https://doi.org/10.22201/dgtic.30618096e.2025.3.3.117 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 61 - 80

Despliegue de OpenStack mediante Kolla-Ansible: una solución modular, automatizada y escalable para infraestructuras cloud

Información del reporte:

Para citar este reporte técnico:

Mares Mendoza, E. (2025). Despliegue de OpenStack mediante Kolla-Ansible: una solución modular, automatizada y escalable para infraestructuras cloud. *Cuadernos Técnicos Universitarios de la DGTIC, 3* (3) páginas(61 - 80).

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.117

Enrique Mares Mendoza

Dirección General de Cómputo y de Tecnologías de Información y Comunicación

Universidad Nacional Autónoma de México

enrique.mares@unam.mx

ORCID: 0009-0008-5529-0080

Resumen

Se analizó la posibilidad de implementar una infraestructura de nube, basada en software libre, en el Centro de Datos de la DGTIC de la UNAM, utilizando una solución que permitiera reducir costos operativos y dependencia de proveedores. Por lo tanto, se realizó una comparación de distintas herramientas de despliegue, para la que se consideraron criterios como facilidad de instalación, escalabilidad, automatización, mantenimiento y adecuación a entornos productivos. El análisis incluyó cinco enfoques distintos de despliegue, implementación manual y el uso de herramientas automatizadas reconocidas en la comunidad de *OpenStack*. Como resultado, se identificó que *Kolla-Ansible* ofreció un equilibrio entre complejidad y eficiencia, al integrar tecnologías de automatización y contenedores que permitieron una instalación modular, reproducible y segura. Si se compara con *OpenStack-Ansible*, TripleO o *DevStack*, demostró mayor estabilidad, menor carga de configuración inicial y facilidad para escalamiento de recursos. La prueba de despliegue con *Kolla-Ansible* validó su capacidad para reducir la intervención manual, minimizar errores humanos y agilizar los tiempos de implementación sin comprometer la disponibilidad del entorno. Se concluyó que, para infraestructuras orientadas a producción con requerimientos de escalabilidad, disponibilidad y gestión simplificada, esta herramienta

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.117 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 62 - 80

representa una solución robusta y estratégica que favorece la administración eficiente de servicios en la nube.

Palabras clave:

Openstack, Kolla-ansible, cloud, IaaS, Ansible.

Abstract

The feasibility of implementing an open-source cloud infrastructure at the UNAM's DGTIC Data Center was analyzed, utilizing a solution aimed at reducing operational costs and vendor dependency. Therefore, a comparison of various deployment tools was conducted, considering criteria such as ease of installation, scalability, automation, maintenance, and suitability for production environments. The analysis included five distinct deployment approaches, manual implementation and the use of automated tools recognized within the OpenStack community. As a result, Kolla-Ansible was identified as offering a balance between complexity and efficiency by integrating automation and container technologies, enabling a modular, reproducible, and secure installation. Compared to OpenStack-Ansible, TripleO, or DevStack, it demonstrated greater stability, lower initial configuration overhead, and easier resource scaling. The deployment test with Kolla-Ansible validated its ability to reduce manual intervention, minimize human errors, and streamline deployment times without compromising environment availability. It was concluded that for production-oriented infrastructures with scalability, availability, and simplified management requirements, this tool represents a robust and strategic solution that promotes efficient cloud service administration.

Keywords:

OpenStack, Kolla-Ansible, cloud, IaaS, Ansible.

1. INTRODUCCIÓN

Hoy en día, las tecnologías en la nube se han convertido en uno de los soportes más importantes de las infraestructuras tecnológicas modernas. *OpenStack* es ya una de las soluciones líderes para nubes privadas en este contexto. Gracias a su flexibilidad, escalabilidad y alto grado de personalización, es una alternativa muy útil para las entidades que buscan mejorar la optimización autónoma de la gestión de sus recursos tecnológicos. Bravo Roldán define *OpenStack* como una "plataforma de computación en la nube de código abierto para la creación tanto de nubes públicas como privadas que permite funcionalidad de Infraestructura como Servicio (IaaS)" (Bravo Roldán, L. A., 2022).

En este caso, *OpenStack* fue implementado en el Centro de Datos de la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC), a través de un proveedor externo con servicios de soporte, para garantizar el servicio que se ofrece a las dependencias de la UNAM. Sin embargo, los altos costos anuales relacionados con el soporte llevaron a considerar la implementación de una versión *open-source* de *OpenStack*.

Se realizó una investigación donde se identificó que la principal forma de instalación que se encuentra en la documentación de *OpenStack* es desplegándolo manualmente, lo cual es un proceso que consume mucho tiempo y requiere tanto un profundo conocimiento técnico como una atención meticulosa a

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.117 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 63 - 80

los detalles en la instalación. Se analizó la implementación de *OpenStack*, ya que requiere configurar manualmente cada servicio que conforma a *OpenStack* desde *Nova*, que es el servicio de cómputo de la nube responsable de administrar el ciclo de vida de las instancias; *Neutron,* encargado de la red y conectividad; *Cinder,* que se encarga de proporcionar almacenamiento en bloques para las instancias; y *Keystone,* que gestiona la autenticación autorización e identidad, por nombrar algunos, para lo cual es necesario saber cómo funcionan juntos. Además, la falta de automatización aumentó el riesgo de inconsistencias, lo que dificulta la escalabilidad y el mantenimiento posterior. La gestión manual de actualizaciones y compatibilidades entre versiones también representa un desafío importante, por los riesgos de generar interrupciones en los servicios.

Para solucionar este problema, se analizaron las herramientas de despliegues automatizadas así como se identificaron las ventajas y desventajas de estas herramientas: destacó la herramienta de *Kolla-ansible,* ya que cubre las necesidades del Centro de Datos de la DGTIC.

El objetivo de este estudio es verificar el uso de *Kolla-Ansible* como herramienta clave en la implementación de un clúster de *OpenStack*, destacando su capacidad para simplificar y automatizar el despliegue de infraestructuras de nube en entornos de producción. Se busca evaluar *Kolla-ansible*, su eficiencia en la configuración de componentes críticos, como redes, almacenamiento y alta disponibilidad, así como su adaptabilidad para escalar en comparación con las demás herramientas de despliegue.

2. DESARROLLO TÉCNICO

2.1 PROPUESTA DE SOLUCIÓN

El cambio técnico en la implementación de O*penStack* ha sido posible gracias a la innovadora fusión de dos tecnologías fundamentales en la construcción de *Kolla-Ansible*: *Docker* y *Ansible*. Esta combinación funciona como un sistema modular y eficiente; cada servicio de *OpenStack* está en su propio contenedor, ligero y replicable, gracias a *Docker*, y *Ansible* se encarga de desplegarlos y configurarlos mediante *scripts* declarativos. Esto no sólo facilita la instalación, sino que también la hace consistente y escalable en entornos de nube complicados.

Para explicar este concepto, se invita a imaginar un sistema en el que cada servicio es encapsulado, en este caso en un contenedor de *Docker,* diseñado para realizar una tarea específica de manera precisa, mientras que una herramienta centralizada como *Ansible* coordina el funcionamiento de todos los contenedores. Esto da lugar a una infraestructura eficiente y flexible, capaz de adaptarse a las necesidades cambiantes de las organizaciones modernas, alineándose con tendencias globales como GitOps, que es una metodología de gestión de infraestructura a través de Git y la Infraestructura como Código (IaC). Según la documentación oficial de *Kolla*, "La misión de *Kolla* es proporcionar contenedores listos para producción y herramientas de implementación para operar nubes *OpenStack*" (*OpenStack Foundation*, 2019).

2.2 METODOLOGÍA

La relevancia de analizar el impacto técnico de *Kolla-Ansible* surge cuando se compara con una instalación manual de cada servicio de *OpenStack* o con otras herramientas alternativas de implementación como TripleO, *OpenStack-Ansible* y *DevStack*. Como una plataforma de computación en la nube de *open-source*,

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.117 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 64 - 80

OpenStack necesita soluciones efectivas de instalación para ser utilizado en entornos de producción y desarrollo. Estas herramientas son críticas para simplificar la configuración, minimizar errores y acelerar los tiempos de implementación.

Como se indica en el estudio presentado por Gudipati y Mithra, "las herramientas del marco de gestión del ciclo de vida (LCMT) ofrecen al administrador de *OpenStack* una forma sencilla de implementar, configurar y gestionar la plataforma en la nube" (Gudipati, S. V., & Mithra, T. V., 2022), por lo que es necesario identificar las herramientas más apropiadas que puedan facilitar la administración de esta tecnología.

Para llevar a cabo este análisis, se utilizó una metodología comparativa, que se centra en examinar y contrastar diferentes alternativas según criterios específicos. Este enfoque permite identificar ventajas, desventajas y características particulares que hacen más apropiada cada herramienta dependiendo de las necesidades o demandas específicas. La metodología incluye la evaluación técnica de las herramientas, su adecuación a distintos contextos (producción, desarrollo, pruebas, etc.) y la presentación de una tabla comparativa que resume las diferencias más relevantes.

Se seleccionaron *Kolla-Ansible*, *OpenStack-Ansible*, TripleO, *DevStack* y la instalación manual como las principales formas de despliegue de *OpenStack*. Estas opciones fueron elegidas por su amplio uso en la comunidad, su implementación en entornos de desarrollo, prueba y producción, así como por contar con soporte y documentación oficial que permite una comparación fundamentada. Cada una ofrece diferentes niveles de automatización, complejidad y adecuación según el caso de uso. A continuación, se describen sus principales características.

2.3 KOLLA-ANSIBLE

Kolla-Ansible despliega los servicios de *OpenStack* mediante contenedores *Docker*. Según Vainio, "*Kolla* es un proyecto oficial de *OpenStack* para crear contenedores *Docker* listos para producción para los servicios de *OpenStack*" (Vainio, A., 2020), lo que respalda su relevancia y solidez como tecnología para gestionar imágenes de contenedores en entornos complejos. El uso de Docker permite a *Kolla-Ansible* una arquitectura modular, ya que cada servicio principal de *OpenStack* se ejecuta como un contenedor independiente, lo que permite instalar, reiniciar o actualizar cada componente sin afectar al resto del clúster.

Además, el elevado nivel de automatización que ofrece permite reducir considerablemente el tiempo necesario para un despliegue completo: mientras que una instalación manual de *OpenStack* puede requerir entre 40 y 60 horas-hombre, debido a la configuración detallada de numerosas dependencias y servicios, la implementación con *Kolla-Ansible* reduce este tiempo a un rango estimado de 8 a 12 horas, incluyendo validaciones y pruebas. Esta eficiencia se alcanza gracias a tareas automatizadas por Ansible, que permiten desplegar múltiples contenedores de forma simultánea.

Asimismo, la escalabilidad que proporciona facilita la expansión horizontal del clúster, ya que la incorporación de nodos de cómputo o control se realiza mediante simples actualizaciones del archivo de inventario y la ejecución de comandos específicos de *Ansible*, sin necesidad de reinstalar todo el entorno.

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.117 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 65 - 80

2.4 OPENSTACK-ANSIBLE

OpenStack-Ansible es una herramienta de implementación de *OpenStack* basado en *Ansible*. A diferencia de usar *Docker*, puede ejecutarse en contenedores basados en LXC o instalarse directamente en hardware físico (*bare-metal*). Es una solución adaptable y poderosa, ideal para entornos complejos que requieren un control detallado sobre la infraestructura.

Sin embargo, como señala Vainio, "es un método de implementación de bajo nivel y requiere un buen entendimiento del entorno objetivo y de *Ansible*" (Vainio, A., 2020), lo cual puede ser un obstáculo para los administradores menos experimentados. Aunque tiene una curva de aprendizaje compleja, su énfasis en configuraciones muy personalizables lo hace adecuado para necesidades más especializadas.

2.5 DEVSTACK

DevStack es una herramienta específicamente diseñada para implementaciones en entornos de desarrollo, pruebas y formación técnica. Permite poner en funcionamiento una instancia de *OpenStack* de manera rápida y sencilla, lo que la convierte en una opción ideal para la experimentación, el aprendizaje y el prototipado en escenarios controlados. Sin embargo, no debe considerarse una alternativa viable para entornos de producción, ya que no ofrece robustez, alta disponibilidad ni soporte para configuraciones avanzadas y escalables. Su arquitectura está orientada a instalaciones temporales de un solo nodo, sin los mecanismos necesarios para garantizar continuidad operativa.

2.6 TRIPLEO

OpenStack se utiliza como herramienta de implementación para desplegar *OpenStack* en sí mismo, esto se conoce como TripleO (*OpenStack* Sobre *OpenStack*), una solución integrada. Es una opción útil en entornos de producción más grandes y despliegues complejos, pero la configuración puede llevar bastante tiempo. TripleO es un instalador que, de manera única, permite gestionar infraestructuras escalables y altamente disponibles.

2.7 DESPLIEGUE MANUAL

En el despliegue manual, se debe instalar y configurar cada servicio de *OpenStack* individualmente. Se obtiene control total, pero es lento y propenso a errores, por lo que sólo debe usarse donde se necesite una configuración extremadamente personalizada o donde otras herramientas no funcionen.

A continuación, en la Tabla 1, se presenta una comparativa que muestra las principales diferencias entre las herramientas y el despliegue manual basándose en los criterios analizados, con el fin de facilitar la comprensión de las diferencias entre los distintos despliegues de *OpenStack*.

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.117 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 66 - 80

Tabla 1

Comparativa técnica de despliegue en OpenStack: automático y manual

Herramienta	Enfoque	Ventajas	Desventajas	Casos de Uso
Kolla-Ansible	Contenedores *Docker* + *Ansible*	Modularidad y escalabilidad Configuración predefinida Fácil integración con Ansible Reducción de conflictos entre dependencias	Requiere conocimientos previos en *Docker* y *Ansible*	Entornos de producción que requieren eficiencia, automatización y facilidad de mantenimiento
OpenStack-Ansible	Contenedores LXC + *Ansible*, Hardware + *Ansible*	Gran flexibilidad Automatización robusta	Gran complejidad Curva de aprendizaje pronunciada	Entornos de producción, desarrollo y pruebas
DevStack	Instalación ligera para desarrollo y pruebas	Rápida implementación Simplicidad para experimentar con OpenStack	No apto para producción Falta de robustez y soporte	Sólo utilizado para pruebas
TripleO	*OpenStack* sobre *OpenStack*	Infraestructura escalable Alta Disponibilidad Solución integrada	Configuración inicial compleja Complejidad operativa	Grandes entornos de producción con alta demanda de escalabilidad
Despliegue manual	Instalación manual de componentes	Control total sobre cada componente	Proceso lento y propenso a errores Altos costos de mantenimiento Demanda de experiencia	Casos de ajuste extremadamente personalizado o donde otras herramientas no sean viables

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.117 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 67 - 80

3. RESULTADOS

La elección de la forma de instalación depende directamente de las necesidades particulares de cada infraestructura. En el caso del Centro de Datos de la DGTIC, el enfoque está en implementar una nube privada que priorice flexibilidad, escalabilidad y automatización.

En este escenario, *Kolla-Ansible* destaca al ofrecer una instalación modular y eficiente en contenedores que utiliza significativamente menos tiempo para completarse que las implementaciones manuales de *OpenStack*; esta herramienta ofrece un punto óptimo entre la sobrecarga operativa y la sobrecarga de mantenimiento gracias a su uso de tecnologías de contenedores. Por el contrario, TripleO tiene una arquitectura para gestionar *OpenStack* más compleja y genera exceso operativo, mientras que *Kolla-Ansible* se beneficia de la simplicidad de usar *Docker* y minimiza los conflictos de dependencias a través de su uso de contenedores ligeros.

Por su parte, *OpenStack-Ansible* proporciona un buen nivel de automatización, actuando como una solución intermedia, aunque requiere dominar bastante *Ansible*, sumando una curva de aprendizaje compleja para el administrador. *DevStack*, por otro lado, está diseñado principalmente para entornos de desarrollo, lo que, si bien permite implementaciones rápidas, lo hace menos viable para operar en un entorno de producción debido a lo provisional que es y la falta de adaptabilidad para dichos entornos. Finalmente, la implementación manual permite el control completo de los servicios; la cantidad de esfuerzo técnico, tiempo y los costos de mantenimiento involucrados la hacen impráctica para implementar en producción.

Para evaluar si *Kolla-Ansible* cumple con los requerimientos como herramienta de implementación, se realizó la instalación de un entorno *OpenStack* compuesto por cinco nodos. Dos nodos se asignaron a tareas de control y los tres restantes se dispusieron para cómputo. Toda la información sobre el proceso de instalación, incluidas las etapas realizadas y configuraciones aplicadas, se encuentra documentada en los anexos para su consulta.

La implementación con *Kolla-Ansible* permitió automatizar eficientemente procesos complejos, como la configuración de nodos, generación de contraseñas seguras, creación de certificados TLS y el despliegue completo de servicios. Este enfoque no sólo simplificó el proceso al reducir drásticamente la intervención manual, sino que disminuyó significativamente los riesgos de errores humanos y proporcionó un ambiente más estable y funcional.

La configuración ofrecida por *Kolla-Ansible* está diseñada para garantizar tanto la operatividad actual como una expansión futura sin inconvenientes. Su arquitectura modular facilita la integración de nuevos nodos en el clúster, ya sean de control o cómputo, sin interrumpir el rendimiento del sistema existente. Además, esta estructura simplifica el crecimiento de la infraestructura mientras asegura un balance dinámico de la carga, y maximiza el uso eficiente de recursos en beneficio del rendimiento global del clúster.

4. CONCLUSIÓN

En general la implementación de *OpenStack* mediante *Kolla-Ansible* representa un avance significativo en la estandarización y automatización de los componentes que integran una infraestructura en la nube.

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.117 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 68 - 80

Este enfoque conduce no sólo a grandes ventajas técnicas (optimización de procesos, mayor gestión de entornos complejos), sino que también refleja la simplificación de operaciones y hace más accesibles tecnologías que eran consideradas complejas.

Además, el hecho de considerar a *Kolla-Ansible* como una opción para una futura implementación permitiría al Centro de Datos mantener un control total sobre la infraestructura, lo que contribuiría no sólo a una reducción significativa de los costos operativos y, en consecuencia, a un ahorro importante para la Universidad, sino también a fortalecer la autonomía técnica del Centro de Datos para gestionar y escalar su nube privada de forma independiente.

No obstante, como todas las soluciones tecnológicas avanzadas, su potencial total sólo se logra cuando se comprende a fondo tanto en teoría como en práctica. Sabemos que alcanzar buenos resultados requiere también una ejecución cuidadosa, diseñada para ajustarse a las características particulares y las demandas específicas de cada entorno de aplicación. Por esta razón, es fundamental llevar a cabo un análisis profundo y contar con un conocimiento sólido que permita guiar este proceso de manera efectiva.

A diferencia de los métodos tradicionales, que suelen ser más rígidos y estructurados, *Kolla-Ansible* se presenta como algo más que un conjunto de herramientas o guías. Este proyecto propone una forma de trabajo innovadora que busca cambiar la manera en que se gestiona la infraestructura en la nube. En un contexto donde la flexibilidad y la eficiencia son esenciales, *Kolla-Ansible* desafía las prácticas convencionales al ofrecer un enfoque que podría transformar la forma en que se diseñan y operan las plataformas, al promover el uso de contenedores como una solución clave.

REFERENCIAS

Bravo Roldán, L.A. (2022). *Implementación de la arquitectura de cloud computing OpenStack para el despliegue y disponibilidad de aplicaciones en la empresa DICONST* [Tesis de maestría, Universidad Tecnológica del Perú]). Repositorio Institucional UTP. https://hdl.handle.net/20.500.12867/6911

OpenStack Foundation. (2019). *Kolla-Ansible Documentation: Stein Release*. Recuperado el 2 de abril de 2025 de https://docs.openstack.org/kolla-ansible/stein/

Tatta Vishwa Mithra, & Gudipati Sai Vivek. (2022). *Investigation of an automatic deployment transformation method for OpenStack* [Tesis de maestría en Ingeniería Eléctrica con énfasis en Sistemas de Telecomunicaciones, Blekinge Institute of Technology]. DiVA Portal. https://www.diva-portal.org/smash/get/diva2:1674200/FULLTEXT02

Vainio, A. (2020). *Automated Software Configuration for Cloud Deployment* [Master's thesis, University of Helsinki]. Helda. https://helda.helsinki.fi/server/api/core/bitstreams/793f47f6-6cf6-47b6-a6be-7d2903550a7e/content

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.117 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 69 - 80

ANEXO A

Instalación de nube privada *OpenStack* con herramienta de despliegue *kolla-ansible*

La implementación de *OpenStack* utilizando *Kolla-Ansible* se posiciona como una alternativa estratégica para establecer una nube privada que sea robusta y escalable.

Esta solución emplea contenedores *Docker* para ejecutar los servicios de *OpenStack*, lo que simplifica tanto el proceso de implementación como el mantenimiento. Además, esta metodología proporciona un enfoque ágil y moderno para gestionar actualizaciones y el ciclo de vida de los servicios dentro de la infraestructura.

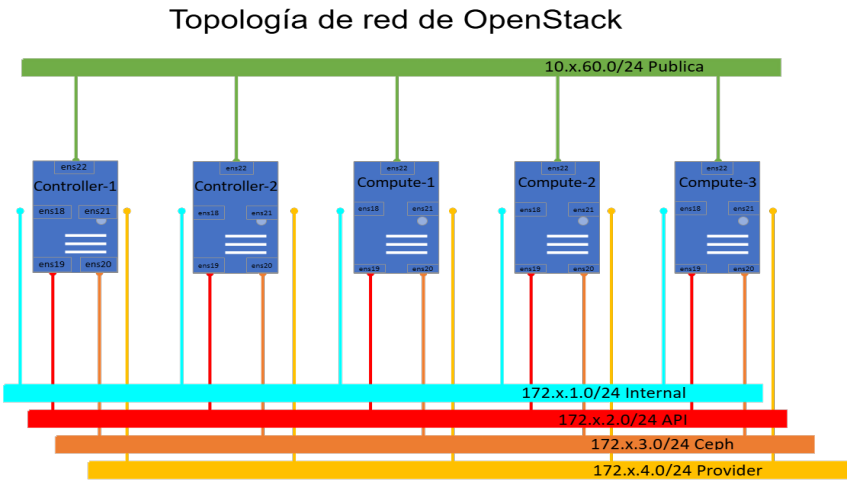
La base de este proyecto está centrada en un entorno virtualizado que funciona con Proxmox, donde se configuraron 5 máquinas virtuales dedicadas a nodos de control (2 *controller*) y nodos de cómputo (3 *compute*).

En el aspecto de red se definieron 5 redes. Red *Ceph* permitirá la comunicación entre nodos para la creación del almacenamiento de *Ceph*, la red Internal sirve para la comunicación entre los *backend* de los servicios de *OpenStack*, la red Pública es utilizado por los servicios de *OpenStack* para proporcionar acceso externo y garantizar alta disponibilidad mediante un mecanismo de *failover*, la red API se emplea por los servicios de *OpenStack* para sus APIs públicas, internas y de administración, y la red *Provider* es la que manejará el tráfico de red externo en el servicio de *Neutron*. Un esquema de cómo será la topología de red se muestra en la Figura 1, las direcciones IP incluyen una "x" para indicar que se utilizan únicamente como ejemplo.

Este enfoque basado en la virtualización ofrece una solución versátil que se adapta con facilidad a los requerimientos futuros. Al seguir el procedimiento detallado, se podrá implementar un sistema *OpenStack* de manera efectiva.

Figura 1

Topología del clúster para implementación de OpenStack



https://doi.org/10.22201/dgtic.30618096e.2025.3.3.117 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 70 - 80

Requisitos de Hardware

Para esta implementación, se utilizarán los siguientes requisitos para la creación de las máquinas virtuales:

Nodo	CPU	RAM	Almacenamiento	Almacenamiento	Almacenamiento	Red
Controller-1	4 núcleos	16 GB	50 GB	40 GB	40 GB	1 Gbps
Controller-2	4 núcleos	16 GB	50 GB	40 GB	40 GB	1 Gbps
Compute-1	4 núcleos	16 GB	50 GB	40 GB	40 GB	1 Gbps
Compute-2	4 núcleos	16 GB	50 GB	40 GB	40 GB	1 Gbps
Compute-3	4 núcleos	16 GB	50 GB	40 GB	40 GB	1 Gbps

1. INSTALACIÓN Y CONFIGURACIÓN DE HERRAMIENTAS Y DEPENDENCIAS

1.1 AGREGAR SERVIDORES AL ARCHIVO /ETC/HOSTS EN EL NODO DEPLOYER (CONTROLLER1)

El archivo /etc/hosts se utiliza para mapear direcciones IP a nombres de host dentro de la red local. Al agregar estos registros, se podrá identificar y comunicarse con los distintos nodos dentro del clúster *OpenStack*.

Editar /etc/hosts con la siguiente información:

```
### Lista de servidores
172.x.3.21 controller-1
172.x.3.22 controller-2
172.x.3.24 compute-1
172.x.3.25 compute-2
172.x.3.26 compute-3
```

1.2 CONFIGURACIÓN DE LA CLAVE SSH PARA EL USUARIO ROOT EN LOS NODOS

El uso de la clave SSH permite una autenticación segura sin necesidad de contraseñas, lo que facilita la comunicación entre los nodos.

Comandos:

Generar la clave SSH:

```
# ssh-keygen -t rsa
```

Copiar la clave pública a cada nodo:

```
# ssh-copy-id root@controller-1
```

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.117 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 71 - 80

```
# ssh-copy-id root@controller-2
# ssh-copy-id root@compute-1
# ssh-copy-id root@compute-2
# ssh-copy-id root@compute-3
```

1.3 CONFIGURACIÓN DEL NOMBRE DE HOST Y ZONA HORARIA EN TODOS LOS NODOS

Es crucial que todos los nodos tengan configurados nombres de host coherentes y la misma zona horaria para garantizar la consistencia en las operaciones del clúster.

Comandos:

```
for node in controller-{1..2} compute-{1..3}
do
  echo "--- $node ---"
  ssh root@$node hostnamectl set-hostname $node
  ssh root@$node timedatectl set-timezone America/Mexico_City
  echo ""
  sleep 2
done
```

1.4 INSTALACIÓN DE DOCKER CE EN CADA NODO

Docker se usa para facilitar la implementación de servicios y componentes en los nodos.

Comandos:

```
# apt-get install apt-transport-https \
   ca-certificates \
   curl \
   gnupg-agent \
   software-properties-common -y
#  curl  -fsSL  https://download.docker.com/linux/ubuntu/gpg  |  sudo  gpg
--dearmor > /etc/apt/trusted.gpg.d/docker-ce.gpg
# echo "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_
release -sc) stable" > /etc/apt/sources.list.d/docker-ce.list
# apt-get update; apt-get install docker-ce docker-ce-cli containerd.io -y
# systemctl enable --now Docker
```

2. INSTALACIÓN DE OPENSTACK CON KOLLA-ANSIBLE

2.1 AGREGAR SERVIDORES AL ARCHIVO /ETC/HOSTS EN CONTROLLER-1

El archivo /etc/hosts se utiliza para mapear direcciones IP a nombres de host en la red local. Al agregar estos registros, *Kolla-Ansible* podrá identificar y comunicarse con los distintos nodos dentro del clúster.

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.117 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 72 - 80

Editar /etc/hosts con la siguiente información:

```
172.x.1.21 controller-1 controller-1.internal.unam.mx
172.x.1.22 controller-2 controller-2.internal.unam.mx
172.x.1.24 compute-1 compute-1.internal.unam.mx
172.x.1.25 compute-2 compute-2.internal.unam.mx
172.x.1.26 compute-3 compute-3.internal.unam.mx
172.x.1.55 internal.unam.mx

10.x.60.21 controller-1.public.unam.mx
10.x.60.22 controller-2.public.unam.mx
10.x.60.24 compute-1.public.unam.mx
10.x.60.25 compute-2.public.unam.mx
10.x.60.26 compute-3.public.unam.mx
10.x.60.55 public.unam.mx
```

2.2 CONFIGURAR CLUSTER CEPH

Instalar y configurar un Clúster *Ceph* con Cephadm, se incorpora desde la instalación inicial en el nodo *deployer* (hosts y nodos), hasta el despliegue de OSDs para gestionar el almacenamiento de *Openstack*.

2.2.1 CREAR UN POOL PARA OPENSTACK

Comando:

```
# ceph osd pool create volumes
# ceph osd pool set volumes size 3
# rbd pool init volumes
# ceph osd pool create images
# ceph osd pool set images size 3
# rbd pool init images
# ceph osd pool create backups
# ceph osd pool set backups size 3
# rbd pool init backups
# ceph osd pool create vms
# ceph osd pool set vms size 3
# rbd pool init vms
```

2.2.2 CREAR CEPH KEYRING

```
# ceph auth get-or-create client.glance mon 'allow r' osd 'allow class-read object_prefix rbd_children, allow rwx pool=images' -o /etc/ceph/ceph.client.glance.keyring

# ceph auth get-or-create client.cinder mon 'allow r' osd 'allow class-read object_prefix rbd_children, allow rwx pool=volumes, allow rwx pool=images' -o /etc/ceph/ceph.client.cinder.keyring
```

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.117 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 73 - 80

```
# ceph auth get-or-create client.nova mon 'allow r' osd 'allow class-
read object_prefix rbd_children, allow rwx pool=vms, allow rx pool=ima-
ges' -o /etc/ceph/ceph.client.nova.keyring
# ceph auth get-or-create client.cinder-backup mon 'allow r' osd 'allow
class-read object_prefix rbd_children, allow rwx pool=backups' -o /etc/
ceph/ceph.client.cinder-backup.keyring
```

2.2.3 ACTUALIZACIÓN DEL SISTEMA Y CREACIÓN DEL ENTORNO VIRTUAL

El sistema debe estar actualizado para asegurar que se cuente con las últimas versiones de paquetes y dependencias necesarias para la instalación.

Comandos:

```
# apt-get update -y

# apt-get install python3-dev libffi-dev gcc libssl-dev python3-selinux
python3-setuptools python3-venv -y
```

2.2.4 ACTIVACIÓN DEL ENTORNO VIRTUAL

Es fundamental crear un entorno virtual en Python para aislar las dependencias específicas de *Kolla-Ansible* y *OpenStack*.

Comandos:

```
# python3 -m venv kolla-venv
# echo "source ~/kolla-venv/bin/activate" >> ~/.bashrc
# source ~/kolla-venv/bin/actívate
```

2.2.5 INSTALACIÓN DE ANSIBLE Y KOLLA-ANSIBLE

Kolla-Ansible depende de *Ansible* para automatizar la instalación y configuración de *OpenStack*. Debemos instalar las versiones adecuadas de estas herramientas.

Comandos:

```
# pip install -U pip
# pip install 'ansible-core>=2.14,<2.16'
# ansible --version
# pip install git+https://opendev.org/openstack/kolla-ansible@master
```

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.117 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 74 - 80

2.3 CONFIGURACIÓN DE INVENTARIO DE MÚLTIPLES NODOS

2.3.1 MODIFICACIÓN DEL ARCHIVO DE INVENTARIO MULTINODE

El archivo *multinode* define los nodos y sus roles dentro del clúster de *OpenStack*. Se debe comprobar que todos los nodos estén correctamente definidos.

Comandos:

```
# cp multinode multinode.bak
# nano multinode
```

Contenido de *multinode*:

```
[control]
controller-1.internal.unam.mx ansible_host=172.x.1.21
controller-2.internal.unam.mx ansible_host=172.x.1.22

[network]
controller-1.internal.unam.mx
controller-2.internal.unam.mx

[compute]
controller-1.internal.unam.mx
controller-2.internal.unam.mx
compute-1.internal.unam.mx ansible_host=172.x.1.24
compute-2.internal.unam.mx ansible_host=172.x.1.25
compute-3.internal.unam.mx ansible_host=172.x.1.26

[monitoring]
controller-1.internal.unam.mx
controller-2.internal.unam.mx

[storage]
controller-1.internal.unam.mx
controller-2.internal.unam.mx
compute-1.internal.unam.mx
compute-2.internal.unam.mx
compute-3.internal.unam.mx

[deployment]
localhost       ansible_connection=local
```

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.117 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 75 - 80

2.3.2 VERIFICACIÓN DE CONEXIÓN CON ANSIBLE

Es importante verificar que los nodos sean accesibles mediante *Ansible* antes de proceder con la implementación.

Comando:

```
# ansible -i multinode all -m ping
```

2.4 GENERACIÓN DE CONTRASEÑAS Y CERTIFICADOS TLS

2.4.1 GENERACIÓN DE CONTRASEÑAS

Para garantizar la seguridad de los servicios de *OpenStack*, es necesario generar contraseñas aleatorias para cada uno de los servicios.

Comando:

```
# kolla-genpwd
```

2.4.2 GENERACIÓN DE CERTIFICADOS TLS

Es importante generar los certificados TLS para garantizar las comunicaciones seguras entre los nodos de *OpenStack*.

Comando:

```
# kolla-ansible certificates -i multinode
```

2.4.3 GENERACIÓN DE CLAVE PRIVADA

Es importante generar una clave privada para generar posteriormente un certificado autofirmado.

Comando:

```
# openssl genpkey -algorithm RSA -out /etc/ssl/private/private.key -aes256
```

2.4.4 GENERAR UN CERTIFICADO AUTOFIRMADO

Este tipo de certificado es útil para configurar el *HAProxy* que permite el direccionamiento de los servicios de *OpenStack*

Comando:

```
# openssl req -new -x509 -key /etc/ssl/private/private.key -out /etc/ssl/
private/haproxy-public.pem -days 365
# openssl req -new -x509 -key /etc/ssl/private/private.key -out /etc/ssl/
private/haproxy-internal.pem -days 365
```

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.117 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 76 - 80

2.5 REACIÓN DE DIRECTORIOS DE CONFIGURACIÓN DE KOLLA-ANSIBLE

Kolla-Ansible necesita directorios específicos para almacenar las configuraciones de los servicios de *OpenStack* como Nova, *Glance* y *Cinder*.

Comandos:

```
# mkdir -p /etc/kolla/config
# mkdir -p /etc/kolla/config/nova
# mkdir -p /etc/kolla/config/glance
# mkdir -p /etc/kolla/config/cinder/cinder-volume
# mkdir -p /etc/kolla/config/cinder/cinder-backup
# cp /etc/ceph/ceph.conf /etc/kolla/config/cinder/
# cp /etc/ceph/ceph.conf /etc/kolla/config/nova/
# cp /etc/ceph/ceph.conf /etc/kolla/config/glance/
# cp /etc/ceph/ceph.client.glance.keyring /etc/kolla/config/glance/
# cp /etc/ceph/ceph.client.nova.keyring /etc/kolla/config/nova/
# cp /etc/ceph/ceph.client.cinder.keyring /etc/kolla/config/nova/
# cp /etc/ceph/ceph.client.cinder.keyring /etc/kolla/config/cinder/cin-
der-volume/
# cp /etc/ceph/ceph.client.cinder.keyring /etc/kolla/config/cinder/cin-
der-backup/
# cp /etc/ceph/ceph.client.cinder-backup.keyring /etc/kolla/config/cin-
der/cinder-backup/

for node in controller-{2} compute-{1..3}
do
  scp -r /etc/ceph/ root@$node:/etc/
done
```

2.6 EDICIÓN DEL ARCHIVO GLOBALS.YML

El archivo globals.yml es el archivo central para la configuración de *Kolla-Ansible* y define parámetros clave como el uso de TLS, la distribución base y las direcciones IP.

Comando:

```
nano /etc/kolla/globals.yml
```

Configuraciones relevantes:

```
---
workaround_ansible_issue_8743: yes
kolla_base_distro: "ubuntu"
openstack_release: "master"
kolla_internal_vip_address: "172.x.2.55"
kolla_internal_fqdn: "internal.unam.mx"
kolla_external_vip_address: "10.x.60.55"
kolla_external_fqdn: "public.unam.mx"
kolla_external_vip_interface: "ens22"
```

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.117 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 77 - 80

```
api_interface: "ens19"
tunnel_interface: "ens20"
neutron_external_interface: "ens21"
neutron_plugin_agent: "ovn"
kolla_enable_tls_internal: "yes"
kolla_enable_tls_external: "yes"
kolla_copy_ca_into_containers: "yes"
kolla_external_fqdn_cert: "/etc/ssl/private/haproxy-public.pem"
kolla_external_fqdn_key: "/etc/ssl/private/haproxy-public.pem"
kolla_internal_fqdn_cert: "/etc/ssl/private/haproxy-internal.pem"
kolla_internal_fqdn_key: "/etc/ssl/private/haproxy-internal.pem"
openstack_cacert: "/etc/ssl/certs/ca-certificates.crt"
kolla_enable_tls_backend: yes
enable_openstack_core: "yes"
enable_cinder: "yes"
enable_fluentd: "yes"
enable_neutron_provider_networks: "yes"
cinder_cluster_name: "cinder-cluster"
ceph_glance_user: "glance"
ceph_glance_keyring: "client.glance.keyring"
ceph_glance_pool_name: "images"
ceph_cinder_user: "cinder"
ceph_cinder_keyring: "client.cinder.keyring"
ceph_cinder_pool_name: "volumes"
ceph_cinder_backup_user: "cinder-backup"
ceph_cinder_backup_keyring: "client.cinder-backup.keyring"
ceph_cinder_backup_pool_name: "backups"
ceph_nova_user: "nova"
ceph_nova_keyring: "client.nova.keyring"
ceph_nova_pool_name: "vms"
glance_backend_ceph: "yes"
cinder_backend_ceph: "yes"
nova_backend_ceph: "yes"
nova_compute_virt_type: "kvm"
neutron_ovn_distributed_fip: "yes"
```

2.7 FASE DE IMPLEMENTACIÓN DE OPENSTACK

2.7.1 CONFIGURACIÓN INICIAL DE LOS SERVIDORES

Es necesario configurar los servidores antes de la instalación de *OpenStack*.

Comando:

```
# kolla-ansible bootstrap-servers -i multinode
```

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.117 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 78 - 80

2.7.2 VERIFICACIÓN PREVIA Y DESPLIEGUE

Antes de desplegar *OpenStack*, se deben realizar verificaciones para asegurar que todo esté en orden.

Comandos:

```
# kolla-ansible prechecks -i multinode
# kolla-ansible deploy -i multinode
```

2.7.3 POST-DESPLIEGUE

Una vez que *OpenStack* está desplegado, se deben realizar pasos adicionales.

Comando:

```
# kolla-ansible post-deploy -i multinode
```

2.8 CONFIGURACIÓN DEL CLIENTE OPENSTACK

2.8.1 INSTALACIÓN DEL CLIENTE OPENSTACK

Para interactuar con *OpenStack* desde la línea de comandos, es necesario instalar el cliente *OpenStack*.

Comandos:

```
# cat /etc/kolla/certificates/ca/root.crt | sudo tee -a /etc/ssl/certs/
ca-certificates.crt
# pip3 install python-openstackclient
```

2.8.2 VERIFICACIÓN DEL CLIENTE

Verifica la instalación del cliente de *OpenStack*.

Comando:

```
# openstack –versión
```

2.8.3 VERIFICACIÓN DEL CLÚSTER DE OPENSTACK

Verificación de los servicios en ejecución para el funcionamiento del clúster de *OpenStack*.

```
# openstack endpoint list
# openstack service list
# openstack compute service list;
# openstack network agent list
# openstack volume service list
```

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.117 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 79 - 80

2.9 AGREGAR NODOS ADICIONALES

2.9.1 AGREGAR LA IP Y NOMBRE DEL HOST AL INVENTARIO

En el archivo *multinode* define el nodo nuevo y sus roles dentro del clúster de *OpenStack*. Para consultar los detalles del archivo *multinode*, se recomienda consultar la sección 3.5.1 del Manual de instalación de *OpenStack*, utilizando *Kolla-Ansible*. Para el ejemplo, el nodo se nombrará controller-3.internal.unam.mx

Comandos:

```
# cp multinode multinode.bak
# nano multinode
```

Contenido de multinode:

```
[control]
controller-1.internal.unam.mx ansible_host=172.x.1.21
controller-2.internal.unam.mx ansible_host=172.x.1.22
controller-3.internal.unam.mx ansible_host=172.x.1.23

[network]
controller-1.internal.unam.mx
controller-2.internal.unam.mx
controller-3.internal.unam.mx

[compute]
controller-1.internal.unam.mx
controller-2.internal.unam.mx
controller-3.internal.unam.mx
compute-1.internal.unam.mx ansible_host=172.x.1.24
compute-2.internal.unam.mx ansible_host=172.x.1.25
compute-3.internal.unam.mx ansible_host=172.x.1.26

[monitoring]
controller-1.internal.unam.mx
controller-2.internal.unam.mx
controller-3.internal.unam.mx

[storage]
controller-1.internal.unam.mx
controller-2.internal.unam.mx
controller-3.internal.unam.mx
compute-1.internal.unam.mx
compute-2.internal.unam.mx
compute-3.internal.unam.mx

[deployment]
localhost       ansible_connection=local
```

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.117 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 80 - 80

2.9.2 VERIFICACIÓN DE CONEXIÓN CON ANSIBLE

Comando:

```
# ansible -i multinode controller-3.internal.unam.mx -m ping
```

2.9.3 CONFIGURACIÓN INICIAL DE LOS SERVIDORES

Es necesario configurar los servidores antes de la instalación de *OpenStack*, colocando las dependencias necesarias.

Comando:

```
# kolla-ansible bootstrap-servers -i multinode --limit controller-3.in-
ternal.unam.mx
```

2.9.4 DESCARGAR LAS IMÁGENES DE CONTENEDORES

Se descargan las imágenes de contenedores necesarias para el nodo *controller* con el siguiente comando:

Comando:

```
# kolla-ansible pull -i multinode --limit controller-3.internal.unam.mx
```

2.9.5 DESPLIEGUE DE OPENSTACK EN EL NODO CONTROLLER-3

Comando:

```
# kolla-ansible deploy -i multinode --limit control
```

2.9.6 VERIFICACIÓN DE NODO AGREGADO

Comando:

```
# openstack host list
```