CORNEJO, CARLOS ALBERTO
BARRERA GARCÍA, LUIS SALVADOR**

Trabajo de Graduación Aprobado por:

Docente Asesor:

MSc. DAMIÁN MORALES

SAN SALVADOR, NOVIEMBRE DE 2022

Trabajo de Graduación Aprobado por:

Docente Asesor:

MSc. DAMIÁN MORALES

Agradecimientos

Primeramente, agradezco a Dios por darme la vida, la salud, la fuerzas y la voluntad para alcanzar esta meta.

Agradezco a mi padre que a pesar de ya no estar con nosotros siempre fue un ejemplo a seguir y gran parte de la persona que soy ahora es gracias a él.

Agradezco a mi madre que a pesar de las limitantes siempre me apoyo para lograr salir adelante.

Luis Barrera

A mi familia, por permitirme culminar esta etapa de la vida. Especialmente a mi madre por el esfuerzo y apoyo que me brindó.

Carlos Martínez

Índice

1. Introducción	1
1. Resumen	3
3. Definición del proyecto	4
3.1 Identificación de la Organización	4
3.2 Contexto.....	5
3.3 Antecedentes	6
3.4 Planteamiento del problema	7
3.4.1 Pregunta de Investigación	7
3.4.2 Justificación	7
3.5 Objetivos	8
3.5.1 Objetivo General.....	8
3.5.2 Objetivos Específicos.....	8
3.6 Alcances y limitaciones	9
3.6.1 Alcances	9
3.6.2 Limitaciones.....	9
4. Marco Teórico	10
4.1 Computación en la nube	10
4.2 Virtualización.....	12
4.3 Contenedores con Docker	13
4.4 Kubernetes.....	13
4.5 Devops.....	14
5. Metodología de la investigación	16
5.1 Enfoque de la investigación	17
5.1.1 Enfoque cualitativo.....	17
5.1.2 Enfoque cuantitativo	18
5.2 Población y muestra.....	19
5.3 Instrumentos	21
5.3.1 Entrevista.....	21
5.3.2 Observación.....	21
5.4 Procedimientos	21
5.4.1 Fuentes de datos	22

5.4.2	Localización de las fuentes de datos	22
5.4.3	Medios o métodos para la recolección de datos	22
6.	Resultados	23
6.1	Resultados cualitativos	23
6.2	Resultados cuantitativos	24
6.3	Análisis de los resultados.....	26
7.	Desarrollo de la solución	28
7.1	Herramientas Utilizadas	28
7.1.1	OpenStack	28
7.1.2	Integración de Ceph con Openstack.....	32
7.1.3	Kubernetes	33
7.1.4	Integración de Ceph con Kubernetes.....	35
7.2	Análisis y Diseño.....	36
7.3	Prototipo.....	36
7.4	Software requerido.....	37
7.5	Hardware requerido	38
7.6	Diagrama del prototipo.....	40
7.7	Diagrama de redes.....	41
8.	Factibilidad	42
8.1	Costos en Nube Pública Google Cloud Platform:.....	42
8.2	Costos implementado nube privada.....	46
8.3	Comparación de costos	48
9.	Conclusiones y Recomendaciones	49
10.	Referencias	50
11.	Anexos	51
Anexo 1.	Introducción al diseño de bajo nivel	51
Anexo 2.	Objetivos del diseño de bajo nivel.....	52
2.1	Objetivo general.....	52
2.2	Objetivos específicos	52
Anexo 3.	Requerimientos técnicos del prototipo	53
3.1	Requerimientos de software	53
3.2	Requerimientos de hardware.....	53

Anexo 4. Diagrama del sistema	55
Anexo 5. Diagrama de red	56
Anexo 6. Despliegue de OpenStack	57
Anexo 7. Despliegue de clúster Ceph	59
Anexo 8. Despliegue de clúster Kubernetes	66
Anexo 9. Integración de OpenStack y Ceph	72
Anexo 10. Integración del clúster de Kubernetes con Ceph	76
Anexo 11. Instalación de servicios en clúster Kubernetes	77
Anexo 12. Instalación de aplicación en instancia OpenStack	84
Anexo 13. Equipos para la implementación en nube privada	85
Anexo 14. Encuesta para trabajadores administrativos de restaurantes	86

Índice de tablas

Tabla 1 Muestra de la investigación	20
Tabla 2 Resultado de las entrevistas	24
Tabla 3 Comparativa entre OpenStack y AWS	29
Tabla 4 Software requerido	38
Tabla 5 Hardware requerido.....	39
Tabla 6 Resumen de costo mensual por servicio.....	45
Tabla 7 Equipo necesario para implementación real del prototipo	47
Tabla 8 Comparación de costos entre nube publica y nube privada	48

Índice de ilustraciones

Ilustración 1 Pregunta 1 de la encuesta	25
Ilustración 2 Pregunta 2 de la encuesta	25
Ilustración 3 Pregunta 3 de la encuesta	26
Ilustración 4 Interacción entre servicios de OpenStack.....	32
Ilustración 5 Interacción entre Rados Gateway, Swift y Keystone.....	33
Ilustración 6 Clúster de Kubernetes	34
Ilustración 7 Pods dentro del clúster de Kubernetes	35
Ilustración 8 Integración entre Ceph y el clúster de Kubernetes	35
Ilustración 9 Prototipo.....	37
Ilustración 10 Diagrama del prototipo.....	40
Ilustración 11 Diagrama de redes.....	41
Ilustración 12 Precio de servicio de Cloud Storage Google Cloud Platform	43
Ilustración 13 Precio de servicio Pubsub Google Cloud Platform.....	43
Ilustración 14 Precio de servicio de Compute Engine Google Cloud Platform	44
Ilustración 15 Precio de disco persistente Google Cloud Platform	44
Ilustración 16 Precio de servicio Google Kubernetes Engine	45

1. Introducción

La forma en la que los negocios manejan sus operaciones está en constante evolución orientada a mejorar sus servicios, minimizar las pérdidas y maximizar las ganancias. Uno de los grandes factores que motiva este desarrollo es la tecnología.

La tecnología es el conjunto de instrumentos, recursos técnicos o procedimientos empleados en un determinado campo o sector. Estos recursos por lo general ayudan a hacer nuestras tareas más fáciles, en menor tiempo y con un estándar de calidad más alto.

Uno de los grandes avances tecnológicos de nuestros tiempos viene dado por el internet. Esta herramienta se ha vuelto indispensable para la mayoría de actividades cotidianas y laborales del mundo.

Este proyecto trata de cómo podemos mejorar las operaciones de un negocio a través de una infraestructura en la nube.

La solución será aplicada a una empresa cuyo rubro consiste en dar soporte y software a restaurantes. Dichos restaurantes poseen software que administran sus operaciones básicas de negocio como lo son: inventario, costos y ventas.

A pesar que dichos restaurantes poseen ya un sistema establecido para la operación de su negocio, carecen de centralización de datos; es decir, para los dueños de dos o más restaurantes les resulta complicado ver las operaciones totalizadas de todos sus negocios y es ahí donde entra nuestra solución.

Se desarrollará un prototipo de infraestructura en la nube la cual será administrada por la empresa Visión y Desarrollo y que pretende proveer las herramientas necesarias para recopilar las transacciones generadas por los diferentes restaurantes y poder procesar datos generales para todos los negocios de un determinado cliente.

1. Resumen

En el funcionamiento de los negocios es indispensable conocer a detalle el resultado de la operación de mismo. Dicho requerimiento se dificulta más cuando existen divisiones logísticas en el mismo. Para nuestro caso, estas divisiones son los locales donde funcionan las sucursales de un determinado restaurante.

Los dueños de los restaurantes requieren tener a disposición los datos de las operaciones de todas sus sucursales, sin embargo, esta operación se vuelve tediosa mientras más sucursales posee el restaurante.

Para agrupar la información y que los dueños de los restaurantes puedan verla de forma general se plantea la creación de un prototipo de infraestructura en la nube. Este cumplirá con la función de almacenar datos relevantes para los negocios, así como también desplegar nuevos servicios según lo requieran.

Se debe conocer las funcionalidades específicas que requieren los diferentes restaurantes que trabajan con la empresa Visión y Desarrollo, esto con el fin de brindar una solución más acertada y generalizada para futuros clientes de la empresa.

3. Definición del proyecto

3.1 Identificación de la Organización

Visión y Desarrollo es una empresa enfocada en la distribución de servicios informáticos a nivel nacional. Cuenta con múltiples clientes de carácter permanente.

Parte de las operaciones fundamentales en las que se centra dicha empresa son:

- Distribución de sistemas de punto de venta (POS) de terceros para restaurantes y hoteles.
- Desarrollo de aplicaciones propias en múltiples plataformas enfocadas en el ámbito comercial administrativo.
- Venta de equipo informático destinado para comercio, oficina o uso personal.
- Reparación de equipos informáticos.
- Soporte técnico para los sistemas que la empresa distribuye.

Visión y Desarrollo surgen en el año 2015 de la necesidad de los negocios de alimentos y bebidas por tener acceso a tecnologías que apoyen en la administración de los negocios.

Luego de realizar un estudio de mercado salvadoreño, se determinó que hay pocos proveedores que combinen productos de calidad y un servicio de asistencia técnica que cubre las diferentes necesidades de sus clientes.

Un restaurante que controle su inventario, su facturación y el resto de operaciones de manera manual, está en clara desventaja, es así que la empresa se coloca entre las primeras 5 del mercado en los primeros dos años.

Los productos que ofrece la empresa son:

- Sistemas Informáticos para Hoteles y Restaurantes, Software administrativo financiero que ayuda con la operación de los negocios controlando el tipo de servicios ofrecidos (Menú a la Carta, Servicios Domicilio, Servicio para Llevar).
- Gestiona el ingreso de los platillos o bebidas requeridos por los clientes por medio de los meseros y realizando la gestión de entrega.
- Equipo Informático especializado para el rubro de restaurantes y hoteles.
- Servicio de Asistencia a los clientes por medio de llamadas telefónicas, asistencia remota y asistencias presencial en caso que sea requerido.

Estos sistemas controlan el ingreso y salida de mercadería (materia prima), además son los que aseguran la correcta facturación del consumo realizado por los clientes.

La empresa en estudio tiene el soporte de parte de proveedores nacionales e internacionales, en el caso de los sistemas para restaurantes son adquiridos a una empresa costarricense por medio de licencias. Esto implica que tiene la distribución exclusiva en El Salvador.

También realiza compras a proveedores locales donde adquiere los equipos necesarios para la instalación del sistema informático.

En caso que sea necesario la empresa también realiza importaciones desde proveedores en Estados Unidos.

La empresa está ubicada en Calle Chiltiupan 1, Cd Merliot, Santa Tecla, La Libertad.

3.2 Contexto

Uno de los principales problemas con los que cuenta la empresa visión y desarrollo es que no cuenta con un servicio para proveer información centralizada sobre

las transacciones que se llevan a cabo en las diferentes cadenas de restaurantes a las que ofrece sus productos.

El contar con información centralizada sobre las diferentes transacciones es crucial para que los dueños del negocio puedan tener un panorama más amplio y con base a esta información puedan tomar decisiones tácticas y estratégicas que ayuden a la optimización de los procesos internos del negocio, aumentar las ventas y por ende incrementar las utilidades de la empresa.

Que la centralización de esta información se encuentre en una infraestructura de nube moderna permitirá a la empresa que su departamento de informática dé el primer paso para la migración de toda la suite hacia la nube y además le permitirá estar en una posición ventajosa con respecto a otras del mismo rubro y de esta manera ayudará a mejorar las operaciones del negocio de sus clientes y por lo tanto al crecimiento de la empresa.

3.3 Antecedentes

La suite de aplicaciones cumplía en sus inicios con las necesidades de los negocios a los que visión y desarrollo brindaba sus servicios, sin embargo, el diseño original de la suite que se provee, no contemplaba que en algún momento algunas cadenas de restaurantes iba a crecer de tal manera que sería necesario, tener un control y una visión macro sobre las diferentes operaciones que se llevan a cabo en las diferentes restaurantes, ya que los aplicativos utilizados operan de forma local; esto significa que cada aplicación tiene información únicamente del restaurante en el que se encuentra instalado.

Actualmente, para la elaboración de reportes generales, se tiene que solicitar a cada restaurante los diferentes tipos de reportes. Un ejemplo de esta problemática sucede cuando se desea mover productos alimenticios para la preparación de platillos para evitar el vencimiento de los productos (usualmente moviendo productos próximos a caducar hacia los restaurantes con mayor demanda) para determinar de dónde y hacia donde mover los productos se tiene que tener un reporte de inventario de cada restaurante de forma separada.

3.4 Planteamiento del problema

3.4.1 Pregunta de Investigación

¿Se puede crear una Infraestructura en la Nube que garantice alta disponibilidad, escalabilidad, seguridad y flexibilidad que permita la centralización de las transacciones llevadas a cabo en los diferentes restaurantes a los cuales la empresa visión y desarrollo brinda sus servicios?

3.4.2 Justificación

Al no contar con información centralizada es muy difícil para los clientes con diferentes sucursales contar con una herramienta de tipo Big Data que les permita hacer un análisis sobre las operaciones de los restaurantes para la toma de decisiones o mejoras en la eficiencia de procesos.

El no contar con reportes generales es uno de los principales obstáculos que afronta los clientes, ya que no se permite ver el estado actual de las sucursales a un nivel macro, por ejemplo: ventas en el último mes, restaurantes con más ventas, platillos

preferidos por los clientes, etc. Contar con la centralización de transacciones en un entorno de nube presenta una posición de ventaja competitiva frente a otras empresas en el mismo rubro ya que se pueden aprovechar todas las bondades que tener una infraestructura en la nube implica.

3.5 Objetivos

3.5.1 Objetivo General

- Diseñar e Implementar un prototipo de una Plataforma como Servicio (PaaS) para la orquestación de servicios con el propósito de centralizar las transacciones y disponer de un ambiente con capacidad incremental.

3.5.2 Objetivos Específicos

- Permitir la configuración dentro de los contenedores para añadir volúmenes de almacenamiento.
- Permitir el despliegue de un servicio de base de datos dentro de los contenedores.
- Permitir el despliegue de un servicio de mensajería dentro de los contenedores.
- Permitir el despliegue de un servicio de almacenamiento de archivos dentro de los contenedores.

3.6 Alcances y limitaciones

3.6.1 Alcances

- Se pretende optimizar el acceso a la información de los clientes de la empresa Visión y Desarrollo mediante una solución de infraestructura en nube privada para la centralización de datos en la nube.

3.6.2 Limitaciones

- La falta de recursos económicos nos obliga a desarrollar la solución a manera de prototipo virtual.

4. Marco Teórico

4.1 Computación en la nube

De acuerdo a los autores (Hernandez Quintero & Florez Fuente, 2014) describen la computación en la nube de la siguiente manera:

Básicamente la computación en la nube consiste en los servicios ofrecidos a través de la red, tales como: correo electrónico, almacenamiento, aplicaciones, entre otras., los cuales son normalmente accesibles mediante un navegador web. Al utilizar estos servicios, la información utilizada y almacenada, así como la mayoría de las aplicaciones requeridas, son procesados y ejecutados por un servidor en Internet.

Características de la computación en la nube

1. Autoservicio a la carta: Un consumidor puede abastecerse unilateralmente de capacidades de computación, como tiempo de servidor y almacenamiento en red, según sus necesidades, de forma automática sin requerir la interacción humana con cada proveedor de servicios.
2. Amplio acceso a la red: Las capacidades están disponibles en la red y se accede a ellas a través de mecanismos estándar que fomentan el uso por parte de plataformas de clientes heterogéneas tanto ligeras como pesadas.
3. Reservas de recursos en común: Los recursos computacionales del proveedor se ponen en reservas en común para que puedan ser utilizados por múltiples consumidores que utilicen un modelo de multiposesión, con diferentes recursos físicos y virtuales asignados dinámicamente y reasignados en función de la demanda de los consumidores.

4. Rapidez y elasticidad: Las capacidades pueden suministrarse de manera rápida y elástica, en algunos casos de manera automática, para poder realizar el redimensionado correspondiente rápidamente.
5. Servicio supervisado: Los sistemas de nube controlan y optimizan el uso de los recursos de manera automática utilizando una capacidad de evaluación en algún nivel de abstracción adecuado para el tipo de servicio.
6. Es auto reparable: En caso de surgir un fallo, el último respaldo (backup) de la aplicación se convierte automáticamente en la copia primaria y a partir de ésta se genera uno nuevo.
7. Es escalable: Todo el sistema y su arquitectura son predecible y eficiente.
8. Virtualización: Las aplicaciones son independientes del hardware en el que corran, incluso varias aplicaciones pueden corren en una misma máquina o una aplicación puede usar varias máquinas a la vez.
9. Posee un alto nivel de seguridad: El sistema está creado de tal forma que permite a diferentes clientes compartir la infraestructura sin preocuparse de ello y sin comprometer su seguridad y privacidad; de esto se ocupa el sistema proveedor que se encarga de cifrar los datos.
10. Disponibilidad de la información: No se hace necesario guardar los documentos editados por el usuario en su computadora o en medios físicos propios ya que la información radicará en Internet permitiendo su acceso desde cualquier dispositivo conectado a la red (con autorización requerida).

(Hernandez Quintero & Florez Fuente, 2014, págs. 47-48)

4.2 Virtualización

Según los autores (Hernandez Quintero & Florez Fuente, 2014) describen la virtualización en la nube de la siguiente manera:

La virtualización es un elemento fundamental en el desarrollo óptimo de la computación en la nube, y se enfoca principalmente a la plataforma. Puede considerarse como una abstracción de los recursos tecno-lógicos que permite a los servidores crear dispositivos virtuales los cuales pueden ser usados para aumentar dichos recursos más que como sistemas separados. Mediante la virtualización se permite tratar a un servidor como muchos servidores. Otro método usado es el clustering, que consiste en tratar a muchos servidores como uno solo. Esto permite muchas mejoras como:

- Reducción de los costos de espacio y consumo.
- Rápida incorporación de nuevos recursos para los servidores virtualizados.
- Administración global centralizada y simplificada.
- Facilidad para la creación de entornos de test que permiten poner en marcha nuevas aplicaciones sin detener el desarrollo, agilizando el proceso de las pruebas.
- Aislamiento: un fallo en una máquina virtual no afecta al resto de máquinas virtuales.

(Hernandez Quintero & Florez Fuente, 2014, pág. 48)

4.3 Contenedores con Docker

Según el sitio oficial (Red Hat, 2018) Docker se define como:

La tecnología Docker utiliza el kernel de Linux y sus funciones, como los grupos de control y los espacios de nombre, para dividir los procesos y ejecutarlos de manera independiente. El propósito de los contenedores es ejecutar varios procesos y aplicaciones por separado para que se pueda aprovechar mejor la infraestructura y, al mismo tiempo, conservar la seguridad que se obtendría con los sistemas individuales.

Las herramientas de contenedores, como Docker, proporcionan un modelo de implementación basado en imágenes, que permite compartir una aplicación o un conjunto de servicios con todas sus dependencias en varios entornos. Docker también automatiza la implementación de las aplicaciones (o los conjuntos de procesos que las constituyen) en el entorno de contenedores.

Estas herramientas están diseñadas a partir de los contenedores de Linux, por eso la tecnología Docker es sencilla y única. Además, ofrecen a los usuarios acceso sin precedentes a las aplicaciones, la posibilidad de realizar implementaciones en poco tiempo y el control sobre las versiones y su distribución.

(Red Hat, 2018)

4.4 Kubernetes

Según la documentación oficial de (Kubernetes, 2022) Kubernetes se define como:

Kubernetes (K8s) es una plataforma de código abierto para automatizar la implementación, el escalado y la administración de aplicaciones en contenedores.

Kubernetes agrupa los contenedores que conforman una aplicación en unidades lógicas para una fácil administración y descubrimiento. Kubernetes se basa en 15 años de experiencia en la ejecución de cargas de trabajo de producción en Google, combinada con las mejores ideas y prácticas de la comunidad.

Preparado para trabajar a escala

Diseñado sobre los mismos principios que le permiten a Google ejecutar miles de millones de contenedores cada semana, Kubernetes puede escalar sin aumentar su equipo de operaciones.

Preparado para cualquier complejidad

Ya sea para pruebas locales o cargas de trabajo de una empresa global, la flexibilidad de Kubernetes crece contigo para entregar sus aplicaciones de manera consistente y fácil sin importar la complejidad.

Preparado para ejecutarse en cualquier lugar

Kubernetes es código abierto lo que le brinda la libertad de aprovechar su infraestructura propia (on-premises), híbrida o de nube pública, lo que le permite mover las cargas de trabajo sin esfuerzo dónde quiera.

(Kubernetes, 2022)

4.5 Devops

El sitio oficial de (Red Hat, 2018) define el termino DevOps de la siguiente manera:

El término "DevOps" es una combinación de las palabras "development" (desarrollo) y "operations" (operaciones), pero representa un conjunto de ideas y prácticas que van más allá de ambos conceptos, ya sea que estén juntos o

separados. DevOps incluye sistemas de seguridad, maneras de trabajar en colaboración, análisis de datos, entre otras características. ¿Pero qué es?

DevOps describe los enfoques para agilizar los procesos con los que una idea (como una nueva función de software, una solicitud de mejora o una corrección de errores) pasa del desarrollo a la implementación, en un entorno de producción en que puede generar valor para el usuario. Estos enfoques requieren que los equipos de desarrollo y operaciones se comuniquen con frecuencia y aborden su trabajo con empatía hacia sus compañeros de equipo. También es necesario contar con capacidad de ajuste y tener flexibilidad a la hora de preparar los sistemas. Con DevOps, es posible brindar una mayor potencia a quienes más la necesitan gracias al autoservicio y la automatización. Los desarrolladores, que generalmente realizan codificaciones en un entorno de desarrollo estándar, trabajan en estrecha colaboración con los equipos de operaciones de TI para agilizar el diseño, las pruebas y el lanzamiento de los sistemas de software, sin comprometer la confiabilidad.

Por supuesto, esto implica cambios más frecuentes en el código y el uso más dinámico de la infraestructura. Las estrategias tradicionales de gestión no logran satisfacer este aumento de la demanda. Necesitará hacer algunos cambios para obtener una ventaja competitiva.

(Red Hat, 2018)

5. Metodología de la investigación

(Bastis Consultores, 2021) define el Enfoque de la Investigación de la siguiente manera:

El enfoque de la investigación es un plan y un procedimiento que consta de los pasos de las hipótesis generales a los métodos detallados de recogida, análisis e interpretación de datos. El enfoque de la investigación se divide esencialmente en dos categorías: el enfoque de la recogida de datos y el enfoque del análisis o razonamiento de los datos.

(Bastis Consultores, 2021)

En esta sección se describe el diseño metodológico de investigación utilizado para la recolección de información que nos ayudara a la toma de decisiones con respecto a la solución planteada para la problemática presentada. La investigación proveerá la información necesaria para determinar si dicha solución es viable o no.

Para dicha investigación, abarcaremos los siguientes enfoques relacionados al ámbito del proyecto:

- **Tecnológico:** Con este enfoque conoceremos la capacidad tecnológica con la que cuenta la empresa Visión y Desarrollo, así como también la capacidad que poseen sus empleados del departamento de informática para el uso de una infraestructura nube de tipo privada. También se requiere investigar la tecnología con la que cuentan los restaurantes donde funcionara dicha solución.
- **Organizacional:** La investigación se centrará en el departamento de TI de la empresa Visión y Desarrollo, así como también la estructura organizacional de los

restaurantes a los que se les brindaría la solución; esto con el fin de conocer las funciones de los miembros y como las desempeñan.

- Económico: Se requiere investigar la capacidad que tiene la empresa Visión y Desarrollo, así como los restaurantes de incurrir en gastos para la actualización de tecnologías u otros insumos para la implementación de una infraestructura en la nube.

5.1 Enfoque de la investigación

La investigación se llevará a cabo mediante el enfoque mixto, teniendo en cuenta los enfoques cuantitativo y cualitativo. Esto con el objetivo de ver la información desde diferentes puntos de vista y de este modo profundizar en el análisis de los datos de forma más certera.

Según (Corbetta, 2007):

Si tuviéramos que indicar un elemento único y general que marque la diferencia entre los dos tipos de investigación, probablemente se encontraría en la estructuración de las fases que abarcan desde el planteamiento de la pregunta inicial hasta las conclusiones finales.

(Corbetta, 2007)

5.1.1 Enfoque cualitativo

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

Para comprender la problemática que existe en el manejo de la información de múltiples restaurantes pertenecientes a un mismo propietario o empresa, es necesario conocer la forma en que estos operan. Debemos investigar la información concreta y precisa que manejan y como centralizan la información actualmente. También es necesario conocer a los actores implicados en la operación y cada una de sus funciones definidas por prioridad.

Por medio de esta información se comprenderá la utilidad que tiene la centralización de datos de los diferentes restaurantes para la toma de decisiones por parte de sus propietarios.

5.1.2 Enfoque cuantitativo

Según (Metodología Cuantitativa, s.f.):

La investigación cuantitativa es aquella en la que se recogen y analizan datos cuantitativos sobre variables y estudia las propiedades y fenómenos cuantitativos. Ente las técnicas de análisis se encuentran: análisis descriptivo, análisis exploratorio, inferencial univariable, inferencial multivariado, modelización y contrastación.

(Metodología Cuantitativa, s.f.)

Para esta problemática también debemos aplicar un enfoque cuantitativo, por ser una metodología de investigación que permite asignar un valor o un grado numérico representativo a las variables de interés en los modelos de recolección de datos utilizados, esto debido a que el manejo optimizado de los datos de los restaurantes de forma centralizada supondría una mejora administrativa y por ende económica. Así como

también se debe controlar los gastos de operación de la infraestructura en la nube y su despliegue.

5.2 Población y muestra

Para el desarrollo de la investigación, tomaremos como población todo el personal del departamento de TI de la empresa Visión y Desarrollo y adicionalmente a todos los actores que de manera directa o indirecta participan en la recolección y centralización de la información de las transacciones que se llevan a cabo en los diferentes restaurantes.

Cargo	Número de personas	Área
Desarrollador de Software	2	Departamento de TI de Visión y Desarrollo
Administrador de Sistemas	1	Departamento de TI de Visión y Desarrollo
Jefe de departamento de TI	1	Departamento de TI de Visión y Desarrollo
Cajero	8	Restaurantes
Gerente de Restaurante	4	Restaurantes
Propietario de Restaurantes	2	Restaurantes

Tabla 1 Muestra de la investigación

Para el área de restaurante la muestra viene reflejada por cualquier restaurante de los mismos. Esto debido a que, en todos ellos, la operación del negocio es la misma. Solo es posible que difieran en afluencia de comensales. Con respecto a la muestra del departamento de TI de la empresa Visión y Desarrollo, la muestra será toda la población por lo cual no se requiere fórmula para calcular la muestra; esto debido a que el personal de TI de la empresa Visión y Desarrollo cuenta únicamente 4 personas.

La empresa Visión y Desarrollo actualmente le brinda sus servicios a 34 restaurantes, pero para la muestra estudiada se seleccionaron 2 restaurantes, los cuales cuentan con sus respectivas sucursales. De estos 2 restaurantes se seleccionaron 2 sucursales de cada uno. Este muestreo se hace así con el fin de estandarizar la recolección de los datos para poder brindar soporte a múltiples restaurantes.

Para el manejo de los datos debemos hablar directamente con el Gerente de todos los restaurantes y entrevistarlos acerca de que datos son los necesarios para su operación e investigar que otros datos podrían mejorar el desempeño de sus funciones administrativas.

5.3 Instrumentos

5.3.1 Entrevista

Se entrevistarán a los miembros de la empresa Visión y Desarrollo con el objetivo de conocer de manera técnica la forma en la que los restaurantes operan los datos para generar su información.

Se realizará una entrevista al gerente del restaurante y al propietario del mismo con el objetivo de indagar acerca de los datos necesarios para la operación logística del negocio. Esto ayudara a definir qué información se requiere centralizar para ayudar en el funcionamiento de los restaurantes pertenecientes a un mismo propietario.

5.3.2 Observación

Se debe llevar a cabo un estudio de campo donde el objetivo principal es presencial la forma en la que se manejan los datos importantes para la operación de los restaurantes. Esto permitirá determinar desde el área informática como debe procesarse la información y cuál es su importancia dentro de la solución planteada.

5.4 Procedimientos

Una vez elegido el método de investigación, se determina la población y la muestra, y también se han desarrollado herramientas, ahora es necesario definir el procedimiento a seguir y aplicar técnicas de recopilación de datos a las muestras identificadas según el método.

5.4.1 Fuentes de datos

Para nuestro caso, la fuente de datos se divide en 2 secciones: los datos que nos pueden proveer los empleados de la empresa Visión y Desarrollo y los datos que nos pueden proveer los empleados y encargados de los restaurantes.

5.4.2 Localización de las fuentes de datos

La localización de los datos pertenecientes a la empresa Visión y Desarrollo se encuentran dentro de la misma empresa en sus diferentes áreas operativas.

La localización de datos pertenecientes a los restaurantes se encuentra en la administración interna de cualquiera de los restaurantes y en la administración global de los mismos.

5.4.3 Medios o métodos para la recolección de datos

Es a través de entrevistas semiestructuradas que realizaremos la recolección de datos en ambas fuentes. La entrevista nos permitirá recolectar información que posteriormente podremos procesar para determinar problemas y necesidades de la operación de trabajo.

6. Resultados

Para el desarrollo del proyecto se realizó una serie de entrevistas a los actores principales del caso de estudio, esto con el fin de recabar información que ayude al desarrollo de una solución concreta para el prototipo planteado.

6.1 Resultados cualitativos

Para la recolección de datos utilizamos la entrevista para los actores principales del caso de estudio.

(Murillo Torrecilla) define la entrevista de la siguiente manera:

La entrevista es la técnica con la cual el investigador pretende obtener información de una forma oral y personalizada. La información versará en torno a acontecimientos vividos y aspectos subjetivos de la persona tales como creencias, actitudes, opiniones o valores en relación con la situación que se está estudiando.

(Murillo Torrecilla, pág. 6)

A partir de la información recabada a través de las entrevistas realizadas a los actores del caso de estudio, se logró establecer las siguientes categorías.

- Datos dispersos
- Administración complicada
- Demora en el análisis de la información

Resultados de las entrevistas	
Datos dispersos	Se complica la obtención de los datos ya que hay que consultar restaurante por restaurante.
Administración complicada	Los sistemas deben administrarse individualmente por restaurante, así como la información que generan.
Demora en el análisis de la información	Muchas veces se requiere tener información general de gastos o ventas de todas las sucursales y generar estos reportes conlleva mucho tiempo ya que hay que consultar a cada una de las sucursales

Tabla 2 Resultado de las entrevistas

Dado que la empresa Visión y Desarrollo maneja un número de restaurantes con diversas sucursales, la obtención de datos se complica, ya que los datos varían de sucursal en sucursal sin importar si son de la misma cadena de restaurantes. A esto se le suma la complejidad administrativa debido a la falta de centralización de la información. Además, al momento de realizar la entrevista, la administración de los restaurantes no contaba con información actualizada sobre las operaciones de los mismos.

6.2 Resultados cuantitativos

Encuesta para trabajadores administrativos de restaurantes

Resultados de la encuesta realizada a los 14 miembros que conforman la muestra del personal de trabajo administrativo de los restaurantes

Pregunta 1: ¿Está de acuerdo en una forma de centralizar los datos administrativos de las diferentes sucursales para su consulta?

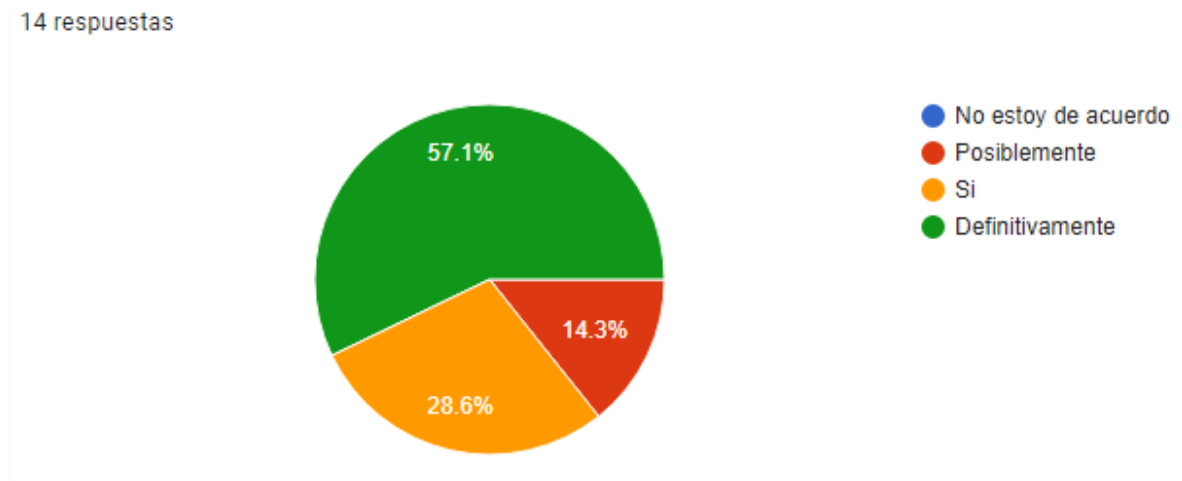


Ilustración 1 Pregunta 1 de la encuesta

Como se puede ver la mayoría de los encuestados están de acuerdo en la centralización de la información, no obstante, un pequeño número (14.3%) no comparte la iniciativa.

Pregunta 2: ¿Estaría de acuerdo en utilizar una herramienta para el envío de los datos administrativos a una ubicación en la nube?

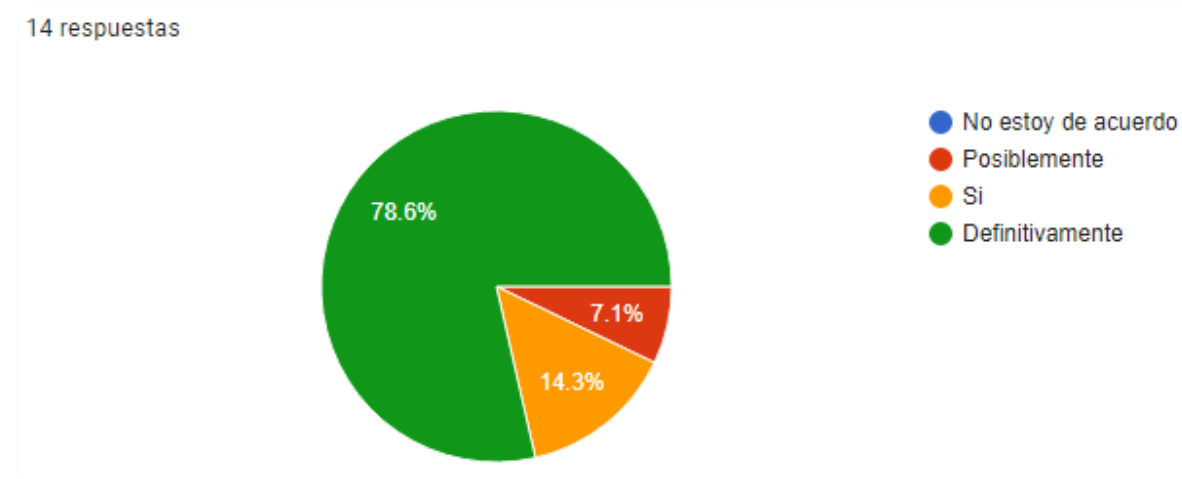


Ilustración 2 Pregunta 2 de la encuesta

La mayoría de los encuestados están a favor de utilizar una herramienta para el envío de los datos administrativos (transacciones) hacia la nube.

Pregunta 3: ¿Considera que reduciría su trabajo al centralizar los datos administrativos del restaurante?

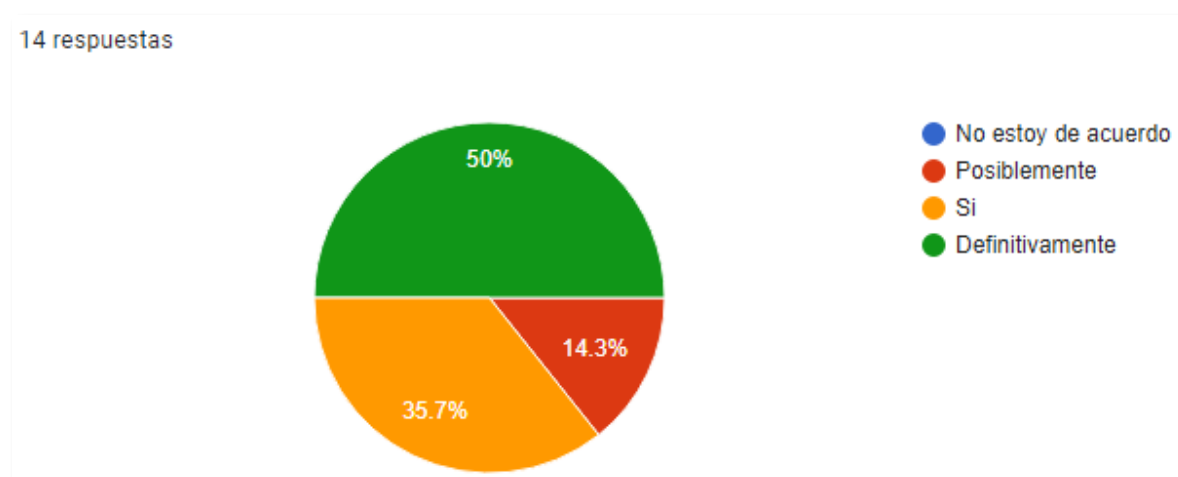


Ilustración 3 Pregunta 3 de la encuesta

La mitad de los encuestados consideran rotundamente que la centralización de datos reduciría su trabajo, un poco más de un tercio de los encuestados (35.7%) consideran que su trabajo se vería reducido y solo un 14.3% de los encuestados consideran que posiblemente se reduciría su trabajo.

6.3 Análisis de los resultados

Con base al resultado de los datos cualitativos y cuantitativos podemos evidenciar lo siguiente:

- A los miembros de la empresa Visión y Desarrollo se les complica trabajar con datos dispersos de cada uno de los restaurantes ya sea para el desarrollo de nuevas aplicaciones o la generación de reportes de los mismos.
- El personal administrativo de los restaurantes presenta demora en la recopilación e interpretación de los datos.
- El personal administrativo de los restaurantes considera beneficioso la centralización de los datos para agilizar su procesamiento.
- El personal administrativo de los restaurantes estaría de acuerdo en transferir datos de los restaurantes a una ubicación en la nube para agilizar su procesamiento.

Para brindar una solución acorde a las necesidades de los restaurantes, se debe poder integrar con las aplicaciones que llevan el control administrativo de los mismos. La empresa Visión y Desarrollo tiene estandarizado en sus restaurantes el siguiente software:

EWAITER POS: Es un sistema de punto de venta (POS) de origen costarricense que cumple con las funcionalidades básicas de un negocio de restaurantes.

El sistema administra las transacciones de manera local y tiene la capacidad de generar reportes con los datos que almacena.

La cantidad de transacciones que manejan los restaurantes varían dependiendo el lugar y especialización de los mismos, pero en promedio cada negocio maneja **800** transacciones por día. Estos son los datos que debemos migrar a la infraestructura en la nube.

7. Desarrollo de la solución

7.1 Herramientas Utilizadas

7.1.1 OpenStack

Retomando la necesidad de la empresa la cual consiste en la centralización de información, especialmente las transacciones llevadas a cabo en los diferentes restaurantes, pero sin dejar de lado que la solución brindada sea una base para la migración gradual de la empresa hacia la nube, se ha decidido optar por una solución genérica y versátil usando OpenStack. OpenStack es una plataforma de tecnología abierta y libre que utiliza recursos virtuales agrupados para diseñar y gestionar nubes privadas y públicas. Las herramientas que la componen se denominan "proyectos" y se encargan de los servicios principales de computación en la nube, es decir, de la informática, las redes, el almacenamiento, las identidades y las imágenes. Además, se puede agrupar más de una docena de proyectos opcionales para crear nubes únicas que pueden implementarse.

Previo a realizar una descripción más detallada las herramientas utilizadas, se listan un comparativo de los servicios más utilizados ofrecidos por Amazon Web Services (El proveedor de servicios de nube pública más popular) y su equivalente en OpenStack:

Servicio Ofrecido	AWS	OpenStack
Servidores Virtuales	EC2	Nova Instance
Dockers	ECS	Magnum
Escalabilidad	AWS Scaling	Heat with Scaling

Balaneo de carga	Elastic	Load	LBaaS
	Balancing		
API	EC2 API		Openstack API
GUI	Console		Horizon
Almacenamiento de Objetos	S3		Swift
Almacenamiento de tipo Block	EBS		Cinder
Redes	Networking		Neutron
Computación	VM		Instance
Seguridad/Identificación	IAM		Keystone
Orquestación	CloudFormation		Heat
Almacenamiento	Glacier		Swift
Plantilla de Imágenes	Amazon Machine Images		Glance
Administrador DNS	Route 53		Designate
Base de datos relacionales	RDS		Trove

Tabla 3 Comparativa entre OpenStack y AWS

Como se puede observar para cada servicio ofrecido en la nube pública existe un equivalente en OpenStack. Una diferencia fundamental a tomar en consideración es que en la solución brindada se está realizando toda la implementación y configuración de cada servicio mientras que si la solución se hubiera desarrollado con nubes públicas únicamente se hubiera hecho uso de los servicios y con la posibilidad de una

configuración al mínimo; esto puede ser una variable muy importante a tener en cuenta para proyectos en donde los requisitos pueden no encajar con los servicios brindados en las nubes públicas.

OpenStack encaja con los requerimientos establecidos en el documento y brinda todos los componentes y servicios necesarios para que la empresa comience su migración gradual hacia la nube. Además, presenta muchas ventajas frente a otras opciones, resaltando por ejemplo su licencia ya que se distribuye bajo una licencia de código abierto Apache, con lo cual ésta hereda todas las ventajas del software libre al tener el código abierto a toda la comunidad. Cuenta con el apoyo por parte de grandes empresas como Red Hat, AT&T, HP, IBM, etc. para seguir evolucionando como nube privada según sean las necesidades del mercado. Es una solución madura y flexible que cuenta con una variedad de servicios y alternativas de éstos permitiendo una adaptación con otros componentes según las necesidades particulares. Según lo expuesto, se elige Openstack como el elemento clave dentro de la solución.

Típicamente cuando se realiza un despliegue de Openstack se requieren de algunos servicios imprescindibles para el funcionamiento del mismo, estos servicios son:

- Keystone: autentica y autoriza todos los servicios de OpenStack. También es el catálogo de extremo de todos los servicios.
- Cinder: proporciona almacenamiento permanente en bloques. Puede acceder a él a través de una API de autoservicio.

- Glance: almacena y recupera imágenes del disco de la máquina virtual desde distintas ubicaciones.
- Nova: es una herramienta integral de gestión y acceso para los recursos informáticos de OpenStack que controla su planificación, creación y eliminación.
- Neutrón: conecta las redes a través de otros servicios de OpenStack.

Además, si se quiere contar con un Dashboard para la administración básica de algunos componentes y servicios de Openstack se puede instalar opcionalmente Horizon:

- Horizon: es la implementación canónica del Dashboard de Openstack, que proporciona una interfaz de usuario basada en la web para los servicios de OpenStack, incluidos Nova, Swift, Keystone, etc.

En la ilustración 4 se puede ver como interactúan todos estos servicios dentro de Openstack. La comunicación se realiza por medio de servicios rest.

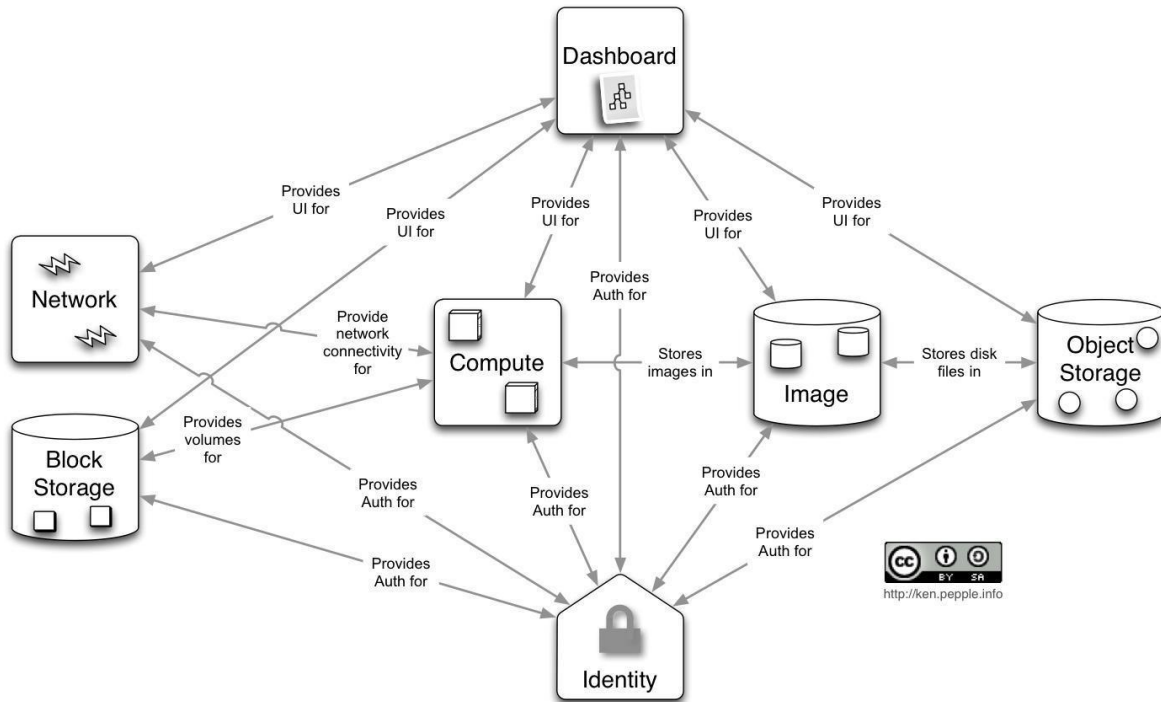


Ilustración 4 Interacción entre servicios de OpenStack

Con el despliegue de Openstack se ofrece a la empresa una amplia gama de servicios, como lo es el despliegue de diferentes instancias con diferentes imágenes, sistemas operativos, sabores etc. También en OpenStack se pueden desplegar sistemas de tipo legacy que la empresa utiliza para sus operaciones diarias configurando diferentes máquinas virtuales según sea la necesidad.

7.1.2 Integración de Ceph con Openstack

Además, se aprovecha la versatilidad y flexibilidad de Openstack de acoplarse con otros productos como Ceph, que es una completa solución de almacenamiento que dispone de un sistema de archivos propio, el Ceph File System (CephFS). para crear un clúster de almacenamiento, pero siempre tomando en consideración que este servicio se ha diseñado en torno a Openstack como eje central de la solución.

En la ilustración 5 se puede observar cómo Openstack interactúa con Ceph por medio de keystone y Swift API, conectándose a RADOS Gateway del clúster de Ceph.

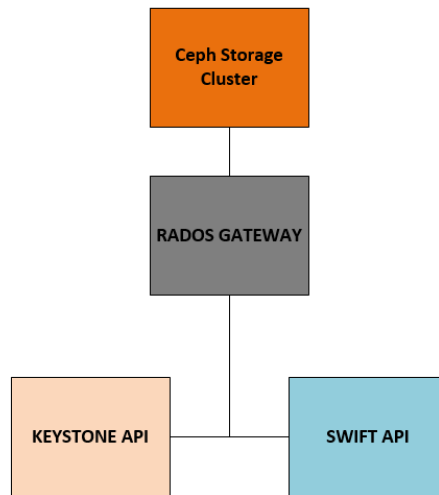


Ilustración 5 Interacción entre Rados Gateway, Swift y Keystone

7.1.3 Kubernetes

Por otro lado, para la orquestación de contenedores se ha optado por un clúster de Kubernetes que es un sistema de código libre para la automatización del despliegue, ajuste de escala y manejo de aplicaciones en contenedores que fue originalmente diseñado por Google y donado a la Cloud Native Computing Foundation. Soporta diferentes entornos para la ejecución de contenedores, incluido Docker. La configuración del clúster cuenta con un nodo master y dos workers) número que se puede ampliar según la necesidad). Se opta por esta solución ya que cuenta con una comunidad muy amplia y una documentación excelente.

En la ilustración 6 se puede ver el detalle de la configuración utilizada para el clúster.

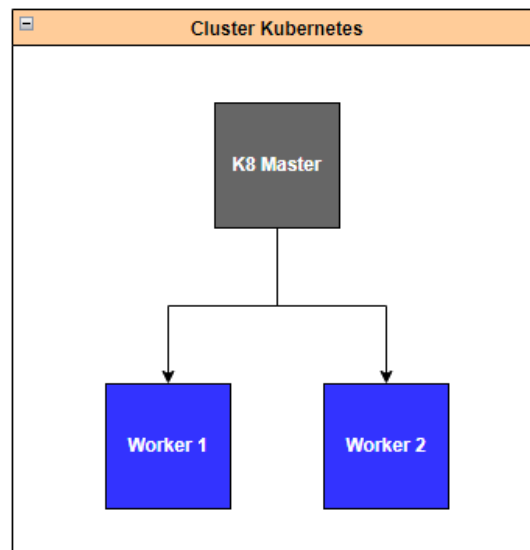


Ilustración 6 Clúster de Kubernetes

Dentro del clúster también se desplegó un contenedor de base de datos tipo MySQL demostrando la facilidad del despliegue de cualquier tipo de base de datos. Además, se expuso un servicio sobre el despliegue para poder ser consumido fuera del clúster.

También dentro del clúster de Kubernetes se ha desplegado un contenedor de Apache Kafka que es un proyecto de intermediación de mensajes de código abierto desarrollado por LinkedIn y donado a la Apache Software Foundation escrito en Java y Scala. Este servicio de Apache Kafka se ha diseñado para un uso genérico, es fácilmente escalable y se pueden crear diferentes tópicos para diferentes propósitos. Se creó un servicio sobre el despliegue del contenedor para ser accedido fuera del clúster.

En la ilustración 7 se pueden observar los dos contenedores desplegados en el clúster:

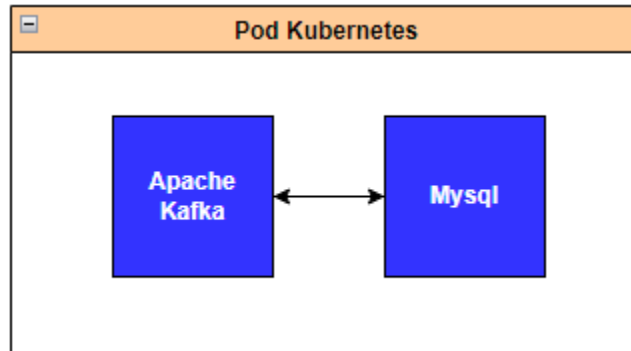


Ilustración 7 Pods dentro del clúster de Kubernetes

7.1.4 Integración de Ceph con Kubernetes

Considerando que algunos contenedores pueden requerir el uso de almacenamiento se ha tomado a bien provisionar el clúster de Kubernetes con el clúster de Ceph, para que los contenedores puedan contar con la garantía de que los datos generados por las aplicaciones (pods) cuentan con alta disponibilidad:

En la ilustración 8 se puede ver cómo el clúster de Kubernetes interactúa con el clúster de Ceph por medio del servicio de rados gateway configurado en Ceph.

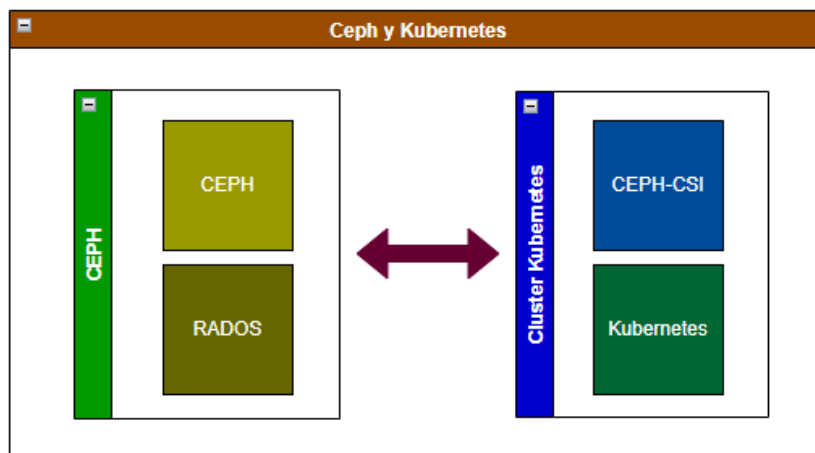


Ilustración 8 Integración entre Ceph y el clúster de Kubernetes

7.2 Análisis y Diseño

A este punto cabe recalcar que la solución brindada puede dividirse en tres partes: 1) contamos con OpenStack el cual permite hacer despliegues de máquinas virtuales y en general ofrecer servicios tipo IAAS, así como migrar sus aplicaciones tipo legacy hacia este entorno. 2) clúster de Kubernetes que van a permitir a la empresa contar con una plataforma moderna, escalable y segura para desplegar los contenedores encargados para la centralización de las transacciones. 3) El clúster de almacenamiento Ceph que es un componente transversal; tanto Openstack como Kubernetes hacen uso de él. La solución se ha diseñado de tal manera que todas las partes se complementan mutuamente.

7.3 Prototipo

Para demostrar la interacción entre las partes de la solución se ha creado una máquina virtual en OpenStack, la cual se encargará de producir un mensaje que se enviará al servicio de Apache Kafka, el cual se encuentra en uno de los contenedores del clúster de Kubernetes y además se encargará de consumir dicho mensaje (actuará como consumidor) y guardará dicho mensaje en la base de datos que se encuentra también en el contenedor de MySQL.

En la ilustración 9 se puede ver la interacción de los tres componentes más importantes dentro de la solución brindada:

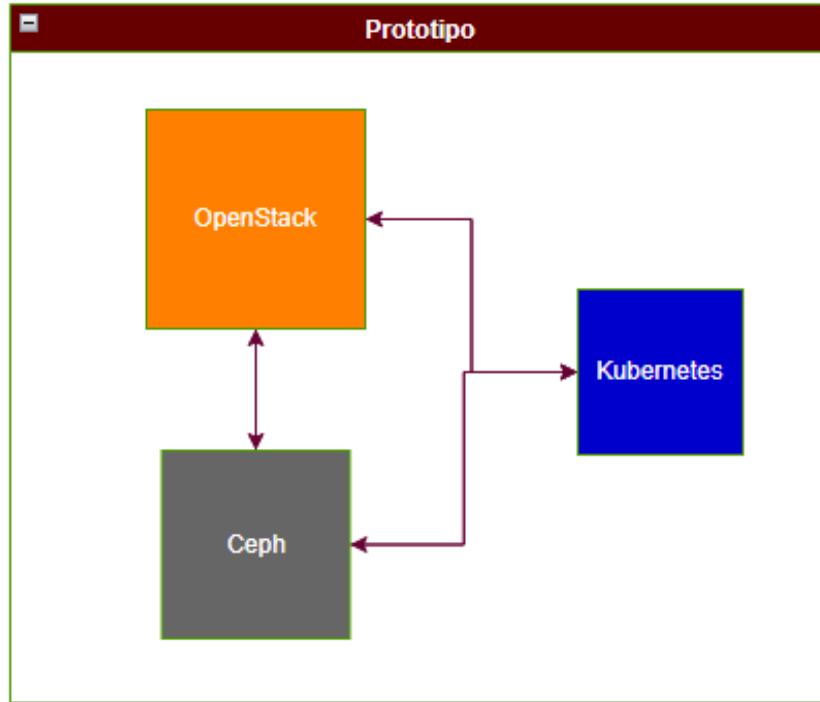


Ilustración 9 Prototipo

7.4 Software requerido



Ubuntu server

Ubuntu Server es una variante de Ubuntu que sale con cada versión y está dedicada especialmente para su uso en servidores. El uso de Ubuntu como servidor se ha extendido mucho en los últimos años, sobre todo para el uso de servidores web, de un modo tanto particular como profesional.



KVM

KVM (para máquina virtual basada en kernel) es una solución de virtualización completa para Linux en hardware x86 que contiene extensiones de virtualización (Intel VT o AMD-V). Consiste en un

módulo de kernel cargable, `kvm.ko`, que proporciona la infraestructura de virtualización central y un módulo específico del procesador, `kvm-intel.ko` o `kvm-amd.ko`.

Tabla 4 Software requerido

7.5 Hardware requerido

OpenStack							
Nombre	CPUs	RAM (GB)	Almacenamiento			SO	
Controller Node	4	8	20 GB	(Sistema Operativo)	Ubuntu 20.04	Server	
Compute Node 1	4	8	20 GB	(Sistema Operativo)	Ubuntu 20.04	Server	
Block Storage	1	1	10 GB	(Sistema Operativo)	Ubuntu 20.04	Server	
			200 GB (Para volúmenes)				
Ceph Admin	2	2	10 GB	(Para Sistema Operativo)	Ubuntu 20.04	Server	
Ceph Mon	1	1	10 GB	(Para Sistema Operativo)	Ubuntu 20.04	Server	
Ceph OSD 1	1	1	20 GB	(Para Sistema Operativo)	Ubuntu 20.04	Server	

			10GB	(Para			
			almacenamiento)				
Ceph OSD 2	1	1	20 GB	(Para Sistema	Ubuntu	Server	
			Operativo)		20.04		
			10GB	(Para			
			almacenamiento)				
Ceph OSD3	1	1	20 GB	(Para Sistema	Ubuntu	Server	
			Operativo)		20.04		
			10GB	(Para			
			almacenamiento)				
K8 Master	1	1	20 GB	(Sistema	Ubuntu	Server	
			Operativo)		20.04		
Worker1	1	1	20 GB	(Sistema	Ubuntu	Server	
			Operativo)		20.04		
Worker2	1	1	20 GB	(Sistema	Ubuntu	Server	
			Operativo)		20.04		

Tabla 5 Hardware requerido

7.6 Diagrama del prototipo

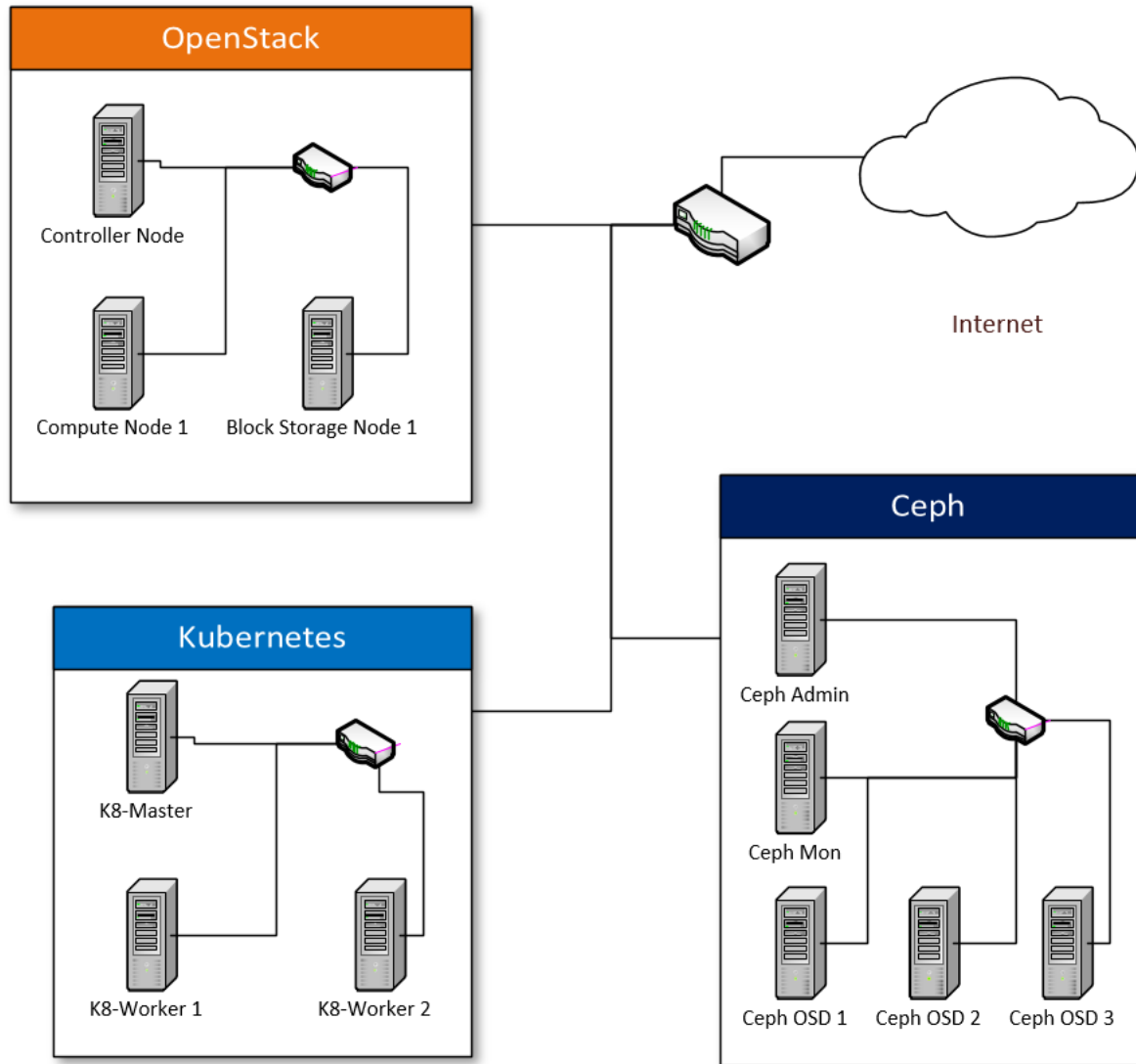


Ilustración 10 Diagrama del prototipo

7.7 Diagrama de redes

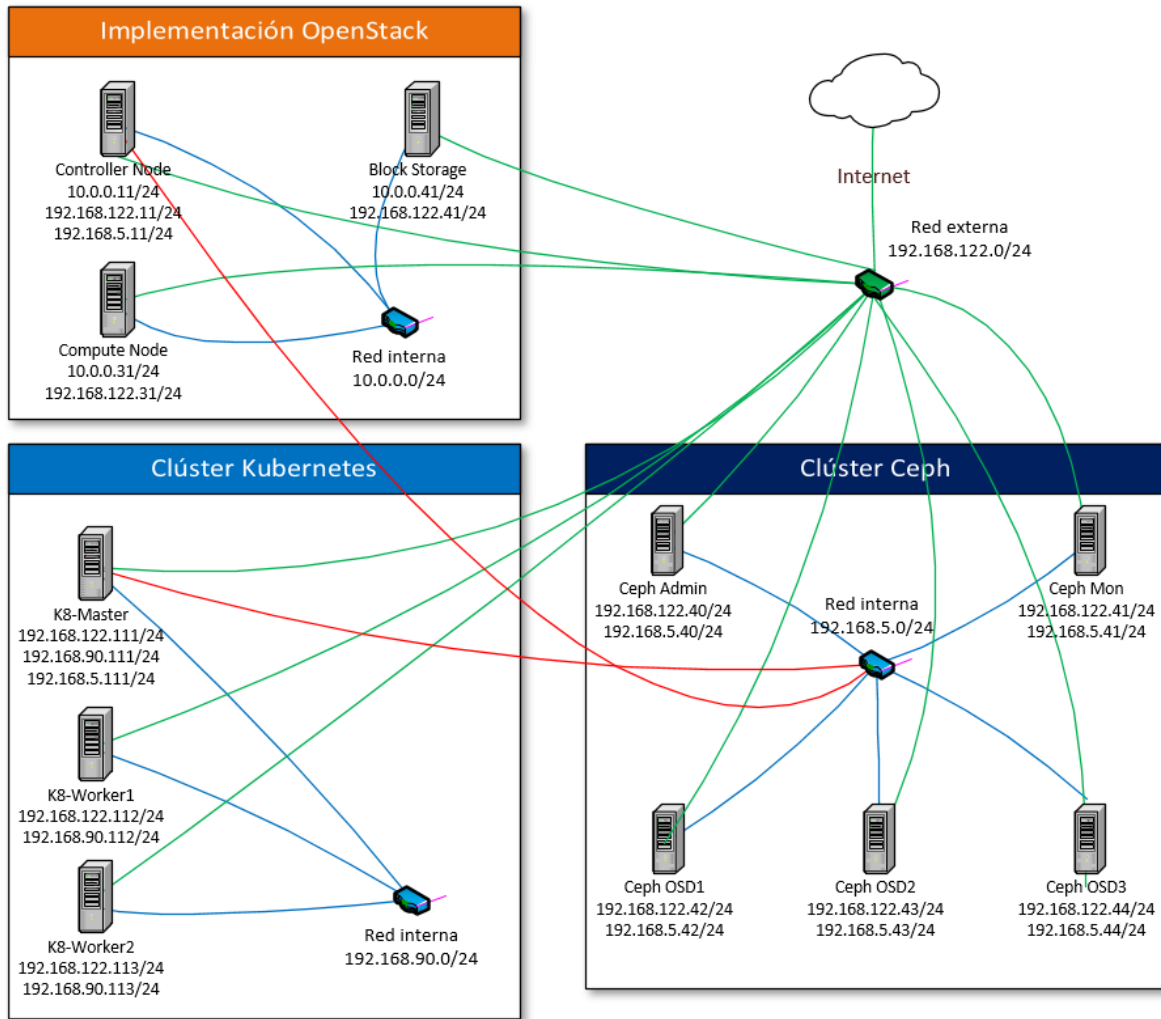


Ilustración 11 Diagrama de redes

8. Factibilidad

Para realizar la factibilidad del prototipo se realiza una comparación de costos entre los servicios de un proveedor de nube pública Google Cloud Platform y la implementación de una nube privada. Debido a que la implementación de la nube privada requiere una inversión inicial significativa, se determinará la cantidad de tiempo mínimo posibles (en el caso de existir) para que los costos de la implementación sobre la nube privada sean iguales o menores a los costos de pagar los servicios de la nube pública.

La elección de la plataforma Google Cloud Platform para realizar el análisis comparativo no obedece a una razón en particular, no obstante, las nubes públicas manejan precios muy estándares entre proveedores para los diferentes servicios que brindan.

8.1 Costos en Nube Pública Google Cloud Platform:

Los costos se han obtenido tomando en cuenta los recursos necesarios para cada servicio y usando la calculadora de precios ofrecida por la plataforma.

Se parte de que se requieren de los siguientes servicios para poder implementar la solución:

- Cloud Storage
- Google Cloud Pub-Sub
- Compute Engine
- Google Kubernetes Engine

Cloud Storage:



Cloud Storage	
1x Standard Storage	 
Location: Iowa	
Total Amount of Storage: 10,240 GiB	USD 204.80
Always Free usage included: No	
USD 204.80	

Ilustración 12 Precio de servicio de Cloud Storage Google Cloud Platform

Google Cloud Pubsub:



Pub/Sub	
	 
Message delivery type: Basic	
Volume: 1 GiB	USD 0.80
Subscriptions: 8	USD 9.11
Retained acknowledged subscriptions: 6	
Snapshots: 1	
Topic retention cost: 1.62	
Subscription retention Cost : 11.34	
Snapshot cost: 0.031	
USD 22.90	

Ilustración 13 Precio de servicio Pubsub Google Cloud Platform

Compute Engine:





Compute Engine	
8 x	  
Region: Iowa	
5,840 total hours per month	
Provisioning model: Regular	
Instance type: e2-standard-2	USD 391.35
Operating System / Software: Free	
Estimated Component Cost: USD 391.35 per 1 month	

Ilustración 14 Precio de servicio de Compute Engine Google Cloud Platform

Al Compute Engine que en este caso son 8 instancias, tenemos que adicionarles los discos de almacenamiento de cada máquina virtual:

Persistent Disk (Accompanying)	
8 x boot disk	
Product accompanying: Compute Engine	
Zonal balanced PD: 100 GiB	USD 10.00
USD 80.00	

Ilustración 15 Precio de disco persistente Google Cloud Platform

Por lo tanto, el costo mensual total por el servicio de compute Engine asciende a:
\$471.35

Google Kubernetes Engine:



GKE Autopilot	
3 x	 
Provisioning model: Regular	
Total vCPU Hours: 547.5	
Total Memory Hours: 2,190	
Total Storage Hours: 22,425,600	
Region: Iowa	
Estimated Component Cost: USD 1,264.07 per 1 month	

Ilustración 16 Precio de servicio Google Kubernetes Engine

Resumen de costo por servicio mensual:

Servicio	Costo Mensual
Cloud Storage	\$204.80
Google Cloud Pub-Sub	\$22.90
Compute Engine	\$471.35
Google Kubernetes Engine	\$1264.07
Total	\$1963.12

Tabla 6 Resumen de costo mensual por servicio

Costo Inicial: \$0

Costo Mensual: \$1963.12

Costo total Anual = Costo Inicial + (Costo Mensual*12)

Costo total Anual: \$23,557.4

8.2 Costos implementado nube privada

Para ello se plantea la creación de un Clúster utilizando 4 servidores, uno serviría como nodo controlador, otro como nodo de cómputo, uno como nodo almacenamiento en bloques y otro para el clúster de Kubernetes. Tomando en cuenta que la construcción desde cero de un Data Center supone una inversión bastante grande para una empresa, se ha optado por utilizar un Módulo Autocontenido, también conocidos como Micro Data Center, la ventaja de estos es que no requieren de un espacio espacial dentro de la empresa puesto que ya traen integrados todos los elementos necesarios para un Data Center y son construidos siguiendo las normas y estándares establecidos en esta área. Dicho esto, se describen los elementos principales y considerados para la implementación de la nube.

Equipo	Precio	Cantidad	Total
Easy Cube I9 Micro Data Center	\$10,000.00	1	\$10,000.00
Ethernet Switch	\$2900.00	1	\$2900.00
Network Security Firewall	\$2000.00	1	\$2000.00

Router	\$599.00	1	\$599.00
Servidor (Nodo Compute y Clúster Kubernetes)	\$5000.00	2	\$10,000.00
Servidor (Nodo Controlador y Nodo Block Storage)	\$2000.00	2	\$2000.00
Enterprise SSD (Nodo Block Storage)	\$569.99	3	\$1709.97
Total			\$31208.97

Tabla 7 Equipo necesario para implementación real del prototipo

El detalle y las especificaciones técnicas de los equipos se encuentran en los anexos en la sección 11.1 Equipos para la implementación en nube privada.

A este total se le agregaría la estimación de \$100 por mantenimiento realizado cada seis meses, es decir, que mensualmente se tendría un gasto adicional de \$16.67 y un gasto aproximado de \$100 dólares mensuales en electricidad.

Costo Inicial: \$31208.97

Costo Mensual: \$116.67

Costo total Anual = Costo Inicial + (Costo Mensual*12)

Costo total Anual: \$ 32609.01

Cabe mencionar que, para los costos de la nube privada, no se han tomado en cuenta aspectos relacionado por ejemplo a la inflación, personal especializado para la

instalación del centro de datos, espacio físico para el centro de datos, aire acondicionado, etc.

8.3 Comparación de costos

Proyectando los valores obtenidos para un periodo de cinco años de obtiene el siguiente resultado.

Proveedor	Primer Año	Segundo Año	Tercer Año	Cuarto Año	Quinto Año
Nube Pública	\$ 23557.4	\$ 23557.4	\$ 23557.4	\$ 23557.4	\$ 23557.4
Nube Privada	\$ 32609.01	\$ 1400.04	\$ 1400.04	\$ 1400.04	\$ 1400.04

Tabla 8 Comparación de costos entre nube publica y nube privada

De lo anterior, se puede observar que en un principio la nube privada supone una fuerte inversión en comparación con la inversión realizada en los proveedores de nubes públicas; sin embargo, también se puede apreciar que a partir del segundo año la cantidad de dinero invertida es mucho menor para la nube privada, lo quiere decir que a largo plazo esta solución se volvería rentable para la empresa Visión y Desarrollo.

El diseño planteado en el prototipo es escalable horizontalmente, es decir, se pueden agregar más recursos de hardware a OpenStack, el clúster de Kubernetes o del clúster de almacenamiento, según sean las necesidades.

Por lo tanto, hay una factibilidad para implementar la nube privada sobre la nube pública claramente ventajosa desde el punto de vista económico y a un plazo mayor a dos años.

9. Conclusiones y Recomendaciones

- Se aplicó una metodología de investigación e instrumentos para recolectar información de tipo cuantitativo y cualitativo, con los cuales se obtuvo información del modelo de negocio de la organización.
- Se ha logrado analizar la problemática actual de la organización y desarrollado una solución que cumple con los requisitos planteados por la organización.
- Se logró definir un diseño flexible, seguro y escalable para que la organización pueda comenzar a hacer una migración gradual hacia la nube.
- Se completo el desarrollo de un prototipo de infraestructura en la nube que cumple con las necesidades de la organización para la centralización de las transacciones.
- Considerar el uso del clúster de Kubernetes para nuevos requerimientos y proyectos dentro de la Organización.

Respondiendo a la pregunta de investigación: Se determinó que la centralización de transacciones realizadas en los diferentes restaurantes a los cuales la empresa Visión y Desarrollo brinda sus servicios se puede llevar a cabo con la implementación de una nube privada con OpenStack y un clúster de Kubernetes garantizando alta concurrencia, escalabilidad, disponibilidad y flexibilidad.

10. Referencias

- Bastis Consultores. (2 de Marzo de 2021). *Enfoque de la Investigación*. Obtenido de <https://online-tesis.com/enfoque-de-la-investigacion/>
- Corbetta, P. (2007). *Metodología y Técnicas de Investigación Social*.
- Hernandez Quintero, N. L., & Florez Fuente, A. S. (Diciembre de 2014). *Computación en la nube*. Obtenido de <https://www.fesc.edu.co/Revistas/OJS/index.php/mundofesc/article/view/48/90>
- Kubernetes*. (2022). Obtenido de Orquestación de contenedores para producción: <https://kubernetes.io/es/>
- Metodología Cuantitativa*. (s.f.). Obtenido de <https://acortar.link/fVcvWH>
- Murillo Torrecilla, J. (s.f.). *La Entrevista*. Obtenido de http://www2.uca.edu.sv/mcp/media/archivo/f53e86_entrevistapdfcopy.pdf?f
- Red Hat*. (19 de Abril de 2018). Obtenido de El concepto de DevOps: <https://www.redhat.com/es/topics/devops>
- Red Hat*. (09 de Enero de 2018). Obtenido de ¿Qué es Docker?: <https://www.redhat.com/es/topics/containers/what-is-docker>

11. Anexos

Anexo 1. Introducción al diseño de bajo nivel

El presente documento detalla de forma ordenada los requerimientos y pasos a seguir para el montaje de una infraestructura en la nube, la cual consta de un despliegue de OpenStack, un clúster Ceph y un clúster de Kubernetes trabajando en conjunto.

Este prototipo está diseñado para solventar necesidades de la empresa Visión y Desarrollo, la cual se encarga de proveer software especializado para restaurantes. El prototipo brindará las herramientas necesarias para centralizar la información de los restaurantes y dará la posibilidad de desplegar nuevas aplicaciones para los mismos.

Anexo 2. Objetivos del diseño de bajo nivel

2.1 Objetivo general



- Desplegar el prototipo funcional que permite la centralización de datos y el despliegue de aplicaciones a través de la implementación de OpenStack, Ceph y Kubernetes.

2.2 Objetivos específicos

- Conocer los requisitos de hardware y software mínimos para el despliegue del prototipo.
- Desplegar de forma correcta una infraestructura de OpenStack.
- Desplegar de forma correcta un clúster de almacenamiento Ceph.
- Desplegar de forma correcta un clúster de Kubernetes.
- Integrar OpenStack y Ceph para el almacenamiento de objetos.
- Integrar el clúster de Kubernetes con Ceph para la creación de volúmenes persistentes.
- Lanzar una instancia funcional en OpenStack
- Desplegar una aplicación funcional en Kubernetes.

Anexo 3. Requerimientos técnicos del prototipo

3.1 Requerimientos de software

 <p>Ubuntu server</p>	<p>Sistema operativo Ubuntu Server versión 22.04</p>
 <p>KVM</p>	<p>Gestor de máquinas virtuales KVM (Solo para entorno de prueba)</p>

3.2 Requerimientos de hardware

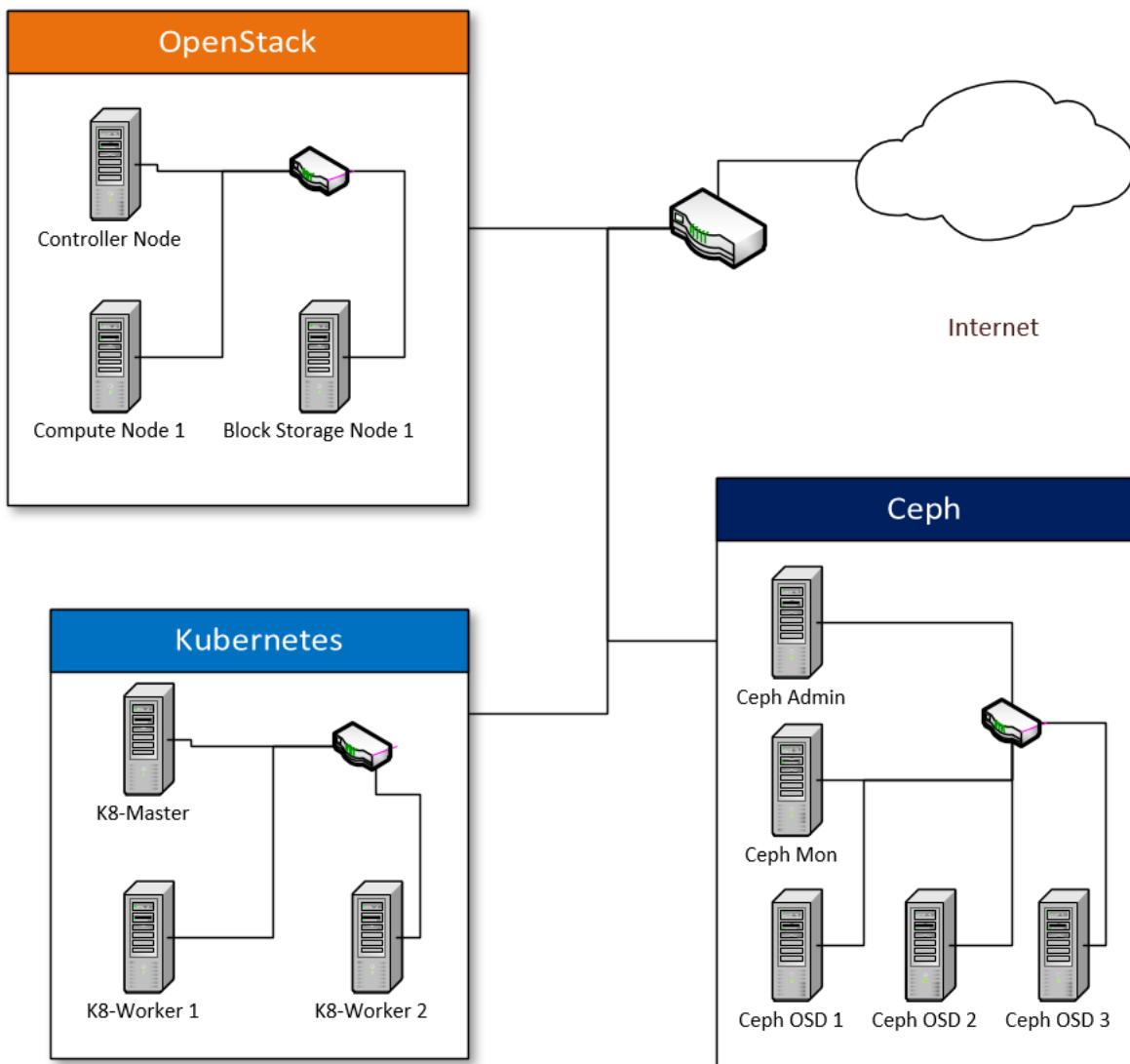
OpenStack				
Nombre	CPUs	RAM (GB)	Almacenamiento	SO
Controller Node	4	8	20 GB (Sistema Operativo)	Ubuntu Server 20.04
Compute Node 1	4	8	20 GB (Sistema Operativo)	Ubuntu Server 20.04
Block Storage	1	1	10 GB (Sistema Operativo) 200 GB (Para volúmenes)	Ubuntu Server 20.04

Clúster Ceph				
Nombre	CPUs	RAM (GB)	Almacenamiento	SO
Ceph Admin	2	2	20 GB (Para Sistema Operativo)	Ubuntu Server 20.04
Ceph Mon	1	1	20 GB (Para Sistema Operativo)	Ubuntu Server 20.04
Ceph OSD 1	1	1	20 GB (Para Sistema Operativo) 10GB (Para almacenamiento)	Ubuntu Server 20.04

Ceph OSD 2	1	1	20 GB (Para Sistema Operativo) 10GB (Para almacenamiento)	Ubuntu Server 20.04
Ceph OSD3	1	1	10 GB (Para Sistema Operativo) 10GB (Para almacenamiento)	Ubuntu Server 20.04

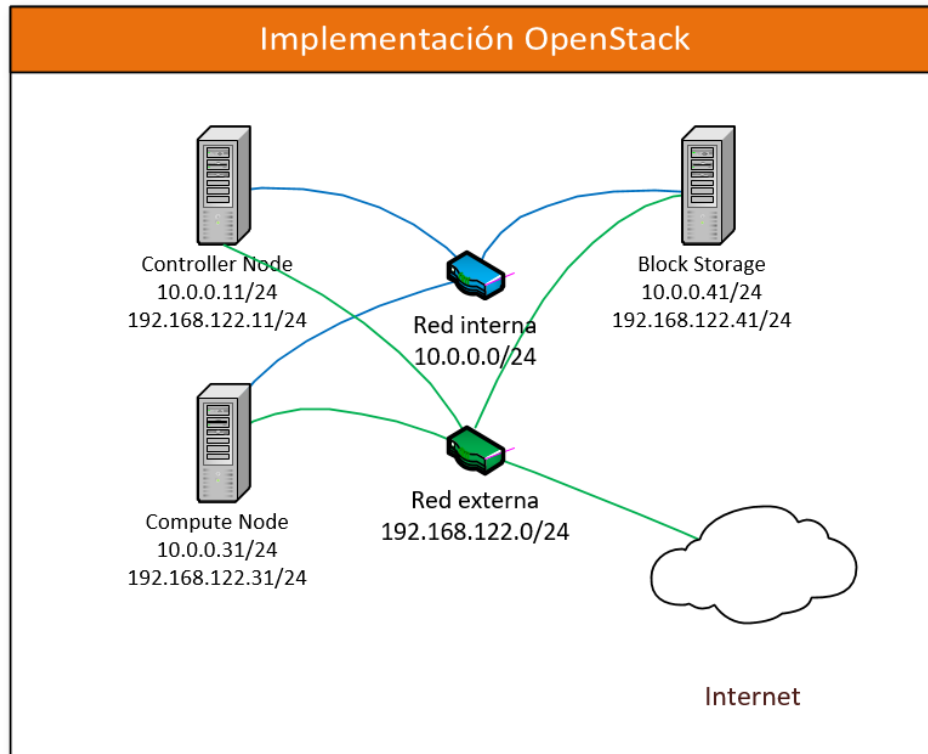
Clúster Kubernetes				
K8 Master	1	1	20 GB (Sistema Operativo)	Ubuntu Server 20.04
Worker1	1	1	20 GB (Sistema Operativo)	Ubuntu Server 20.04
Worker2	1	1	20 GB (Sistema Operativo)	Ubuntu Server 20.04

Anexo 4. Diagrama del sistema



Anexo 6. Despliegue de OpenStack

Para comenzar con la configuración del prototipo es necesario crear las máquinas virtuales con las mismas características del equipo listado en los requerimientos. Dichas maquinas deben tener instalado el sistema operativo Ubuntu 20.04



Para el despliegue de OpenStack utilizaremos la guía de implementación oficial presente en el sitio web del software.

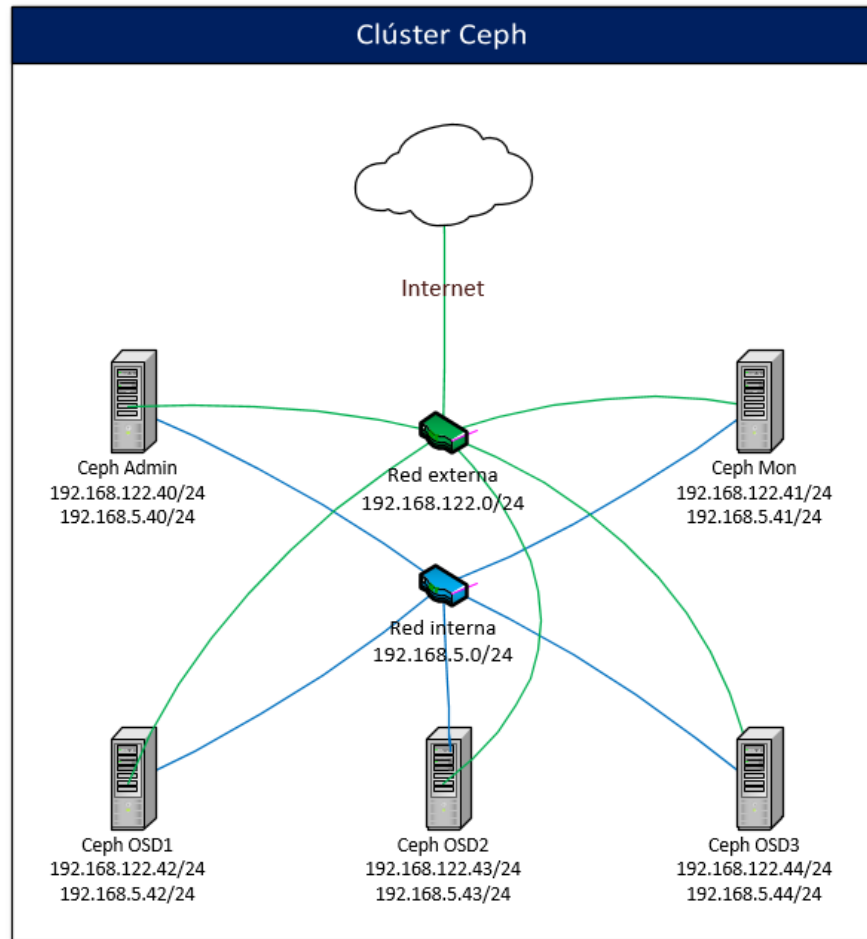
OpenStack Installation Guide, OpenStack de <https://docs.openstack.org/install-guide/>

Se deben desarrollar los siguientes pasos aplicando las IP reflejadas en el diagrama.

- Environment
 - Security
 - Host networking
 - Controller node
 - Compute node
 - Block storage node
 - Verify connectivity
 - Network Time Protocol (NTP)

- Controller node
 - Other nodes
 - Verify operation
- OpenStack packages
 - OpenStack packages for Ubuntu
- SQL database
 - SQL database for Ubuntu
- Message queue
 - Message queue for Ubuntu
- Memcached
 - Memcached for Ubuntu
- Etcd
 - Etcd for Ubuntu
- Install OpenStack services
 - Minimal deployment for Yoga

Anexo 7. Despliegue de clúster Ceph



Para la instalación de Ceph nos basaremos en la documentación oficial:

INSTALLING CEPH, Ceph de <https://docs.ceph.com/en/latest/install/#recommended-methods>

1. Configurar archivo `/etc/hosts` en todos los nodos

```
192.168.5.40 ceph-admin
192.168.5.41 ceph-mon
192.168.5.42 ceph-osd1
192.168.5.43 ceph-osd2
192.168.5.44 ceph-osd3
```

2. Instalación y activación de ssh y chrony en todos los nodos

```
apt install chrony openssh-server curl
systemctl enable --now chrony
systemctl enable --now sshd
```

3. Instalación de Docker en todos los nodos

```
apt install apt-transport-https ca-certificates curl gnupg-agent
software-properties-common -y
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo
apt-key add -
```

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

```
apt update
apt install docker-ce docker-ce-cli containerd.io -y
systemctl enable --now docker
```

4. Crear usuario admin para desplegar el clúster nodo ceph-admin

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

```
passwd cephadmin
```

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

5. Permitir el acceso al usuario root en el servicio ssh en los nodos mon y osds

```
nano /etc/ssh/sshd_config
```

```
PermitRootLogin yes
```

```
/etc/init.d/ssh restart
```

6. Establecer contraseña root

```
passwd
```

7. Instalar cephadm en el nodo ceph-admin

```
sudo su
```

```
sudo su  
wget -q  
https://github.com/ceph/ceph/raw/pacific/src/cephadm/cephadm -P  
/usr/bin/
```

```
chmod +x /usr/bin/cephadm
```

8. Desplegar servicio

```
sudo cephadm bootstrap --mon-ip 192.168.5.40
```

9. Se devolverá las credenciales del dashboard de ceph

Ceph Dashboard is now available at:

URL: <https://ceph-admin:8443/>

User: admin

Password: 8964vjghsi

Enabling client.admin keyring and conf on hosts with "admin" label

Enabling autotune for osd_memory_target

You can access the Ceph CLI with:

```
sudo /usr/bin/cephadm shell --fsid
```

```
f959b65e-91c2-11ec-9776-abbffb8a52a1 -c /etc/ceph/ceph.conf -k
```

10. Activar Ceph CLI

```
sudo /usr/bin/cephadm shell --fsid
```

```
f959b65e-91c2-11ec-9776-abbffb8a52a1 -c /etc/ceph/ceph.conf -k
```

```
/etc/ceph/ceph.client.admin.keyring
```

11. Instalar ceph-common

```
sudo cephadm add-repo --release quincy
```

```
sudo cephadm install ceph-common
```

```
sudo ceph -s
```

12. Listar componentes del clúster en ceph-admin

```
sudo ceph orch host ls
```

HOST	ADDR	LABELS	STATUS
ceph-admin	192.168.5.40	_admin	

13. Registrando nodo monitor desde ceph-admin

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

```
sudo ceph orch host add ceph-mon
```

```
Added host 'ceph-mon' with addr '192.168.5.41'
```

```
sudo ceph orch host label add ceph-mon mon/osd
```

14. Registrar los OSD

```
for i in ceph-osd1 ceph-osd2 ceph-osd3; do sudo ssh-copy-id -f -i /etc/ceph/ceph.pub root@$i; done
```

```
sudo ceph orch host add ceph-osd1
```

```
sudo ceph orch host add ceph-osd2
```

```
sudo ceph orch host add ceph-osd3
```

```
for i in ceph-osd1 ceph-osd2 ceph-osd3; do sudo ceph orch host label add $i osd; done
```


15. Preparar discos de los nodos OSD (Todos los OSD)

```
vgcreate vg01 /dev/vdb  
lvcreate -L 10G -n lv01 vg01
```

16. Agregar volúmenes al clúster

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

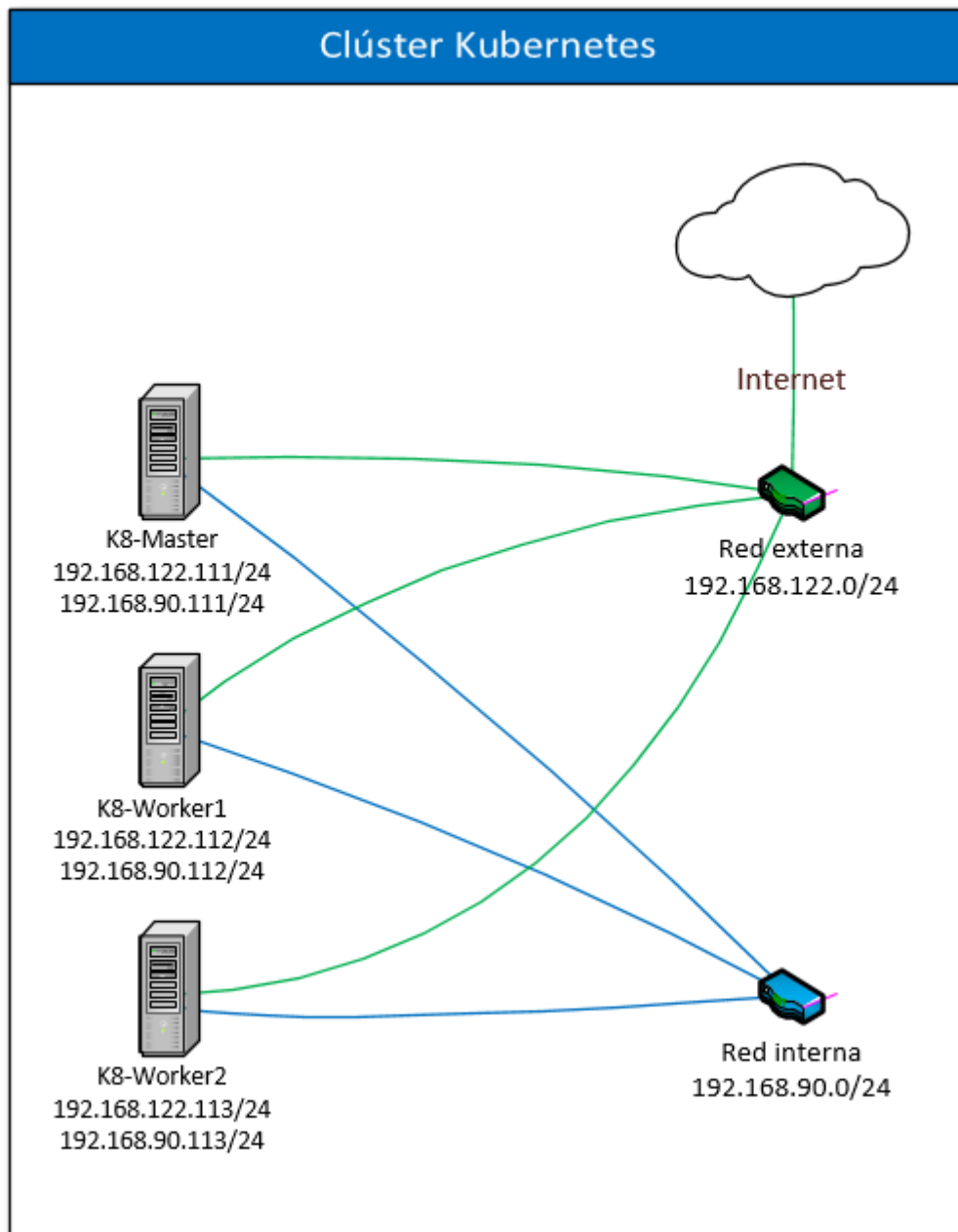
17. Verificar estado del clúster

```
sudo ceph -s
```

Debe retornar lo siguiente

```
cluster:
  id:          180727f4-d7bd-11ec-87e9-ad4e36c6f01b
  health:      HEALTH_WARN
               mon ceph-admin is low on available space
               3 osds down
               3 hosts (3 osds) down
               1 root (3 osds) down
  services:
    mon: 5 daemons, quorum
    ceph-admin,ceph-mon,ceph-osd1,ceph-osd3,ceph-osd4 (age 3m)
    mgr: ceph-mon.xyiubq(active, since 4h), standbys:
    ceph-admin.zzvwpk
    osd: 3 osds: 0 up (since 3m), 3 in (since 3h)
  data:
    pools:     1 pools, 1 pgs
  objects:     2 objects, 449 KiB
  usage:       61 MiB used, 57 GiB / 57 GiB avail
```

Anexo 8. Despliegue de clúster Kubernetes



Para la instalación del clúster de Kubernetes nos basaremos en la documentación oficial:

Kubernetes Documentation, Kubernetes de <https://kubernetes.io/docs/home/>

1. Instalar chrony en todos los nodos

```
sudo apt install chrony
```

2. Configurar servidor NPT en nodo Master

```
sudo nano /etc/chrony/chrony.conf
```

```
server time.google.com iburst
```

3. Configurar servidor NPT en nodos Worker

```
sudo nano /etc/chrony/chrony.conf
```

```
server k8-master iburst
```

4. Modificar archivo /etc/hosts en todos los nodos

```
192.168.90.111 k8-master  
192.168.90.112 k8-worker1  
192.168.90.113 k8-worker2
```

5. Deshabilitando swap y configurando aspectos de kernel en todos los nodos

```
sudo swapoff -a  
sudo sed -i ' / swap / s/^\(.*\)$/#\1/g' /etc/fstab
```

6. Habilitando módulos de kernel en todos los nodos

```
sudo tee /etc/modules-load.d/containerd.conf <<EOF
overlay
br_netfilter
EOF

sudo modprobe overlay
sudo modprobe br_netfilter
```

7. Configurando en los archivos de módulos, en todos los nodos

```
sudo tee /etc/sysctl.d/kubernetes.conf <<EOF
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
EOF
```

8. Poniéndolos en ejecución, en todos los nodos

```
sudo sysctl --system
```

9. Instalación del container runtime

```
sudo apt install -y curl gnupg2 software-properties-common
apt-transport-https ca-certificates
```

10. Activar el repositorio de Docker

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo
apt-key add -

sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"

sudo apt update

sudo apt install docker containerd.io
```

11. Configuración de containerd

12. Reinicio de servicios y activación del servicio de containerd, en todos los nodos.

```
containerd config default | sudo tee /etc/containerd/config.toml
>/dev/null 2>&1
sudo sed -i 's/SystemdCgroup \= false/SystemdCgroup \= true/g'
/etc/containerd/config.toml
```

```
sudo systemctl restart containerd

sudo systemctl enable containerd
```

13. Configuración de Kubernetes a través del repositorio

```
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg |
sudo apt-key add -

sudo apt-add-repository "deb http://apt.kubernetes.io/
kubernetes-xenial main"
```

14. Instalación de Kubernetes, así como los componentes kubectl, kubelet y kubeadm

```
sudo apt update
```

```
sudo apt install -y kubelet kubeadm kubectl
```

```
sudo apt-mark hold kubelet kubeadm kubectl
```

15. Inicializando el clúster de Kubernetes

```
sudo kubeadm init --control-plane-endpoint=k8-master
```

Debería de darle una salida, como la que se le presenta a continuación:

```
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a
rotatable kubelet client certificate and key
```

```
[addons] Applied essential addon: CoreDNS
```

```
[addons] Applied essential addon: kube-proxy
```

Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

```
mkdir -p $HOME/.kube
```

```
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
```

```
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Alternatively, if you are the root user, you can run:

```
export KUBECONFIG=/etc/kubernetes/admin.conf
```

You should now deploy a pod network to the cluster.

Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:

```
https://kubernetes.io/docs/concepts/cluster-administration/addons/
```

You can now join any number of control-plane nodes by copying certificate Authorities and service account keys on each node and then running the following as root:

```
kubeadm join k8-master:6443 --token mwk1fd.bwhhfdx3l3pxywk \
--discovery-token-ca-cert-hash
sha256:be38c6a1d651632867df5b816a31b71e95a51eac4d446371b518143
7dadaf77b \
--control-plane
```

Then you can join any number of worker nodes by running the following on each

as root:

16. Ejecutar en nodo Master

```
mkdir -p $HOME/.kube  
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config  
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

17. Verificar clúster

```
kubectl cluster-info  
kubectl get nodes
```

NAME	STATUS	ROLES	AGE
VERSION			

18. Instalar calico

```
curl https://projectcalico.docs.tigera.io/manifests/calico.yaml  
-O  
  
kubectl apply -f calico.yaml
```

19. Incluir workers al clúster (Ejecutar en nodos worker)

```
sudo kubeadm join k8-master:6443 --token mwk1fd.bwhhfdx3l3pxykw  
--discovery-token-ca-cert-hash  
sha256:be38c6a1d651632867df5b816a31b71e95a51eac4d446371b518143  
7dada
```

Nota: Debe utilizarse los valores retornados en el paso 15

Anexo 9. Integración de OpenStack y Ceph

Para manejar el Object Storage de OpenStack lo haremos mediante el almacenamiento en el clúster de Ceph

1. Agregamos la red 192.168.5.0/24 al nodo Controller de OpenStack
2. Instalamos Apache en el nodo Ceph-Admin

```
sudo apt install apache2
```

3. Creamos o editamos el archivo `/etc/apache2/conf-available/rgw.conf`

```
<VirtualHost *:80>
ServerName localhost
DocumentRoot /var/www/html

ErrorLog /var/log/apache2/rgw_error.log
CustomLog /var/log/apache2/rgw_access.log combined

# LogLevel debug

RewriteEngine On

RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization},L]

SetEnv proxy-nokeepalive 1

ProxyPass / fcgi://localhost:9000/
```

4. Editamos el archivo /etc/ceph/ceph.conf

```
[client.radosgw.gateway]

host = ceph-admin
keyring = /etc/ceph/ceph.client.radosgw.keyring
rgw socket path = /var/run/ceph/ceph.radosgw.gateway.fastcgi.sock
log file = /var/log/ceph/client.radosgw.gateway.log
rgw print continue = false
rgw enable usage log = true
rgw usage log tick interval = 30
rgw usage log flush threshold = 1024
rgw usage max shards = 32
rgw usage max user shards = 1
rgw frontends = "beast port=81"
rgw keystone api version = 3
rgw keystone url = http://192.168.122.11:5000
rgw keystone accepted roles = admin
rgw keystone token cache size = 10
rgw keystone implicit tenants = false
rgw keystone admin user = admin
rgw keystone admin password = admin
rgw keystone admin domain = default
rgw keystone admin project = admin
```

5. Generamos la llave de autenticación de rados para autenticarse en el clúster

```
ceph-authtool -C -n client.radosgw.gateway --gen-key  
/etc/ceph/keyring.radosgw.gateway
```

```
ceph-authtool -n client.radosgw.gateway --cap mon 'allow rw' --cap osd  
'allow rwx' /etc/ceph/keyring.radosgw.gateway
```

6. Agregamos la llave a las entidades de autenticación

```
ceph auth add client.radosgw.gateway --in-file=keyring.radosgw.gateway
```

7. Iniciamos los servicios de rados y apache

```
sudo /etc/init.d/apache2 start  
sudo /etc/init.d/radosgw start
```

8. Ejecutamos los siguientes comandos en el nodo controller de OpenStack

```
openstack endpoint create --region RegionOne object-store public  
http://ceph-admin:81/swift/v1/AUTH_%(project_id)s  
  
openstack endpoint create --region RegionOne object-store admin  
http://ceph-admin:81/swift/v1/AUTH_%(project_id)s  
  
openstack endpoint create --region RegionOne object-store internal  
http://ceph-admin:81/swift/v1/AUTH %(project id)s
```

Anexo 10. Integración del clúster de Kubernetes con Ceph

Para manejar volúmenes persistentes dentro del clúster de Kubernetes, utilizaremos una integración con el clúster Ceph.

Para esto desarrollamos los pasos de la guía oficial de Ceph para su implementación con Kubernetes

CONFIGURE CEPH-CSI, Ceph de <https://docs.ceph.com/en/latest/rbd/rbd-kubernetes/>

Anexo 11. Instalación de servicios en clúster Kubernetes

Para desplegar los servicios de nuestra aplicación de prueba es necesario ejecutar los siguientes archivos mediante la siguiente forma

00-namespace.yaml

```
apiVersion: v1
kind: Namespace
metadata:
  name: "kafka"
labels:
  name: "kafka"
```

Kubectl apply -f 00-namespace.yaml

01-zookeeper.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: zookeeper
  name: zookeeper
  namespace: kafka
spec:
  replicas: 1
  selector:
    matchLabels:
      app: zookeeper
  template:
    metadata:
      labels:
        app: zookeeper
    spec:
      containers:
        - image: wurstmeister/zookeeper
          imagePullPolicy: IfNotPresent
          name: zookeeper
          ports:
            - containerPort: 2181
```

Kubectl apply -f 01-zookeeper.yaml

02-kafka.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: kafka-broker
  name: kafka-broker
  namespace: kafka
spec:
  replicas: 1
  selector:
    matchLabels:
      app: kafka-broker
  template:
    metadata:
      labels:
        app: kafka-broker
    spec:
      hostname: kafka-broker
      containers:
        - env:
            - name: KAFKA_BROKER_ID
              value: "1"
            - name: KAFKA_ZOOKEEPER_CONNECT
              value: 10.98.189.30:2181
            - name: KAFKA_LISTENER_SECURITY_PROTOCOL_MAP
              value: "INTERNAL_PLAINTEXT:PLAINTEXT,EXTERNAL_PLAINTEXT:PLAINTEXT"
            - name: KAFKA_ADVERTISED_LISTENERS
              value: "INTERNAL_PLAINTEXT://:9092,EXTERNAL_PLAINTEXT://192.168.122.191:9093"
            - name: KAFKA_LISTENERS
              value: "INTERNAL_PLAINTEXT://0.0.0.0:9092,EXTERNAL_PLAINTEXT://0.0.0.0:9093"
            - name: KAFKA_INTER_BROKER_LISTENER_NAME
              value: "INTERNAL_PLAINTEXT"
          image: wurstmeister/kafka
          imagePullPolicy: IfNotPresent
          name: kafka-broker
          ports:
            - containerPort: 9092
            - containerPort: 9093
```

kubectl apply -f 02-kafka.yaml

03-kafka-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: kafka-broker
  name: kafka-broker
  namespace: kafka
  resourceVersion: "170410"
  uid: 19516ae1-49b2-4804-a64a-e1ab931f7574
spec:
  clusterIP: 10.102.45.207
  clusterIPs:
    - 10.102.45.207
  externalTrafficPolicy: Clúster
  internalTrafficPolicy: Clúster
  ipFamilies:
    - IPv4
  ipFamilyPolicy: SingleStack
  ports:
    - nodePort: 30883
      port: 9093
      protocol: TCP
      targetPort: 9093
  selector:
    app: kafka-broker
  sessionAffinity: None
  type: NodePort
  externalIPs:
    - 192.168.122.191
status:
  loadBalancer: {}
```

kubectl apply -f 03-kafka-service.yaml

00-pvc.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mysql-pvc
spec:
  accessModes:
    - ReadWriteMany
  volumeMode: Block
  resources:
    requests:
      storage: 2Gi
  storageClassName: csi-rbd-sc
```

kubectl apply -f 00-pvc.yaml

01-mysql-deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mysql
spec:
  selector:
    matchLabels:
      app: mysql
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
        - image: mysql:5.6
          name: mysql
          env:
            # Use secret in real usage
            - name: MYSQL_ROOT_PASSWORD
              value: icc115
          ports:
            - containerPort: 3306
              name: mysql
          volumeDevices:
            - name: mysql-persistent-storage
              devicePath: /var/lib/mysql
      volumes:
        - name: mysql-persistent-storage
          persistentVolumeClaim:
            claimName: mysql-pvc
          readOnly: false
```

kubectl apply -f 01-mysql-deployment.yaml

02-mysql-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: mysql
  namespace: default
  resourceVersion: "158955"
  uid: 83f80016-7ffd-4132-b42d-7e3af59e3da5
spec:
  clusterIP: 10.109.47.246
  clusterIPs:
  - 10.109.47.246
  externalTrafficPolicy: Cluster
  internalTrafficPolicy: Cluster
  ipFamilies:
  - IPv4
  ipFamilyPolicy: SingleStack
  ports:
  - nodePort: 30852
    port: 3306
    protocol: TCP
    targetPort: 3306
  selector:
    app: mysql
  sessionAffinity: None
  type: NodePort
status:
  loadBalancer: {}
```

kubectl apply -f 02-mysql-service.yaml

Anexo 12. Instalación de aplicación en instancia OpenStack

Para instalar la aplicación de prueba necesitamos disponer de una instancia del sistema operativo Ubuntu ejecutándose en OpenStack

Para ello puede apoyarse de la documentación oficial

Launch an instance, OpenStack de <https://docs.openstack.org/install-guide/launch-instance.html>

Con la instancia corriendo debe instalar Docker

Install Docker Engine on Ubuntu, Docker de <https://docs.docker.com/engine/install/ubuntu/>

Ahora puede correr la aplicación ejecutando los siguientes comandos:

```
docker run -d -p3000:80 mc11057/kafka-client:1.0
docker run -d -p8080:8080 mc11057/kafka-server:1.0
```

Anexo 13. Equipos para la implementación en nube privada

- Easy Cube I9 Micro Data Center
<http://en.eastups.com/u/cms/en/202005/26110912hbcj.pdf>
- Ethernet Switch <https://centec2021.en.made-in-china.com/product/oKiQxBhUOEaX/China-48-port-10GE-40GE-Network-Switch-Ethernet-Switch.html>
- Network Security Firewall
<https://www.cisco.com/c/en/us/support/security/asa-5516-x-firepower-services/model.html>
- Router <https://www.cisco.com/c/en/us/support/routers/4331-integrated-services-router-isr/model.html>
- Servidor (Nodo Compute) y Clúster de Kubernetes
<https://ark.intel.com/content/www/es/es/ark/products/199345/intel-xeon-gold-6238r-processor-38-5m-cache-2-20-ghz.html>
- Servidor (Nodo Controller y Block Storage)
<https://ark.intel.com/content/www/es/es/ark/products/197101/intel-xeon-bronze-3206r-processor-11m-cache-1-90-ghz.html>
- Enterprise SSD (Nodo Block Storage)
<https://semiconductor.samsung.com/ssd/datacenter-ssd/pm9a3/>

Anexo 14. Encuesta para trabajadores administrativos de restaurantes

