

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



**“Estrategia de migración desde entorno
OpenShift hacia Oracle Kubernetes Engine”**

FELIPE ANDRÉS BASSO VERA

TESINA PARA OPTAR AL GRADO
DE MAGÍSTER EN TECNOLOGÍAS DE
LA INFORMACIÓN

Profesor Guía: RAÚL MONGE



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☐ Memoria o trabajo de título ☒ Tesis de Postgrado

Título del trabajo: Estrategia de migración desde entorno OpenShift hacia Oracle Kubernetes Engine

Nombre del candidato(a): Felipe Andrés Basso Vera

Carrera / Grado: Magíster en Tecnologías de la Información.

Campus: San Joaquín **Departamento:** informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Raúl Monge, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (**marcar una opción**):

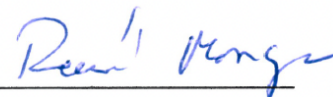
☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

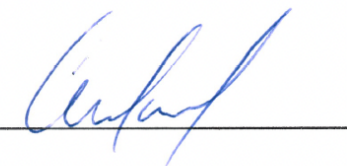
Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 16-10-2026

Firma: 

Estudiante o Candidato(a):

Fecha: 16-10-2026

Firma: 

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.



Estrategia de migración desde entorno OpenShift hacia Oracle Kubernetes Engine

Felipe Andrés Basso Vera

Sistemas Oracle Chile
Avenida Vitacura 2939, Piso 6, Las Condes, Santiago

Felipe.basso.vera@gmail.com

Resumen: La adopción de plataformas basadas en microservicios y contenedores ha permitido a las organizaciones acelerar el ciclo de entrega de software, mejorar el time to market, estandarizar entornos y mejorar la escalabilidad de las aplicaciones. Dentro de estas plataformas, OpenShift se ha consolidado como una distribución empresarial de Kubernetes con un sistema operativo optimizado, seguridad, facilidad para despliegue y operación. No obstante, en la actualidad muchas organizaciones, tal como ocurre en la industria FinOps, están apostando a priorizar escenarios de optimización de costos, reducción de dependencia tecnológica y automatización de infraestructura, con la finalidad de facilitar la adopción de servicios administrados. En base a lo anterior, diversas organizaciones evalúan migrar sus cargas de trabajo hacia soluciones de Kubernetes administrado en la nube.

Esta tesina desarrolla, formaliza y valida una estrategia estructurada para migrar cargas de trabajo desde OpenShift hacia el servicio cloud Oracle Kubernetes Engine (OKE). La estrategia fue aplicada y evaluada mediante un caso real, comparando aspectos técnicos, operacionales y económicos entre ambas plataformas. Los resultados obtenidos evidencian que una migración basada en un proceso metodológico permite mantener el rendimiento de las aplicaciones, mejorar determinados aspectos operacionales y reducir el costo total de propiedad en un horizonte de mediano plazo. El principal aporte del trabajo es proponer una estrategia reproducible que puede servir como guía para organizaciones que enfrenten procesos de modernización de plataformas Kubernetes y migración hacia servicios administrados en la nube.

Palabras Clave: Oracle Kubernetes Engine (OKE), OpenShift, Automatización de infraestructura, FinOps.

1 Introducción

La modernización de plataformas tecnológicas basadas en contenedores se ha convertido en un desafío estratégico para las organizaciones que buscan reducir costos, aumentar la flexibilidad y disminuir la dependencia de soluciones propietarias. En este contexto, la migración desde Red Hat OpenShift [1] hacia un servicio Kubernetes [2] administrado o Engine¹ nativo en *cloud* representa una alternativa que permite aprovechar un entorno Kubernetes administrado, manteniendo las capacidades de escalabilidad, disponibilidad y automatización requeridas por aplicaciones empresariales.

Sin embargo, este proceso implica desafíos técnicos y operacionales que pueden afectar la continuidad del servicio y el desempeño de las aplicaciones, si no se ejecuta de manera planificada. Por ello, la presente investigación propone una estrategia estructurada para la migración de cargas de trabajo desde OpenShift hacia Oracle Kubernetes Engine (OKE) [3], integrando etapas de evaluación, preparación, migración, validación y optimización.

¹ Kubernetes Engine: Entorno gestionado en la nube que automatiza la implementación, el escalado y la gestión de aplicaciones en contenedores

Finalmente, mediante la ejecución de pruebas comparativas de rendimiento, escalabilidad y resiliencia, junto con un análisis económico del costo total de propiedad, se busca demostrar que una migración ordenada permite mantener o incluso mejorar algunos aspectos de la administración y desempeño de las aplicaciones y *clusters*, obteniendo beneficios operacionales y financieros que justifican la inversión inicial en un horizonte de tres años.

Es importante señalar que, si bien esta tesina se enfoca en la migración hacia OKE, la misma estrategia y etapas son homologables para cualquier proveedor *cloud*, teniendo en consideración las dependencias propias de cada uno.

1.1 Motivación

La modernización de plataformas tecnológicas basadas en contenedores se ha convertido en un desafío estratégico para las organizaciones que buscan reducir costos, aumentar la flexibilidad y disminuir la dependencia de soluciones propietarias. En este contexto, la migración desde Red Hat OpenShift [1] hacia un servicio Kubernetes [2] administrado o Engine nativo en *cloud* representa una alternativa que permite aprovechar un entorno Kubernetes administrado, manteniendo las capacidades de escalabilidad, disponibilidad y automatización requeridas por aplicaciones empresariales.

Sin embargo, este proceso implica desafíos técnicos y operacionales que pueden afectar la continuidad del servicio y el desempeño de las aplicaciones, si no se ejecuta de manera planificada. Por ello, la presente investigación propone una estrategia estructurada para la migración de cargas de trabajo desde OpenShift hacia Oracle Kubernetes Engine (OKE) [3], integrando etapas de evaluación, preparación, migración, validación y optimización.

Finalmente, mediante la ejecución de pruebas comparativas de rendimiento, escalabilidad y resiliencia, junto con un análisis económico del costo total de propiedad, se busca demostrar que una migración ordenada permite mantener o incluso mejorar algunos aspectos de la administración y desempeño de las aplicaciones y *clusters*, obteniendo beneficios operacionales y financieros que justifican la inversión inicial en un horizonte de tres años.

Es importante señalar que, si bien esta tesina se enfoca en la migración hacia OKE, la misma estrategia y etapas son homologables para cualquier proveedor *cloud*, teniendo en consideración las dependencias propias de cada uno.

1.2 Contexto y problemática

En la actualidad las organizaciones buscan distinguirse dentro de un mercado competitivo y que se renueva con celeridad. En este contexto, una de las premisas es mejorar el *time to market* de sus aplicaciones y mejoras tecnológicas de cara a sus clientes, apuntando a arquitecturas basadas en microservicios y contenedores.

En base a lo anterior, muchas empresas optan por el uso de OpenShift como plataforma para administración y despliegue de contenedores, basando su decisión en el enfoque empresarial de la herramienta, su modelo de seguridad y el ecosistema de componentes integrados, como lo son operadores DevOps (p. ej., Tekton² y ArgoCD³).

Si bien OpenShift responde de manera apropiada a las necesidades previamente mencionadas, con el tiempo aparecen diversas motivaciones para migrar a un Kubernetes Engine, el cual es un servicio gestionado

² Tekton: Framework de código abierto, *cloud native* y de Kubernetes, diseñado para crear despliegues mediante CI/CD.

³ ArgoCD: Herramienta de entrega continua (CD) diseñada específicamente para Kubernetes, que apunta a GitOps

por un proveedor cloud que permite el uso de Kubernetes. Existen servicios gestionados tanto en GCP, AWS y Azure, entre otros, pero esta tesina se enfocará en el uso de Oracle Kubernetes Engine (OKE), servicio entregado por Oracle Cloud Infrastructure (OCI). Dentro de las motivaciones más frecuentes podemos mencionar:

- Optimización de costos y FinOps⁴. Costos de licenciamiento por suscripción (en este caso específico se utilizará la suscripción OpenShift Container Platform [4]), costos asociados a soporte y esfuerzos de operación (como tareas manuales y repetitivas), como también a infraestructura sobredimensionada.
- Estrategia cloud y servicios administrados. Preferencia por delegar en un clúster a un proveedor cloud el plano de control (nodos de control) y componentes críticos, como pueden ser actualizaciones, alta disponibilidad (*High Availability* o HA) y plan de recuperación ante desastres (*Disaster Recovery Plan* o DRP).
- Reducción de dependencia tecnológica. Dependencia de capacidades propias de OpenShift puede dificultar la portabilidad de las aplicaciones; como por ejemplo las rutas (*Routes*) que exponen servicios desde el clúster hacia el exterior, restricciones de contexto de seguridad (*Security Context Constraints* o SCC) u operadores específicos (software que automatiza el ciclo de vida completo de aplicaciones complejas; también llamado OLM u Operators),
- Estandarización y gobernanza. Necesidad de políticas, trazabilidad y control de cambios consistentes en entornos híbridos.
- Integración con servicios nativos de nube. Tales como red, identidad y seguridad (IAM Policy), observabilidad, almacenamiento y DRP.

Oracle Cloud ofrece un servicio de Kubernetes administrado OKE y un ecosistema de recursos integrados (como red, balance de carga, política IAM, *Vault*, *logging/monitoring* y almacenamiento, entre otros). Sin embargo, migrar desde OpenShift hacia cualquier Kubernetes Engine requiere una aproximación metodológica debido a diferencias en componentes, seguridad y prácticas operacionales.

Por tal razón, muchas de estas empresas se enfrentan a la adopción de arquitecturas basadas en microservicios y contenedores, mirando las fortalezas tecnológicas y económicas de diversos proveedores cloud, llevándolos (en la mayoría de los casos) a desplegar arquitecturas multicloud e híbridas, repartiendo aplicaciones y bases de datos entre las distintas plataformas cloud existentes y on premise.

1.3 Definición del problema

Las organizaciones que intentan migrar desde OpenShift hacia un entorno Kubernetes administrado como el de OKE, frecuentemente se enfrentan a las mismas dificultades:

- Ausencia de una estrategia formal de migración: la migración suele tratarse como un conjunto de pasos ad-hoc, sin criterios de aceptación ni controles de riesgo.
- Dependencias de OpenShift: uso de componentes específicos como Routes, utilizadas para exponer servicios fuera del clúster; SCC (*Security Context Constraints*), que establecen restricciones de seguridad para la ejecución de Pods; ImageStreams, que permiten gestionar y referenciar imágenes de contenedores mediante etiquetas (*tags*); BuildConfigs, que automatizan la construcción de imágenes a partir del código fuente (además, de plantillas de despliegue); Operators, que automatizan la administración de aplicaciones; y OLM (*Operator Lifecycle Manager*), encargado de gestionar el ciclo de vida de estos operadores. Estos componentes pueden requerir adaptación o reemplazo al migrar hacia un entorno Kubernetes estándar.
- Brechas en seguridad y gobierno: diferencias entre SCC, que establece restricciones de seguridad en OpenShift; RBAC, que define roles junto a permisos de acceso y PSA, que

⁴ Ver en FinOps Foundation: <https://www.finops.org/introduction/what-is-finops/>

controla las condiciones de seguridad para la admisión de Pods. También se consideran la gestión de registros y secretos, las políticas de seguridad y los mecanismos de auditoría.

- Complejidad en red y exposición [5]: diferencias en mecanismos como Routes e Ingress, utilizados para exponer aplicaciones; TLS, para comunicaciones seguras; balanceadores de carga de capa 4 (transporte) y capa 7 (aplicación); además de DNS y *service mesh* para comunicación y gestión del tráfico entre servicios.
- Persistencia y estado: considera la migración de volúmenes persistentes, el uso de CSI (Container Storage Interface) para integrar almacenamiento, *snapshots* para generar copias de un volumen, mecanismos de *backup/restore* y posibles dependencias de sistemas de almacenamiento específicos.
- Continuidad operacional: considera mecanismos para minimizar el downtime o tiempo de indisponibilidad durante la migración, utilizando estrategias como blue/green, que mantiene dos entornos en paralelo, o canary, que realiza una transición gradual, además de disponer de un plan de reversa ante fallas.

En base a lo anterior, podemos resumir el problema como:

- Actualmente, muchas organizaciones que utilizan OpenShift enfrentan dificultades para migrar sus cargas de trabajo hacia algún servicio Kubernetes nativo, en este caso, Oracle Kubernetes Engine, debido a la ausencia de una estrategia estructurada que guíe el proceso. Esto puede generar riesgos operativos, interrupciones del servicio, sobrecostos, retrabajo e incertidumbre respecto al rendimiento y la continuidad operacional posterior a la migración.
- En consecuencia, es necesario disponer de una estrategia estructurada, repetible y medible para migración de cargas de trabajo desde OpenShift hacia OKE, que apunte a reducir riesgos operativos, optimice costos y mantenga o mejore el rendimiento y la eficiencia de despliegue. Para tener confianza en la estrategia, es necesario que esté validada su aplicación.

1.3.1 Proceso de migración actual

Actualmente existen alternativas para apoyar procesos de migración hacia OCI, entre ellas Rackware⁵, una solución comercial que permite automatizar la migración de cargas de trabajo entre diferentes entornos tecnológicos. Su utilización considera costos asociados al servicio y está orientada principalmente a facilitar la transferencia de cargas de trabajo desde una infraestructura de origen hacia una de destino.

Este tipo de solución permite abordar escenarios de migración cercanos al enfoque *Lift and Shift*, en los cuales se busca trasladar una carga de trabajo hacia una nueva plataforma procurando si realizar cambios en su configuración y características originales. Si bien este enfoque permite automatizar y simplificar determinadas actividades asociadas al traslado de las cargas, no constituye por sí mismo una estrategia integral para planificar y controlar todas las etapas involucradas en una migración desde OpenShift hacia OKE.

En este contexto, la propuesta de esta tesina presenta un enfoque diferente y complementario. En lugar de desarrollar una nueva herramienta de migración, se propone una estrategia estructurada, reproducible y medible que permita abordar el proceso desde una perspectiva técnica y operacional. Esta estrategia considera el análisis del entorno de origen, la identificación de dependencias específicas de OpenShift, la definición de la arquitectura objetivo en OKE, la preparación de la plataforma, la adaptación y migración de las aplicaciones, la transición al nuevo entorno y las actividades posteriores de estabilización y optimización. La propuesta desarrollada en esta tesina busca proporcionar un marco metodológico que permita organizar, controlar y reproducir el proceso completo de migración en distintos escenarios.

⁵ RackWare es una plataforma empresarial que automatiza y simplifica tanto la migración desde diversos entornos a OCI y la recuperación ante desastres de cargas de trabajo tecnológicas.

1.3.2 Dificultades encontradas en procesos de migraciones con Rackware

Al ser Rackware una herramienta pensada en una migración AS-IS, una vez finalizado el proceso de migración, nos encontramos con dependencias no resueltas, que requieren un replanteamiento de la estrategia de migración una vez ya finalizado tal proceso.

Por tal razón, el tiempo y esfuerzo se duplica tanto dentro de los equipos de desarrollo como en los equipos de operación y administración. Esto conlleva volver a realizar un levantamiento de servicios, configuraciones y dependencias, entre otros, generando retraso en proyectos, agotamiento en personal e intermitencias en servicios post-migración.

1.4 Propuesta de solución

A diferencia de otras iniciativas existentes, que se enfocan en casos aislados o específicos para pruebas de concepto (Proof of Concept, POC), como la presentada en el ejemplo “Migrate any OpenShift or Kubernetes workloads to OKE cloud using Migration as a Service” [6], el principal aporte de esta tesina consiste en el diseño, implementación y validación de una estrategia estructurada, reproducible y medible para la migración de cargas de trabajo desde OpenShift hacia OKE. Esta estrategia está orientada a organizaciones que buscan optimizar costos, reducir la dependencia tecnológica de plataformas propietarias y mejorar la eficiencia operativa de sus entornos de contenedores.

Asimismo, la innovación propuesta en esta investigación no radica en el desarrollo de una nueva herramienta o servicio de migración, como Rackware o la solución presentada en [6], sino en la formalización y validación empírica de una estrategia aplicable y reproducible en distintos entornos organizacionales. De esta forma, el trabajo entrega un marco metodológico que puede ser utilizado como guía para planificar, ejecutar y validar procesos de migración, disminuyendo los riesgos asociados y facilitando su adopción en escenarios reales.

1.5 Objetivos generales planteados

Diseñar y validar una estrategia para la migración desde entornos OpenShift hacia Oracle Kubernetes Engine (OKE), considerando aspectos técnicos, económicos y operacionales, que permita ejecutar la transición de forma segura, eficiente y controlada.

En base a lo anterior, se plantean los siguientes objetivos específicos:

1. Levantamiento y comprensión del entorno OpenShift de origen, mediante inventario de aplicaciones, dependencias y componentes específicos relacionados, criticidad de aplicaciones y requisitos no funcionales.
2. Definir la arquitectura objetivo en OKE, a nivel de red, seguridad, almacenamiento, observabilidad y estándares operacionales.
3. Diseñar la estrategia de migración con fases, actividades, entregables, controles de riesgo y criterios de aceptación.
4. Construir una matriz de evaluación de equivalencias y brechas entre OpenShift y OKE, considerando rutas de exposición de servicios, seguridad en Pods, Operadores, repositorio centralizado de imágenes de contenedores (registry) y almacenamiento de imágenes, pipelines CI/CD y monitoreo.
5. Evaluar beneficios económicos y tecnológicos mediante un modelo comparativo.
6. Validar la estrategia con un caso de estudio, midiendo antes y después métricas operacionales, de entrega y rendimiento.

1.6 Beneficios esperados

1.6.1 Beneficios tecnológicos

Dentro de los beneficios tecnológicos se busca la integración con OCI, mediante red, IAM, almacenamiento, registry, stack de observabilidad nativo, la portabilidad y despliegue de aplicaciones agnósticas al proveedor (como lo son Red Hat, Oracle, AWS, etc.), estandarización de despliegue de infraestructura y mejor capacidad de auditoría, mejoras en gobernanza, mediante políticas, RBAC, control de *drift* (desvíos), configuración declarativa.

1.6.2 Beneficios económicos

Como beneficio económico, se busca la reducción o eliminación de licenciamiento asociado a la plataforma OpenShift, la optimización de costos de operación al usar un servicio administrado, la optimización del uso de recursos mediante el dimensionamiento y escalado automático (*autoscaling*) de los nodos del clúster, permitiendo adaptar la capacidad computacional al consumo real y a la carga de trabajo de las aplicaciones, costos de oportunidad, reducción de tiempos de entrega y de interrupciones (impacto en negocio).

1.7 Definición de hipótesis y validación

Este trabajo propone la siguiente hipótesis:

- Si se aplica una estrategia estructurada de migración desde OpenShift hacia OKE, apoyada en automatización, entonces se logrará reducir costos operativos y de plataforma, mejorar la eficiencia de despliegues de infraestructura y mantener o mejorar el rendimiento operacional de las aplicaciones migradas.

La estrategia de validación de la hipótesis considera una variable independiente de innovación que compara un escenario inicial de base (AS-IS) con un escenario final de aplicación de la estrategia propuesta (TO-BE). Como variables dependientes se usan indicadores tanto técnicos descritos en el capítulo 4 como operacionales y económicos descritos en el capítulo 3, que permiten comparar ambos escenarios. El escenario AS-IS corresponde a aplicaciones ejecutadas sobre Red Hat OpenShift. El escenario TO-BE se obtiene aplicando la estrategia estructurada de migración propuesta en esta tesina, mediante la cual las aplicaciones son migradas de forma controlada hacia Oracle Kubernetes Engine (OKE).

1.8 Variables dependientes

Las variables dependientes corresponden a los efectos que se espera observar como resultado de la aplicación de la estrategia estructurada de migración desde OpenShift hacia Oracle Kubernetes Engine. Estas variables se desprenden directamente de la hipótesis planteada y serán evaluadas mediante la comparación entre el escenario inicial AS-IS (OpenShift) y el escenario posterior a la migración TO-BE (OKE).

Se consideran como variables la eficiencia de despliegue y operación, el rendimiento operacional, el costo operativo y de plataforma, los cuales serán detallados a continuación.

1.8.1 Eficiencia de despliegue y operación

La segunda variable corresponde a la eficiencia de despliegue y operación de infraestructura, cuyo objetivo es determinar si la aplicación de la estrategia y de los mecanismos de automatización permite mejorar los procesos asociados al despliegue, aprovisionamiento y administración de la infraestructura subyacente.

Para su evaluación se consideran indicadores como el lead time, entendido como el tiempo transcurrido desde que se registra un cambio hasta que este se encuentra desplegado, el tiempo requerido para aprovisionar infraestructura, la cantidad de tareas manuales necesarias y el nivel de trazabilidad de los cambios. También se considera la capacidad de identificar diferencias no deseadas entre la configuración declarada y el estado real del entorno (drift).

Adicionalmente, se consideran mecanismos de automatización y prácticas GitOps que permitan mantener trazabilidad sobre los cambios realizados y facilitar procesos de reversión (rollback) cuando sea necesario. Estos elementos permitirán evaluar si la estrategia propuesta disminuye la dependencia de tareas manuales y facilita una operación más estandarizada y reproducible.

1.8.2 Rendimiento operacional

La variable de rendimiento operacional tiene como propósito determinar si las aplicaciones mantienen o mejoran su comportamiento después de la migración hacia OKE. Para ello se consideran indicadores relacionados con el tiempo de respuesta, throughput, tasa de errores, consumo de recursos, escalabilidad y capacidad de recuperación ante fallas. Estos indicadores serán obtenidos bajo condiciones comparables en ambos escenarios, permitiendo evaluar posibles variaciones atribuibles al cambio de plataforma.

Dentro de esta variable también se considera el comportamiento de la plataforma frente a cambios en la demanda. Para ello se evalúan los tiempos asociados al escalado de Pods y nodos, con especial atención al escalado automático de nodos debido a su relación con el aprovechamiento de la capacidad computacional disponible. Asimismo, se considera el tiempo de recuperación ante fallas y la disponibilidad percibida de las aplicaciones mediante indicadores como errores, reinicios o degradaciones del servicio.

La finalidad de estas mediciones no es demostrar que OKE posee necesariamente un rendimiento superior a OpenShift, sino determinar si, después de aplicar la estrategia de migración, las aplicaciones pueden mantener niveles de rendimiento equivalentes o presentar mejoras en determinados indicadores. Esto es consistente con la hipótesis planteada, que establece como resultado esperado mantener o mejorar el rendimiento operacional de las aplicaciones migradas.

1.8.3 Costos operativos y de plataforma

La tercera variable corresponde a los costos operativos y de plataforma, cuyo propósito es determinar el impacto económico asociado a la migración. Para ello se consideran los costos de infraestructura, licenciamiento y servicios requeridos para la operación de OpenShift y OKE, además del esfuerzo operacional asociado a tareas manuales o repetitivas, expresado en horas de trabajo.

A partir de estos antecedentes será posible comparar los costos correspondientes a los escenarios AS-IS y TO-BE y determinar los potenciales ahorros derivados de la migración. Asimismo, la evaluación económica considerará la inversión necesaria para ejecutar el proceso de migración y su relación con los ahorros posteriores, permitiendo analizar indicadores como el retorno de la inversión (ROI) y el período de recuperación (Payback).

Esta variable permitirá evaluar la componente económica de la hipótesis, determinando si la reducción de costos recurrentes de infraestructura, licenciamiento y operación permite compensar la inversión inicial requerida para realizar la migración dentro del horizonte temporal considerado por la investigación, apuntando a la optimización de los presupuestos, uniendo a los equipos de tecnología y finanzas, FinOps.

1.9 Variables independientes

La variable independiente de esta investigación corresponde a la aplicación de la estrategia estructurada y controlada de migración desde OpenShift hacia Oracle Kubernetes Engine, apoyada en mecanismos de automatización. Esta variable representa el factor introducido entre el escenario AS-IS y el escenario TO-BE, cuyos efectos serán evaluados mediante las variables dependientes definidas en la sección anterior.

1.10 Alcance y delimitaciones

Se considera dentro del alcance, la migración de aplicaciones contenidas y sus manifiestos Kubernetes (Deployments, StatefulSets, Services, Ingress), el mapeo y reemplazo de dependencias específicas de OpenShift por alternativas en OKE, definiciones e implementaciones de red, seguridad, almacenamiento y observabilidad necesarios para operar en OKE y la definición de estrategia con fases, artefactos, matrices y métricas.

Se considera fuera del alcance (para acotar y asegurar viabilidad), la migración profunda de pipelines CI/CD específicos (solo se considera su adaptación mínima a GitOps), el rediseño completo de la aplicación (se consideran ajustes necesarios para compatibilidad), la optimización avanzada de performance de código (se mide performance operacional/infra).

1.11 Áreas interesadas en la solución

Esta tesina apunta al trabajo en conjunto de los siguientes equipos tecnológicos, equipo de desarrollo (desarrolladores, ingenieros DevOps y QA), equipo cloud (administradores cloud y arquitectos cloud), equipo de monitoreo y operaciones, equipo FinOps

1.12 Organización del informe

La presente tesina se organiza en los siguientes capítulos:

- Capítulo 2: marco teórico, alcances, proceso de migración existente y sus dificultades, resultados esperados y trabajos relacionados.
- Capítulo 3: especificación de la propuesta de la solución, nueva arquitectura OKE propuesta, etapas de la migración, esfuerzos estimados, estimación de tiempos por etapas, beneficios, ROI y Payback.
- Capítulo 4: detalle de las pruebas realizadas y análisis de resultados.
- Capítulo 5: conclusiones.
- Finalmente, se agregan anexos e información de interés.

2 Marco teórico y estado del arte

El marco teórico adoptado se basa en los principios de los entornos cloud, la orquestación de contenedores y las estrategias de migración tecnológica, proponiendo a Kubernetes como entorno estándar para la gestión y administración de aplicaciones contenerizadas, analizando las diferencias entre OpenShift y Oracle Kubernetes Engine (OKE), así como los desafíos asociados a la portabilidad entre ambas plataformas.

2.1 Arquitectura actual de OpenShift (AS-IS)

La arquitectura actual existente desde un entorno Azure consta de nodos maestros, denominados *Control Plane*, los cuales actúan como centro de control del cluster de OpenShift y nodos *workers*, en los cuales viven las aplicaciones desplegadas.

Estos coexisten dentro de una *subnet privada* que se consumen recursos de una base de datos PostgreSQL como también de otros servicios de Azure, como lo son el balanceador de carga, firewall y WAF.

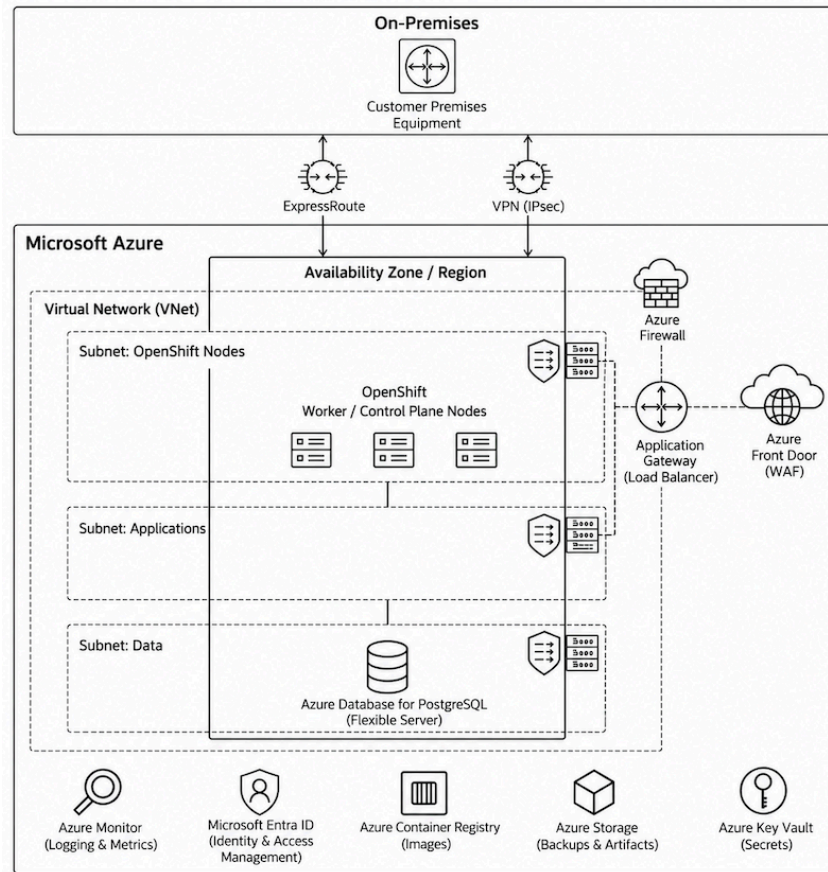


Figura 1: Arquitectura de referencia OpenShift

2.2 Kubernetes⁶ (k8s) y servicios administrados

Kubernetes es una plataforma portable y extensible de código abierto para administrar cargas de trabajo y servicios, provee abstracciones para desplegar y operar aplicaciones en base a contenedores. Dentro de sus componentes más importantes se pueden destacar:

- **Pod:** unidad más pequeña de ejecución, contiene uno o más contenedores.
- **Node:** corresponde a los servidores, los cuales pueden ser servidores virtuales o físicos; dentro de éstos se ejecuta los Pod.
- **Clúster:** conjunto de nodos físicos o virtuales que, administrados por Kubernetes, proporcionan la infraestructura necesaria para desplegar, ejecutar y orquestar aplicaciones basadas en contenedores.
- **Control Plane:** conjunto de nodos de control del clúster que actúa como el componente central de Kubernetes, encargado de administrar y coordinar el funcionamiento del clúster.
- **Data Plane:** conjunto de nodos en los cuales se ejecutan las aplicaciones basadas en los Pods.
- **Deployment.** recurso de Kubernetes que define declarativamente la forma en que una aplicación es desplegada y administrada dentro del clúster. Permite gestionar el número de réplicas, realizar actualizaciones controladas (*rolling updates*), volver atrás una actualización (*rollbacks*) y escalar la aplicación.

⁶ Kubernetes es un proyecto de la CNCF, ver: <https://kubernetes.io/>

- **Service:** recurso de Kubernetes que expone uno o más Pods mediante una dirección IP y un nombre estable, permitiendo la comunicación entre aplicaciones dentro del clúster o con clientes externos.
- **Ingress:** recurso de Kubernetes que administra el acceso HTTP y HTTPS hacia los servicios del clúster, definiendo reglas de enrutamiento basadas en direcciones URL o nombres de dominio. En OKE, su implementación se integra con un OCI Load Balancer, administrado por el OCI Cloud Controller Manager.
- **ConfigMap:** objeto de Kubernetes utilizado para almacenar configuraciones no sensibles de las aplicaciones, permitiendo desacoplar los parámetros de configuración del código de las imágenes de contenedores.
- **Secret:** objeto de Kubernetes diseñado para almacenar y administrar información sensible, como contraseñas, tokens, certificados o claves de acceso, proporcionando un mecanismo más seguro para su utilización por parte de las aplicaciones.

Por otro lado, un servicio administrado de Kubernetes (Engine) busca reducir la carga operativa, asumiendo tareas del Control Plane tales como actualizaciones, alta disponibilidad, entre otros, y estandarizando integraciones de red y almacenamiento.

2.3 OpenShift Container Platform (OCP⁷)

OpenShift, por su parte, es una plataforma empresarial de Red Hat construida sobre Kubernetes, que incorpora mecanismos adicionales de seguridad, políticas de ejecución, componentes para el desarrollo y despliegue de aplicaciones y herramientas que facilitan su operación. Estas capacidades agregan valor, pero también pueden generar una mayor dependencia con la plataforma propietaria de Red Hat.

2.4 Oracle Kubernetes Engine (OKE⁸) en OCI

OKE, por otro lado, es un servicio de Kubernetes administrado e integrado con OCI, que facilita la gestión de recursos de red mediante redes virtuales (VCN), subredes, gateways y grupos de seguridad (NSG), además de balanceadores de carga a nivel de red y a nivel de aplicación, DNS y certificados. Asimismo, permite integrar almacenamiento mediante Block Volume o File Storage y utilizar IAM para gestionar la autenticación y el control de acceso.

2.5 Automatización

Para el desarrollo de esta tesina se consideran como herramientas de apoyo a Ansible⁹, que permite estandarizar aprovisionamiento y configuración (infraestructura, clúster, add-ons), reduciendo drift y trabajos manuales.

2.6 Trabajos relacionados más relevantes

- Migrate any OpenShift or Kubernetes workloads to OKE cloud using Migration as a Service [6] que presenta el uso de Rackware para automatizar la migración de cargas desde OpenShift u otros Kubernetes hacia OKE, sin realizar los cambios requeridos en las configuraciones.
- Understanding Argo CD: Kubernetes GitOps Made Simple [7] describe Argo CD y el modelo GitOps para automatizar y mantener de forma declarativa el despliegue y sincronización de aplicaciones sobre Kubernetes, pero no aplica a una migración como tal.

⁷ OCP, ver: <https://www.redhat.com/en/technologies/cloud-computing/openshift/container-platform>

⁸ OKE, ver: <https://www.oracle.com/cloud/cloud-native/kubernetes-engine/>

⁹ Ansible, ver: <https://docs.ansible.com/>

- VMware, Top 5 Kubernetes Operations Challenges and How to Mitigate [8] que analiza los principales desafíos asociados a la operación de Kubernetes y las prácticas que permiten reducir su complejidad operacional.

Los trabajos analizados abordan aspectos relacionados y complementarios a esta investigación. Rackware [6] se enfoca en una herramienta para automatizar la migración de cargas hacia OKE, sin modificar configuraciones ni agregar ningún tipo de cambio; Argo CD [7] aborda la automatización y gestión de aplicaciones mediante GitOps; y VMware [8] analiza desafíos asociados a la operación de plataformas Kubernetes.

A diferencia de estos trabajos, esta tesina propone una estrategia específica, estructurada y reproducible para la migración desde OpenShift hacia OKE, considerando el proceso completo desde el análisis del entorno de origen y sus dependencias hasta la definición del escenario objetivo, migración, continuidad operacional y optimización. Adicionalmente, incorpora una dimensión económica y establece criterios para validar posteriormente la estrategia mediante un caso de estudio. Por lo tanto, los trabajos analizados constituyen herramientas, prácticas o lineamientos que pueden complementar determinadas etapas, mientras que el aporte de esta investigación consiste en formalizar y validar metodológicamente el proceso integral de migración.

3 Propuesta de solución

Como se estableció en la sección 1.4, la propuesta de solución de esta tesina consiste en el diseño de una estrategia estructurada, reproducible y medible para la migración de cargas de trabajo desde Red Hat OpenShift hacia Oracle Kubernetes Engine (OKE). Su propósito es proporcionar un marco de trabajo que permita planificar, ejecutar y controlar este tipo de migraciones de manera sistemática, disminuyendo la dependencia de decisiones ad hoc y del conocimiento individual de los especialistas.

En este capítulo se presenta exclusivamente la definición de la estrategia propuesta, estableciendo su estructura, etapas, objetivos, entradas, actividades, entregables y criterios de salida. La aplicación de esta estrategia sobre un caso de estudio, incluyendo la planificación y ejecución de la migración, las pruebas realizadas y el análisis de los resultados obtenidos, se presenta posteriormente en el capítulo 4. De esta forma, se establece una separación explícita entre la propuesta de solución y el trabajo experimental realizado para su aplicación y validación.

La estrategia se construye a partir de las principales dificultades identificadas durante el análisis del problema, entre ellas las dependencias específicas de OpenShift, las diferencias de arquitectura, seguridad, red y almacenamiento respecto de un servicio Kubernetes administrado, así como la necesidad de mantener la continuidad operacional durante la transición y evaluar los beneficios derivados de la migración. A partir de estos elementos, el proceso se descompone en un conjunto de etapas relacionadas entre sí, permitiendo abordar de manera progresiva los distintos dominios involucrados.

Para efectos de su definición, la propuesta se estructura en dos dimensiones complementarias: técnica-operacional y económica. Por una parte, la dimensión técnica-operacional establece el proceso necesario para caracterizar el entorno de origen, definir el escenario objetivo, preparar la plataforma OKE, adaptar y migrar las aplicaciones, realizar la transición hacia el nuevo entorno y efectuar posteriormente su estabilización y cierre. Por otra parte, la dimensión económica establece los criterios e indicadores necesarios para evaluar la conveniencia de la migración, considerando la inversión requerida, los costos de plataforma e infraestructura, los potenciales ahorros y el período de recuperación de la inversión.

La dimensión técnica-operacional se organiza en siete etapas, desde la preparación y gobernanza inicial del proyecto hasta la optimización y cierre. Cada etapa posee un objetivo específico y considera entradas, actividades, entregables y criterios de salida que permiten verificar el cumplimiento de las condiciones

necesarias antes de avanzar hacia la etapa siguiente. Adicionalmente, la estrategia contempla actividades transversales de gestión de riesgos, comunicación, seguimiento, gestión de cambios y gobernanza, las cuales acompañan el proceso completo de migración.

En consecuencia, las etapas y actividades descritas en este capítulo representan el marco metodológico que se propone para abordar una migración, y no la descripción de una migración particular. En el capítulo 4, esta estrategia será aplicada a un caso de estudio con el propósito de evaluar su aplicación y contrastar los resultados obtenidos con la situación inicial, considerando aspectos técnicos, operacionales y económicos.

3.1 Resultados esperados

Como resultados esperados, se busca obtener una estrategia documentada, reproducible y apoyada en artefactos reutilizables, que permita estructurar futuros procesos de migración desde OpenShift hacia OKE. Asimismo, se espera reducir las tareas manuales y mejorar la trazabilidad de los cambios mediante prácticas de automatización y GitOps, manteniendo o mejorando indicadores asociados al despliegue y la operación de las aplicaciones.

Finalmente, se espera obtener una comparación entre ambos escenarios desde una perspectiva técnica, operacional y económica, además de documentar las principales dificultades identificadas durante la migración y las soluciones aplicadas para resolverlas.

3.2 Beneficios de la migración

Se consideran como beneficios derivados de la migración pueden clasificarse en beneficios cuantificables y beneficios cualitativos.

Beneficios cuantificables:

- Eliminación del costo de licenciamiento de OpenShift.
- Reducción del costo de infraestructura.
- Disminución del costo operativo asociado a la administración de la plataforma.
- Optimización del consumo de recursos mediante mecanismos de escalabilidad automática.

Beneficios cualitativos:

- Reducción del vendor lock-in.
- Mayor portabilidad de las aplicaciones.
- Simplificación de la administración del clúster.
- Integración con servicios nativos de Oracle Cloud Infrastructure.
- Mayor flexibilidad para futuras estrategias multicloud.
- Mejor soporte para prácticas DevSecOps y automatización.

Aunque estos beneficios no siempre pueden expresarse monetariamente, representan ventajas relevantes para organizaciones que buscan modernizar su plataforma tecnológica y reducir riesgos operacionales.

3.3 Propuesta técnica

3.3.1 Arquitectura propuesta para la migración (TO-BE)

La arquitectura propuesta se define en un entorno híbrido en el cual las cargas de trabajo son ejecutadas en OKE dentro de una red privada (VCN) en la región de OCI Santiago (ver figura 1).

Esta se encuentra conectada de forma segura al *datacenter* del cliente mediante enlaces dedicados de OCI FastConnect, el cual es un servicio que permite conectar una red on-premise (centro de datos que cliente posee en su infraestructura local) directamente con Oracle Cloud mediante un enlace privado y dedicado.

En este esquema, las aplicaciones desplegadas sobre OKE se ubican en una subred de aplicación, mientras que los servicios de base de datos, como PostgreSQL, residen en una subred separada para reforzar la seguridad y el aislamiento.

La arquitectura incorpora controles de acceso mediante OCI Identity and Access Management, así como el uso de servicios nativos como Object Storage y Service Gateway para el acceso privado a recursos sin exposición a Internet. Este diseño permite desacoplar las aplicaciones de componentes propietarios de plataforma, favoreciendo su portabilidad sobre Kubernetes estándar, al mismo tiempo que mantiene integración segura con sistemas on-premise y capacidades de escalabilidad y operación propias de la nube.

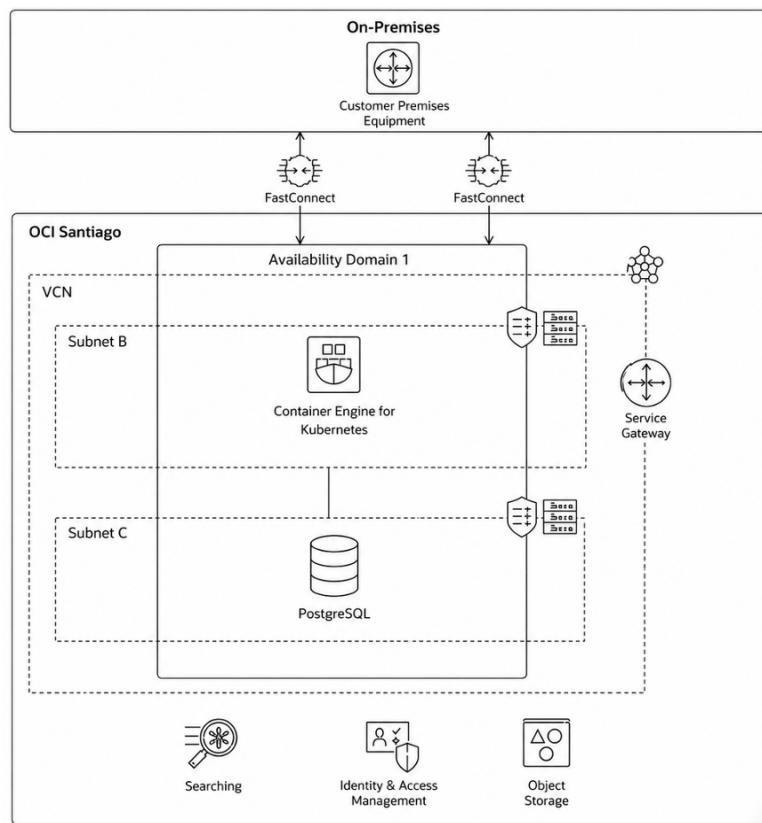


Figura 2: Arquitectura de referencia para migración

Para que los servicios y aplicaciones puedan ser expuestos a Internet se recomienda el uso de un Ingress Controller (ver figura 3), de esta forma se despliega un solo balanceador de carga que apunte a las aplicaciones de OKE mediante el path de la URL y no tener que crear un balanceador de carga por aplicación.

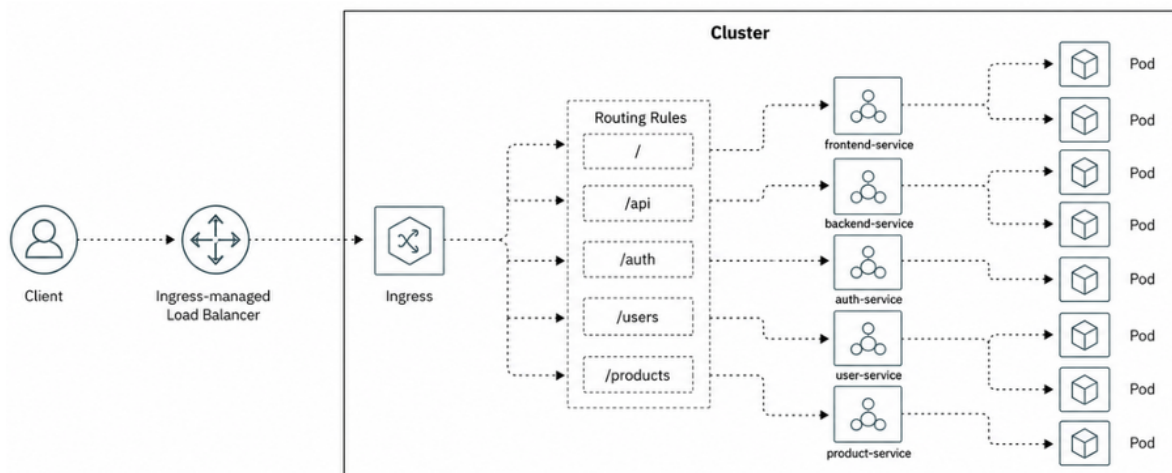


Figura 3: Arquitectura Referencia Ingress Controller

3.3.2 Propuesta de estrategia de migración

La migración de aplicaciones desde OpenShift hacia cualquier entorno OKE, no es un proceso del tipo *Lift and Shift* (migración de aplicaciones desde un entorno a otro, sin modificar su código ni rediseñar su arquitectura), en el cual se toma lo existente y se mueve tal cual, a un nuevo entorno, no obstante, tampoco es un proceso donde se deban reescribir completamente las aplicaciones existentes si se hace de una forma metodológica, tal como es la propuesta de esta tesina.

Para obtener una comprensión completa del entorno, primero requerimos de una visión general del proceso. Para ello, la estrategia propuesta se define en 7 etapas, cada una con entradas, actividades definidas y criterios de salida.

La definición de las etapas de la estrategia se realizó a partir de la descomposición del proceso de migración en los principales dominios de trabajo identificados durante el análisis de la problemática. Para ello, se consideraron las dependencias existentes en OpenShift, los requerimientos de arquitectura del entorno objetivo, la necesidad de disponer previamente de la infraestructura OKE, la adaptación y migración de las aplicaciones, los requerimientos de continuidad operacional durante la transición al nuevo entorno y, finalmente, las actividades de estabilización y optimización posteriores a la migración.

A partir de este análisis, las actividades fueron agrupadas de acuerdo con su objetivo, dependencias y secuencia lógica de ejecución. Adicionalmente, se incorporó una etapa inicial de preparación y gobernanza, destinada a establecer el alcance, los responsables, los criterios de éxito y los mecanismos de control del proyecto. De esta forma, la estrategia resultante se estructura en siete etapas consecutivas, donde los resultados obtenidos en cada una constituyen insumos para las etapas posteriores.

Esta organización permite abordar la migración como un proceso controlado y trazable, en lugar de un conjunto de actividades aisladas, facilitando además la definición de entradas, actividades, entregables y criterios de salida para cada etapa.

A modo de resumen, y en base a cómo se representa en la figura 4, las etapas serían:

Etapla 0 de preparación y gobernanza de proyecto, etapa 1 de análisis entorno OpenShift, etapa 2 de arquitectura y TO-BE en OKE, etapa 3 de despliegue entorno OKE, etapa 4 de migración de aplicaciones, etapa 5 de transición final o cutover y continuidad operativa, etapa 6 de optimizaciones y cierre de proyecto.



Figura 4: Etapas propuestas para la estrategia de migración

3.4 Aspectos técnicos complementarios

Además de las variables directamente asociadas a la hipótesis, la aplicación de la estrategia requiere considerar aspectos técnicos relacionados con seguridad, gobierno, networking y exposición de servicios, debido a las diferencias existentes entre OpenShift y OKE. Estos elementos no constituyen variables dependientes principales de la hipótesis, pero son necesarios para determinar que las aplicaciones puedan operar correctamente después de la migración.

En materia de seguridad y gobierno, se considera la adaptación desde mecanismos específicos de OpenShift, como Security Context Constraints (SCC), hacia mecanismos disponibles en Kubernetes, como Pod Security Admission (PSA) y Role-Based Access Control (RBAC). Mientras SCC establece restricciones sobre las condiciones bajo las cuales pueden ejecutarse los Pods en OpenShift, PSA permite establecer condiciones de seguridad para su admisión en Kubernetes y RBAC permite controlar las acciones que usuarios e identidades pueden realizar sobre los recursos del clúster. También se consideran la gestión de secrets, la auditoría de cambios y la detección de drift.

Respecto del networking y exposición de servicios, se considera la adaptación de mecanismos específicos de OpenShift, como Routes, hacia mecanismos de Ingress utilizados en OKE. Asimismo, se consideran los certificados SSL/TLS necesarios para establecer comunicaciones seguras, junto con balanceadores de carga y mecanismos de protección como Web Application Firewall (WAF).

La revisión de estos aspectos permite verificar que las diferencias entre ambas plataformas hayan sido consideradas durante la aplicación de la estrategia y que las aplicaciones puedan operar correctamente en el escenario TO-BE

4 Pruebas realizadas y resultados obtenidos

Posterior a la migración, se realizaron diferentes pruebas, con la finalidad de validar que, mediante una estrategia ordenada de migración desde OpenShift hacia OKE, las aplicaciones pueden mantener un rendimiento equivalente o superior, reduciendo costos a nivel de infraestructura, licenciamiento y operativos en el mediano plazo.

La hipótesis de esta tesina nunca buscó demostrar que OKE en todos los escenarios presenta un mejor performance, sino que puede entregar igual o mejor desempeño con menor costo total a tres años, aunque con un esfuerzo inicial de migración mayor.

4.1 Validación de la propuesta y consistencia con la hipótesis

La validación de la estrategia propuesta busca establecer una relación directa entre la hipótesis de investigación y las variables definidas para evaluar los efectos de su aplicación. La hipótesis plantea que la utilización de una estrategia estructurada para la migración desde Red Hat OpenShift hacia Oracle Kubernetes Engine (OKE), apoyada en mecanismos de automatización, permitirá reducir los costos operativos y de plataforma, mejorar la eficiencia de despliegue y operación, y mantener o mejorar el rendimiento operacional de las aplicaciones migradas. A partir de esta hipótesis se establecen tres variables dependientes principales: rendimiento operacional, eficiencia de despliegue y operación, y costos operativos y de plataforma.

Para evaluar estas variables se utilizará un enfoque basado en un estudio de caso comparativo, considerando dos escenarios: AS-IS y TO-BE. El escenario AS-IS corresponde a la operación de las aplicaciones sobre Red Hat OpenShift y permitirá establecer una línea base mediante las métricas e indicadores definidos para cada variable de estudio. Estas mediciones serán complementadas con pruebas de carga que permitan caracterizar el comportamiento de las aplicaciones antes de la migración.

Posteriormente, se aplicará la estrategia propuesta para realizar la migración hacia OKE, incluyendo las actividades de análisis, diseño, automatización, despliegue, migración y validación definidas en sus distintas fases. Una vez completada la migración, se repetirán las mediciones bajo condiciones equivalentes o comparables a las utilizadas en el escenario inicial, obteniendo de esta manera el escenario TO-BE.

Los resultados de ambos escenarios serán comparados cuantitativamente mediante los indicadores asociados a las variables de investigación. Este análisis permitirá identificar variaciones en rendimiento, tiempos y esfuerzo asociados al despliegue y operación, así como en los costos de plataforma. Adicionalmente, la evaluación cuantitativa será complementada con un análisis cualitativo de aspectos relacionados con la complejidad operacional, los riesgos identificados durante la migración y las dificultades encontradas en la aplicación de la estrategia.

La validación se realizará sobre al menos tres aplicaciones con diferentes niveles de criticidad y características técnicas, con el propósito de analizar el comportamiento de la estrategia frente a distintos tipos de cargas de trabajo y reducir la dependencia de los resultados respecto de una única aplicación. La comparación entre los escenarios AS-IS y TO-BE permitirá determinar los cambios observados después de aplicar la estrategia y establecer, a partir de la evidencia obtenida, en qué medida los resultados respaldan la hipótesis de investigación.

4.2 Escenario post-migración

Para realizar las pruebas, se planteó validar que distintos objetos a comprar estuvieran en las mismas versiones y poseyeran las mismas configuraciones

Tabla 6: Validación de versiones por objeto

Objeto	OpenShift	OKE
Aplicación	Misma versión	Misma versión
Imagen container	Misma imagen	Misma imagen
Réplicas iniciales	Igual cantidad	Igual cantidad
Configuración CPU y Memory request	Igual	Igual
Configuración CPU y Memory limit	Igual	Igual
Ingress	Configuración equivalente	Configuración equivalente
Herramienta de prueba	JMeter / K6 / Locust	JMeter / K6 / Locust

4.3 Herramientas utilizadas en pruebas

Tabla 7: Herramientas para testeo

Herramienta	Uso
K6	Pruebas de carga HTTP/API
Apache JMeter	Pruebas transaccionales más completas
Prometheus + Grafana	Métricas de CPU, memoria, pods y nodos
Kubernetes Metrics Server	Métricas básicas de pods/nodos
HPA	Escalado horizontal de pods
Cluster Autoscaler	Escalado de nodos
OCI Monitoring	Métricas nativas en OKE
OpenShift Monitoring	Métricas nativas en OpenShift

4.4 Set de pruebas realizadas

Se propuso un set de 10 pruebas que fueron realizadas tanto en OpenShift como en OKE.

Tabla 8: Set de pruebas

ID	Prueba	Objetivo
P01	Carga base	Medir comportamiento con tráfico normal
P02	Estrés progresivo	Aumentar usuarios hasta degradación
P03	Aumento repentino	Simular aumento brusco de tráfico
P04	Escalado de pods	Validar reacción del HPA
P05	Escalado de nodos	Medir tiempo para agregar nodos
P06	Saturación de CPU	Forzar consumo intensivo
P07	Saturación de memoria	Validar límites y estabilidad
P08	Resiliencia ante eliminación de pod	Eliminar pods durante carga
P09	Rolling update	Medir impacto durante despliegue
P10	Costo/performance	Relacionar rendimiento con costo

4.5 P01: Carga Base

Se midió el comportamiento con tráfico normal, en base a:

- Usuarios concurrentes: 50
- Duración: 15 minutos
- Ramp-up: 5 minutos
- Endpoint: /api/health, /login, /consulta, /transaccion
- Criterio esperado: < 1%

Tabla 9: Resultados P01

Métrica	OpenShift	OKE	Resultado esperado	Logrado
Tiempo promedio respuesta	180 ms	170–190 ms	Similar	Sí
P95 latency	450 ms	430–460 ms	Similar	Sí
Error rate	<1%	<1%	Similar	Sí
CPU promedio	45%	40–45%	Similar	Sí

4.6 P02: Pruebas de estrés

Se aumentó la cantidad de usuarios progresivamente, hasta lograr la degradación de los servicios:

- Usuarios iniciales: 50
- Usuarios máximos: 1.000
- Incremento: +100 cada 5 minutos
- Duración total: 60 minutos
- Criterio de corte: Error rate > 5% o P95 > 2 segundos

Tabla 10: Resultados P02

Métrica	OpenShift	OKE
Máximo usuarios estables	850	850–900
P95 antes de degradación	1,8 s	1,6–1,8 s
Error rate máximo aceptable	5%	5%
Resultado	Bueno	Similar

4.7 P03: Aumento repentino de tráfico

Esta prueba simuló un aumento brusco de tráfico mediante:

- Usuarios normales: 100
- Aumento repentino: aumentar a 800 usuario
- Tiempo de subida: 1 minuto
- Duración del aumento: 10 minutos
- Duración total: 30 minutos

Tabla 11: Resultados P03

Métrica	OpenShift	OKE
Tiempo de estabilización	5–7 min	4–6 min
Error rate durante aumento	2–4%	1–3%
Resultado	Bueno	Mejor

4.8 P04: Escalado Horizontal de Pods

Se validó el correcto funcionamiento de las configuraciones para HPA en base a:

- Réplicas iniciales, 2
- Réplicas máximas, 10
- Métrica HPA, CPU > 70%
- Carga, 700 usuarios concurrentes

Tabla 12: Resultados P04

Métrica	OpenShift	OKE
Tiempo para crear nuevos pods	45–90 seg	40–80 seg
Réplicas finales	8–10	8–10
Errores durante escalado	Bajo	Bajo
Resultado	Bueno	Similar

4.9 P05: Escalado de nodos

Medir los tiempos para agregar nodos en ambos entornos mediante:

- Nodos iniciales: 2
- Nodos máximos: 5
- Pods pendientes, sí
- Carga, alta y superior a capacidad inicial

Tabla 13: Resultados P05

Métrica	OpenShift	OKE
Tiempo para agregar nodo	8–12 min	4–7 min

Tiempo para programar pods	10–14 min	6–8 min
Interrupción percibida	Media	Baja
Resultado	Aceptable	Mejor

4.10 P06 Saturación de CPU

Forzar el consumo intensivo de CPU mediante:

- CPU request, 500m
- CPU limit, 1 vCPU
- Carga, Operaciones intensivas
- Duración, 30 minutos

Tabla 14: Resultados P06

Métrica	OpenShift	OKE
CPU máxima pod	95–100%	95–100%
Throttling	Similar	Similar
Error rate	<3%	<3%
Resultado	Similar	Similar

4.11 P07: Saturación de memoria

Forzar el consumo intensivo de memoria mediante:

- Memory request, 512 Mb
- Memory limit, 1 GB
- Duración, 30 minutos
- Carga, Alto volumen de sesiones / objetos

Tabla 15: Resultados P07

Métrica	OpenShift	OKE
Memory usage promedio	Similar	Similar
OOMKilled	0	0
Reinicios	0	0
Resultado	Similar	Similar

4.12 P08: Resiliencia ante eliminación de Pods

Se elimina pods durante su ejecución en base a:

- OKE: `kubectl delete pod <pod-name> -n <namespace>`
- OPC: `oc delete pod <pod-name> -n <namespace>`

Tabla 16: Resultados P08

Métrica	OpenShift	OKE
Tiempo de recuperación	30–60 seg	30–60 seg
Error rate momentáneo	<2%	<2%
Disponibilidad	>99%	>99%
Resultado	Similar	Similar

4.13 P09: Rolling Update

Se midió si un despliegue controlado afecta la experiencia del usuario, mediante:

- OKE: `kubectl rollout restart deployment <app> -n <namespace>`
- OPC: `oc rollout restart deployment/<app> -n <namespace>`

Tabla 17: Resultados P09

Métrica	OpenShift	OKE
Tiempo de despliegue	2–4 min	2–4 min
Errores durante update	<1%	<1%
Downtime	0	0
Resultado	Similar	Similar

4.14 P10: Costos vs Performance

Se Validó la relación entre costo y rendimiento, mediante:

Tabla 18: Resultados P10

Concepto	OpenShift	OKE
Costo plataforma anual	USD 21.838,68	USD 5.821,08
Ahorro anual estimado	-	USD 16.017,60
Esfuerzo inicial migración	Bajo	Alto
Costo proyecto migración	-	USD 33.200
Tiempo retorno estimado	-	Año 3
Beneficio acumulado 3 años	Menor	Mayor

5 Análisis económico

El presente análisis tiene como propósito evaluar la conveniencia económica de una migración desde Red Hat OpenShift hacia Oracle Kubernetes Engine (OKE), considerando tanto la inversión inicial requerida para ejecutar el proyecto como los beneficios económicos obtenidos durante la operación de la nueva plataforma.

Esta evaluación incorpora la estimación de esfuerzo de implementación, costo de capital humano especializado, costos de infraestructura, licenciamiento y sus comparativas, además de indicadores financieros que permitan determinar la rentabilidad del proyecto en un horizonte de mediano plazo, apuntando a los beneficios mencionados en el punto 3.2.

Es importante señalar que este análisis se realizó en base a los precios lista públicos, no a descuentos aplicados, los cuales se pueden revisar en el documento BOM_Kubernetes.xlsx en Anexos.

5.1 Supuestos de la evaluación económica

Con el objetivo de realizar una comparación homogénea entre ambas plataformas, la evaluación económica fue desarrollada considerando un escenario de referencia correspondiente a una organización que requiere migrar entre cuatro y cinco aplicaciones contenerizadas desde OpenShift hacia Oracle Kubernetes Engine, con una duración estimada del proyecto entre cuatro y seis semanas.

Los cálculos fueron realizados considerando los siguientes supuestos, infraestructura desplegada sobre Microsoft Azure, servicio Oracle Kubernetes Engine habilitado, valores de infraestructura y licenciamiento obtenidos desde precios de lista publicados por Oracle y Red Hat.

5.2 Costos económicos de OpenShift¹⁰

Es importante señalar que estos valores son referenciales y de precio lista, obtenidos de la página de Red Hat, bajo los supuestos que toda la infraestructura se encontraba desplegada en Azure y no existían volúmenes persistentes:

- Infraestructura:

¹⁰ Valores obtenidos desde Red Hat OpenShift: <https://www.redhat.com/es/technologies/cloud-computing/openshift/pricing>

- 3 Nodos Control Plane (Shape Dsv3, 8 vCPU, 32 GB de memoria RAM): USD 963,6
- 3 Nodos Data Plane (Shape Dsv3, 4 vCPU, 16 GB de memoria RAM): USD 481,8
- Licenciamiento:
 - Licencia Nodo 3 Data Plane: USD 374,4

Esto nos da un costo mensual de **USD 1.819,8** y un costo anual de **USD 21.837,6** para infraestructura y licenciamiento.

5.3 Costos económicos de OKE¹¹

Es importante señalar que estos valores son referenciales y de precio lista, obtenidos de Oracle, bajo el supuesto que no existía volúmenes persistentes:

- Infraestructura:
 - Kubernetes Engine: USD 74,4
 - 3 Nodos Data Plane (Shape E5.Flex, 4 vCPU, 32 GB de memoria RAM): USD 410,6

Esto nos da un costo mensual de **USD 491,4** y un costo anual de **USD 5.896,8** para infraestructura. A este monto se le debe sumar el capital humano, costo que se detallará en la siguiente sección.

5.4 Comparativa económica entre OpenShift y Oracle Kubernetes Engine (OKE)

En base al detalle de los costos revisados en las secciones 3.2.3 y 3.2.4, podemos comprar las costos a nivel de infraestructura y licenciamiento (este último solo en OpenShift).

Tabla 19: Comparativa de costos OpenShift y OKE

Concepto	OpenShift (Mensual)	OKE (Mensual)
Engine	USD 0	USD 74,4
Licencia 3 Nodo Control Plane	USD 0	USD 0
Infraestructura 3 Nodo Control Plane	USD 963,6	USD 0
Licencia Nodo 3 Data Plane	USD 374,4	USD 0
Infraestructura 3 Nodo Data Plane	USD 481,8	USD 410,6
Total Mensual	USD 1.819,8	USD 485,0
Total Anual	USD 21.838,68	USD 5.821,08

Si comparamos los costos asociados a OpenShift y a OKE podemos validar un ahorro anual de **16.017,60 USD**, lo que se transforma en un ahorro del 73,35%, solo a nivel de infraestructura y licenciamiento.

5.5 Capital humano requerido para la migración

La inversión inicial está compuesta principalmente por el esfuerzo del equipo técnico responsable de la migración. Los roles considerados, junto con su nivel de dedicación estimado, se presentan en la Tabla 1. La dedicación fue determinada a partir de la experiencia profesional del autor en proyectos de migración de plataformas Kubernetes y validada con prácticas habituales de proyectos de modernización tecnológica.

Tabla 20: Roles, tiempos y responsabilidades

Rol	Dedicación	Responsabilidad
-----	------------	-----------------

¹¹ Valores obtenidos desde Oracle Cost Estimator: <https://www.oracle.com/cloud/costestimator.html>

Arquitecto Cloud / Kubernetes	20–30%	Diseño, decisiones técnicas
Ingeniero Kubernetes / DevOps	50–100%	Migración, manifiestos, GitOps
Ingeniero Aplicaciones	20–40%	Ajustes de app, pruebas
Operaciones / SRE	10–20%	Monitoreo, cutover
Seguridad (ad-hoc)	Puntual	Validación de accesos y políticas

5.6 Estimación de tiempo por etapa

Las estimaciones fueron definidas a partir de la experiencia profesional del autor y el project manager de cliente en proyectos de infraestructura, plataformas Kubernetes y servicios cloud, complementadas con actividades comúnmente ejecutadas en procesos reales de migración, tales como levantamiento de información, análisis de compatibilidad, diseño de arquitectura, implementación, pruebas funcionales y de rendimiento, validación con usuarios y estabilización de la plataforma.

Debido a que la complejidad de las aplicaciones puede variar entre organizaciones, los tiempos se presentan como rangos estimados y no como valores absolutos.

Considerando un tiempo total de 4 a 6 semanas, se estiman los tiempos por etapas según la Tabla 21.

Tabla 21: Estimación de tiempos por etapas

Etapas	Horas estimadas
Etapas 1, Análisis entorno OpenShift	80–120
Etapas 2, Arquitectura y TO-BE OKE	40–60
Etapas 3, Despliegue entorno OKE	30–50
Etapas 4, Migración de aplicaciones	120–200
Etapas 5, Cutover y continuidad operativa	30–40
Etapas 6, Optimización y cierre	20–40
Total	320–510 horas

Se observa que la etapa de migración de aplicaciones concentra el mayor porcentaje del esfuerzo total del proyecto, debido a las actividades de adaptación, validación funcional, resolución de incompatibilidades y pruebas de rendimiento requeridas antes del paso a producción.

Según información recopilada en Tabla 21, los tiempos estimados son de 320 a 510 horas de trabajo, si consideramos un promedio de 20 USD por hora de trabajo (considerando solo las horas de trabajo relacionadas a la migración), podemos estimar el esfuerzo de la siguiente forma:

Tabla 22: Promedio de horas hombre. Fuente propia de empresa migración¹²

Concepto	Cantidad de personas	Valor HH	Total
320 Horas de migración	5	USD 20	USD 25.600
510 Horas de migración	5	USD 20	USD 40.800
415 Promedio de horas	5	USD 20	USD 33.200

¹² Importante, se debe señalar que la empresa donde se propuso e implementó esta estrategia solicitó no exponer su nombre, no obstante, todos los datos son reales

5.7 Evaluación financiera a nivel de infraestructura y capital humano

5.7.1 Retorno de la inversión (ROI)

ROI Anual, con relación al ahorro (USD 16.017,60) y la inversión promedio (USD 33.200)

$$ROI = \frac{16.017,60 - 33.200}{33.200} = -0.5175$$

El ROI dentro del primer año es de -51,75%, lo cual no se debe entender como un proyecto negativo o errado, este tipo de comportamientos es normal dentro de un proyecto de migración y transformación digital.

Esto se entiende debido a la inversión de migración realizada solo en el primer año, generando un ahorro recurrente y en un futuro, como se detalla en la sección 3.2.8.

5.7.2 Análisis del período de recuperación (Payback)

El *payback* permite conocer el momento en que se generará el retorno de la inversión

$$Payback = \frac{33.200}{16.017,60} = 2,7 \text{ años}$$

Por lo que podemos estimar que el retorno de la inversión se verá dentro de 3 años de la siguiente forma:

Tabla 22: Tabla de ahorros acumulados

Año	Ahorro acumulado (USD)	Balance neto (USD)	ROI acumulado
1	16.017,60	-17.182,40	-51,8 %
2	32.035,20	-1.164,80	-3,5 %
3	48.052,80	14.852,80	44,70%
4	64.070,40	30.870,40	93,00%
5	80.088,00	46.888,00	141,20%

Al comparar entre realizar la migración y no realizarla, podemos estimar que el costo acumulado en 3 años, considerando el ahorro de 16.017,60 anual es de 48.052,80 USD, una vez comprendidos la inversión y ahorros generados, podemos concluir que el análisis de retorno de inversión evidencia que la migración presenta un ROI negativo durante el primer año debido a la concentración de la inversión inicial en capital humano producto de la migración. No obstante, considerando los ahorros recurrentes anuales en infraestructura y licenciamiento, el proyecto alcanza su punto de equilibrio aproximadamente al finalizar el segundo año y comienza a generar beneficios económicos netos a partir del tercer año, justificando la iniciativa desde una perspectiva de mediano plazo.

6 Conclusiones

La presente tesina tuvo como propósito determinar si un proceso de migración desde Red Hat OpenShift hacia Oracle Kubernetes Engine (OKE) puede ser abordado mediante una metodología estructurada que permita reducir la incertidumbre técnica y operacional, preservar el rendimiento de las aplicaciones y generar beneficios económicos para las organizaciones. En este sentido, el principal aporte del trabajo no radica únicamente en analizar las diferencias técnicas y económicas entre ambas plataformas, sino en establecer un

procedimiento sistemático, repetible y medible que permita abordar este tipo de migraciones disminuyendo su dependencia de la experiencia individual de los equipos responsables.

A partir del análisis realizado y de las pruebas comparativas definidas, los resultados respaldan la hipótesis planteada dentro del alcance establecido para esta investigación. La comparación entre los escenarios AS-IS y TO-BE indica que una migración adecuadamente planificada puede mantener niveles de rendimiento equivalentes a los observados en la plataforma de origen, considerando variables como latencia, throughput, tasa de errores, escalabilidad y recuperación ante fallas. Al mismo tiempo, determinadas capacidades propias de un servicio Kubernetes administrado permiten simplificar tareas operacionales y obtener mejoras en aspectos específicos, como el escalado de nodos y la integración con servicios nativos de Oracle Cloud Infrastructure.

Desde el punto de vista económico, la migración hacia OKE requiere una inversión inicial significativa, concentrada principalmente en el esfuerzo de los profesionales involucrados en las etapas de análisis, planificación, implementación, pruebas y estabilización. Sin embargo, este esfuerzo se concentra principalmente durante el proceso de migración, mientras que los beneficios derivados de la reducción de costos de infraestructura, licenciamiento y operación se mantienen durante la utilización posterior de la plataforma. De acuerdo con el escenario económico evaluado, esta diferencia permite recuperar la inversión en un horizonte aproximado de tres años, aunque dicho período dependerá de factores como la cantidad y configuración de los nodos, el modelo de licenciamiento de OpenShift, la infraestructura utilizada y la complejidad de las aplicaciones migradas. Adicionalmente, este análisis económico no incorpora una valoración directa de beneficios como la reducción del riesgo operacional, la automatización de procesos, la simplificación de tareas administrativas y la mejora de las capacidades de escalabilidad. Estos elementos pueden generar beneficios adicionales para las organizaciones, aunque su impacto resulta más complejo de cuantificar mediante indicadores financieros tradicionales. Por este motivo, la evaluación económica de una migración no debiera limitarse únicamente a los costos del primer año, sino considerar un horizonte de mediano plazo que permita observar tanto la recuperación de la inversión como los beneficios acumulados.

Como se observa en la Tabla 23, el objetivo de la migración no consiste necesariamente en obtener mejoras significativas en todas las variables técnicas, sino en mantener un comportamiento equivalente al entorno OpenShift mientras se obtienen beneficios principalmente en los ámbitos operacional y económico. Se espera conservar niveles comparables de rendimiento, latencia, throughput, tasa de errores, escalado de pods y recuperación ante fallas, mientras que OKE presenta ventajas asociadas al escalado administrado de nodos y a menores costos operacionales. Estas mejoras requieren asumir un esfuerzo inicial mayor, cuyo beneficio se hace visible al evaluar la plataforma en un horizonte de mediano plazo.

Tabla 23: Comparativa resumen

Dimensión	Resultado esperado
Performance aplicación	Similar entre OpenShift y OKE
Latencia	Similar o levemente mejor en OKE
Throughput	Similar
Error rate	Similar
Escalado de pods	Similar
Escalado de nodos	Mejor en OKE
Recuperación ante falla	Similar
Costo operacional	Mejor en OKE
Esfuerzo inicial	Mayor en OKE
Beneficio económico	Visible en horizonte de 3 años

Uno de los hallazgos más relevantes de esta investigación es que el valor de una metodología de migración no se limita únicamente a la reducción de costos o tiempos de implementación. Su aporte reside también en disminuir el riesgo inherente a este tipo de proyectos mediante un proceso ordenado de evaluación, planificación, ejecución y validación. La definición previa de actividades, criterios de aceptación, mecanismos de rollback, pruebas y métricas permite reducir la incertidumbre, minimizar errores humanos y

disminuir la probabilidad de indisponibilidad de los servicios. De esta manera, la migración deja de ser una actividad altamente dependiente de la experiencia individual y se transforma en un proceso estructurado y reproducible.

Entre las principales contribuciones del trabajo se encuentra, por lo tanto, la integración de criterios técnicos, operacionales y económicos dentro de una misma metodología. La propuesta contempla el levantamiento y evaluación del entorno existente, el análisis de compatibilidad, el diseño de la arquitectura objetivo, la preparación de la plataforma, la migración de las aplicaciones, la validación funcional y operacional, el cutover y la optimización posterior. Esta estructura permite abordar la migración no solamente como un traslado tecnológico entre plataformas Kubernetes, sino como un proceso integral de transformación en el que deben considerarse simultáneamente arquitectura, operación, continuidad y costos.

Es importante señalar que los resultados deben interpretarse considerando las condiciones bajo las cuales fue desarrollada esta investigación. La metodología fue diseñada específicamente para migraciones desde Red Hat OpenShift hacia Oracle Kubernetes Engine, considerando aplicaciones contenerizadas compatibles con Kubernetes y una arquitectura de destino basada en Oracle Cloud Infrastructure. En consecuencia, aunque sus principios pueden servir como referencia para otros proyectos de modernización, su aplicación sobre otros proveedores cloud, distribuciones Kubernetes o aplicaciones con dependencias específicas de OpenShift requerirá adaptaciones. Asimismo, factores como la complejidad de las aplicaciones, el nivel de madurez de los equipos DevOps y las características particulares de cada organización pueden modificar los esfuerzos, costos y resultados obtenidos.

Como trabajo futuro, se propone validar la metodología mediante su aplicación en proyectos reales de distintas industrias y con cargas de trabajo de diferente complejidad. También resulta de interés profundizar la incorporación de herramientas de automatización como Ansible, estrategias GitOps y pipelines CI/CD para disminuir la intervención manual durante las distintas fases de la migración. Adicionalmente, futuras investigaciones podrían explorar el uso de técnicas de inteligencia artificial para apoyar el análisis de compatibilidad, la transformación de manifiestos Kubernetes y la estimación de riesgos, tiempos y costos, así como extender la metodología hacia escenarios híbridos y multicloud.

Finalmente, el desarrollo de esta investigación permitió evidenciar que el éxito de una iniciativa de modernización tecnológica no depende exclusivamente de las capacidades de la plataforma seleccionada. La planificación, la automatización, la gestión del riesgo y la existencia de una metodología estructurada son factores fundamentales para ejecutar el cambio de manera controlada. Desde esta perspectiva, el aporte de la investigación trasciende la comparación específica entre OpenShift y OKE, proponiendo un enfoque que puede servir como referencia para futuros procesos de transformación de plataformas Kubernetes, considerando de manera conjunta sus dimensiones técnicas, operacionales y económicas.

Agradecimientos. Rocío por todo su amor, paciencia y apoyo incondicional durante todo este proceso, su compañía y confianza, sin ti no estaría aquí, a nuestra hija Mía quién con su alegría, amor y sonrisas logra alegrar mi día a día y darme las fuerzas necesarias para seguir adelante y al profesor Raúl Monge por toda su ayuda y tiempo dedicado a este proyecto de tesis.

Referencias

- [1] Red Hat OpenShift - Red Hat 2026
<https://www.redhat.com/es/technologies/cloud-computing/openshift>
- [2] ¿Qué es Kubernetes?
<https://kubernetes.io/es/docs/concepts/overview/what-is-kubernetes/>
- [3] Visión general de Kubernetes Engine (OKE)
<https://docs.oracle.com/es-ww/iaas/Content/ContEng/Concepts/contengoverview.htm>
- [4] Red Hat OpenShift editions and pricing
<https://www.redhat.com/en/technologies/cloud-computing/openshift/pricing>
- [5] Redes en OpenShift
https://docs.redhat.com/en/documentation/openshift_container_platform/4.22/html/config_apis/ingress-config-openshift-io-v1
- [6] Migrate any OpenShift or Kubernetes workloads to OKE cloud using Migration as a Service
<https://blogs.oracle.com/developers/migrate-any-openshift-or-kubernetes-workloads-oke-cloud-using-migration-as-a-service>
- [7] Understanding Argo CD: Kubernetes GitOps Made Simple
<https://codefresh.io/learn/argo-cd/>
- [8] VMware, Top 5 Kubernetes Operations Challenges and How to Mitigate
<https://www.vmware.com/docs/white-paper-top-5-kubernetes-operations-challenges-and-how-to-mitigate>
- [9] OpenShift Builds
https://docs.redhat.com/en/documentation/openshift_container_platform/4.20/html/cicd_overview/ci-cd-overview
- [10] Que es GitOps
<https://www.gitops.tech/>
- [11] Setting Up an Nginx Ingress Controller on a Cluster
<https://docs.oracle.com/en-us/iaas/Content/ContEng/Tasks/contengsettingupingresscontroller.htm>
- [12] Ingress NGINX Retirement: What You Need to Know
<https://kubernetes.io/blog/2025/11/11/ingress-nginx-retirement/>

Anexos

A.1 Ejemplos YamRepositorios Git

Automatización con Ansible

- Git Global (todo lo utilizado para el despliegue de la infraestructura)
 - <https://github.com/whiplash0104/Tesina>
- Vault con todos los parámetros y variables necesarias para despliegue de Redes (desencriptado)
 - <https://github.com/whiplash0104/Tesina/blob/main/Ansible-VCN/vault.yaml>
- Creación de VCN
 - https://github.com/whiplash0104/Tesina/blob/main/Ansible-VCN/create_vcn.yaml
- Creación de Internet Gateway, asignada a VCN
 - https://github.com/whiplash0104/Tesina/blob/main/Ansible-VCN/ig_vcn.yaml
- Creación de Default Route
 - https://github.com/whiplash0104/Tesina/blob/main/Ansible-VCN/dr_vcn.yaml
- Creación de Subnets
 - https://github.com/whiplash0104/Tesina/blob/main/Ansible-VCN/subnet_vcn.yaml
- Creación de Reglas de ingress y egress
 - https://github.com/whiplash0104/Tesina/blob/main/Ansible-VCN/ingress_egress_vcn.yaml
- Creación cluster OKE
 - https://github.com/whiplash0104/Tesina/blob/main/Ansible-OKE-Cluster/create_oke_cluster.yaml
- Creación Kubeconfig
 - https://github.com/whiplash0104/Tesina/blob/main/Ansible-OKE-Cluster/create_kubeconfig_oke.yaml
- Vault con todos los parámetros y variables necesarias para despliegue de Cluster OKE (desencriptado)
 - <https://github.com/whiplash0104/Tesina/blob/main/Ansible-OKE-Cluster/vault.yaml>
- Automatización para iniciar instancias
 - https://github.com/whiplash0104/Ansible-OCI/blob/main/inicia_instancia.yaml
- Automatización para detener (apagar) instancias
 - https://github.com/whiplash0104/Ansible-OCI/blob/main/detiene_instancia.yaml
- Obtención de Inventario desde OCI (aplicable tanto para Ansible Core como para Oracle Linux Automation Manager)
 - <http://github.com/whiplash0104/OLAM-OCI-Inventario/blob/main/inventario.oci.yaml>
- BOM (Bill of Materials) reales de proyecto
 - https://github.com/whiplash0104/Tesina/blob/main/BOM_Kubernetes.xlsx

A.2 Yaml demos de Kubernetes

- Namespace
 - <https://github.com/whiplash0104/Tesina/blob/main/Kubernetes-yaml/namespace.yaml>
- Auth
 - <https://github.com/whiplash0104/Tesina/blob/main/Kubernetes-yaml/auth-service.yaml>
- Auth Service
 - <https://github.com/whiplash0104/Tesina/blob/main/Kubernetes-yaml/service-auth.yaml>
- Backend
 - <https://github.com/whiplash0104/Tesina/blob/main/Kubernetes-yaml/backend.yaml>
- Backend Service
 - <https://github.com/whiplash0104/Tesina/blob/main/Kubernetes-yaml/service-backend.yaml>
- Frontend
 - <https://github.com/whiplash0104/Tesina/blob/main/Kubernetes-yaml/frontend.yaml>
- Frontend Service
 - <https://github.com/whiplash0104/Tesina/blob/main/Kubernetes-yaml/service-frontend.yaml>
- Ingress Controller Frontend y Auth Service
 - <https://github.com/whiplash0104/Tesina/blob/main/Kubernetes-yaml/ingress-frontend-auth.yaml>
- Batch Job
 - <https://github.com/whiplash0104/Tesina/blob/main/Kubernetes-yaml/batch-job.yaml>