

# **ESCUELA POLITÉCNICA NACIONAL**

**ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

**IMPLEMENTACIÓN DE *CLOUD COMPUTING* UTILIZANDO  
HERRAMIENTAS *OPEN SOURCE***

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR  
EN REDES Y TELECOMUNICACIONES**

**STEVEN MAURICIO HERRERA HERNÁNDEZ**

**DIRECTOR: FERNANDO VINICIO BECERRA CAMACHO**

**DMQ, FEBRERO 2022**

## CERTIFICACIONES

Yo, STEVEN MAURICIO HERRERA HERNÁNDEZ declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



---

**STEVEN MAURICIO HERRERA HERNÁNDEZ**

**Steven.herrera01@epn.edu.ec**

**Steven2001herrera@gmail.com**

Certifico que el presente trabajo de integración curricular fue desarrollado por STEVEN MAURICIO HERRERA HERNÁNDEZ, bajo mi supervisión.



---

**FERNANDO VINICIO BECERRA CAMACHO**

**DIRECTOR**

**Fernando.becerrac@epn.edu.ec**

## **DECLARACIÓN DE AUTORÍA**

A través de la presente declaración, afirmo que el trabajo de integración curricular aquí descrito, así como el producto resultante del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

STEVEN MAURICIO HERRERA HERNÁNDEZ

## **DEDICATORIA**

A toda mi familia, pues todos me han apoyado y me han brindado su ayuda durante mi carrera y durante toda mi vida.

De manera especial quiero dedicar este trabajo en memoria de mi abuelito Galo Herrera quien siempre quiso ver a uno de sus nietos ser una persona profesional, siempre me animaba a seguir adelante, sé que estará muy orgulloso por este logro.

## **AGRADECIMIENTO**

En primer lugar, agradezco a Dios por llenarme de sabiduría en todo este trayecto, de la misma manera por guiar mis pasos.

A mis padres, por todos sus consejos, sus conocimientos que me han servido para ser una persona de bien, su apoyo incondicional, por toda la confianza que me han dado, por su amor y sus enseñanzas; gracias por su esfuerzo durante esta etapa de mi vida y gracias por enseñarme a nunca rendirme.

A mi hermano quien siempre me ha apoyado en todas las decisiones de mi vida y ha estado junto a mi ayudándome.

A toda mi familia por todo el respaldo y la ayuda que me han dado durante mi carrera y en general en mi vida.

Agradezco al Ing. Fernando Becerra por toda la ayuda brindada, por compartir su conocimiento, por la paciencia que me tuvo y de la misma manera por el apoyo que me dio en la realización de este trabajo.

Agradezco a mis amigos por toda la ayuda brindada y por siempre apoyarme en los momentos difíciles de mi vida.

# ÍNDICE DE CONTENIDOS

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA .....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
RESUMEN.....	VI
ABSTRACT.....	VII
1 Descripción del componente desarrollado .....	1
1.1 Objetivo general.....	1
1.2 Objetivos específicos.....	2
1.3 Alcance .....	2
1.4 Marco Teórico .....	2
2 Metodología.....	5
3 Resultados.....	6
3.1 Investigación sobre <i>Cloud Computing</i> , sus modelos de servicio y sus diferentes plataformas. ....	6
3.2 Determinación de una <i>Cloud Computing</i> que utilice <i>software open source</i> para su implementación.....	15
3.3 Implementación de una <i>Cloud Computing</i> sobre una máquina virtual.....	16
3.4 Pruebas de funcionamiento de la <i>Cloud Computing</i> .....	23
4 Conclusiones.....	37
5 Recomendaciones .....	38
6 Referencias Bibliográficas .....	39
7 ANEXOS.....	41
ANEXO I.....	41
ANEXO II .....	42

## RESUMEN

El presente proyecto de titulación, tiene como finalidad la investigación y el estudio de *Cloud Computing* y de al menos tres plataformas que brindan este servicio en la actualidad. Además de la selección de una *Cloud* para su implementación.

Con la *Cloud Computing* implementada se debe acceder desde un *browser* que se encuentre en la red para poder desarrollar servicios tales como: creación de instancias de máquinas virtuales, acceso a la plataforma y la gestión de red en la nube con varias instancias de máquinas virtuales.

A continuación, se detalla la estructura del presente trabajo, el cual va a poseer 5 capítulos que se dividen de la siguiente manera.

El capítulo uno contiene una descripción en donde se expone el contexto del proyecto, objetivo general, objetivos específicos, el alcance del proyecto y el marco teórico.

El capítulo dos va a contemplar la metodología aplicada para el desarrollo del proyecto de titulación.

El capítulo tres va a consistir en los resultados, aquí se va a presentar la investigación sobre *Cloud Computing*, algunas plataformas y la selección de una plataforma para ser implementada. Con la *Cloud Computing* seleccionada se va a presentar su instalación, la creación de redes, *routers* e instancias de máquinas virtuales dentro de la plataforma.

Y finalmente en el capítulo cuatro y cinco se encuentran las conclusiones y recomendaciones respectivamente, las cuales fueron obtenidas durante el desarrollo del presente trabajo de titulación.

**PALABRAS CLAVE:** red, virtualización, OpenStack, *cloud*, instancias, servicios en la nube

## **ABSTRACT**

The present degree project is to research and study cloud computing and at least three cloud computing platforms that are used today. In addition to the selection of a cloud for its implementation.

With the implemented cloud computing must be accessed from a browser located on the network to develop services such as: creation of virtual machine instances, access to the platform and network management in the cloud with several virtual machine instances.

Next, the structure of the present work is detailed, which is going to have 5 chapters that are divided as follows.

Chapter one will contain a description which will be exposed the context of the project, general objective, specific objectives, the scope of the project and the theoretical framework.

Chapter two will cover the methodology applied for the development of the degree project.

Chapter three will consist of the results, here will be presented the research on cloud computing, some platforms and the selection of a platform to be implemented. With the cloud computing selected will be presented its installation, the creation of networks, creation of routers and virtual machines instances within the platform.

And finally, in chapter four and five are the conclusions and recommendations respectively, which were obtained during the development of this degree work.

**KEYWORDS:** network, virtualization, OpenStack, cloud, instance, cloud services



# 1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El presente trabajo de titulación en primer lugar abarca una investigación sobre *Cloud Computing*, en donde se puede detallar ciertos aspectos como el concepto, el funcionamiento, características esenciales, por otra parte, se define los diferentes tipos de nubes que se pueden implementar. De igual manera se presentan los tres modelos de servicio con sus respectivos ejemplos los cuales pueden ayudar a diferenciar las plataformas que existen debido a que estas poseen cierto control dependiendo del modelo que utilicen.

Una vez comprendido *Cloud Computing* y sus aspectos más importantes, se presenta la investigación de tres plataformas las cuales son: OpenStack, OpenNebula y Google Cloud, de estas tres plataformas se define su concepto, sus principales características, sus módulos o componentes los cuales son necesarios para que funcionen de manera correcta. Después de haber presentado dichas plataformas, se determina una plataforma de *Cloud Computing* que sea *software* de código abierto para ser implementada.

Después se expone la plataforma elegida para su implementación y sobre todo se brindan las razones por las cuales fue designada comparándola con las otras dos plataformas.

Una vez finalizada la investigación y la selección de *Cloud Computing*, se procede a su implementación en donde se detalla paso a paso la instalación de la plataforma dentro de una máquina virtual, y es por ello que previamente se verifica que la virtualización dentro del computador esté habilitada.

Después de que haya concluido el proceso de instalación se procede con las pruebas de funcionamiento de la *Cloud Computing* implementada, para lo cual se accede a la interfaz gráfica de la plataforma y en esta parte se presenta detalladamente la creación de redes, también la implementación de *routers* y sobre todo el establecimiento de instancias de máquinas virtuales.

## 1.1 Objetivo general

Desplegar un prototipo de *Cloud Computing* mediante *software Open Source*.

## 1.2 Objetivos específicos

1. Investigar sobre *Cloud Computing*, sus modelos de servicio y sus diferentes plataformas.
2. Determinar una *Cloud Computing* que utilice *software open source* para su implementación.
3. Implementar una *Cloud Computing* sobre una máquina virtual.
4. Realizar pruebas de funcionamiento de la *Cloud Computing* implementada.

## 1.3 Alcance

El presente proyecto de titulación tiene como alcance el estudio de *Cloud Computing*, sus características, sus tipos de nubes y sus modelos de servicio. Además, se investigará al menos tres plataformas de *Cloud Computing* que se utilizan en la actualidad, se especificará sus características y los componentes que cada uno de estos posee.

También se seleccionará la mejor opción de *Cloud Computing* que utilicen *open source* para ser implementada sobre una máquina virtual. Para ello se debe verificar que esté operativa la virtualización en el *host* y en el caso de que no esté activa, se deberá habilitar dicha opción.

Una vez habilitada la virtualización e implementada la *Cloud Computing* se debe desarrollar servicios como el acceso a la plataforma mediante una interfaz gráfica el cual es un componente de dicha *cloud*. La creación de instancias de máquinas virtuales y la implementación de una red de datos con instancias de máquinas virtuales la cual puede ser gestionada desde un *browser*.

## 1.4 Marco Teórico

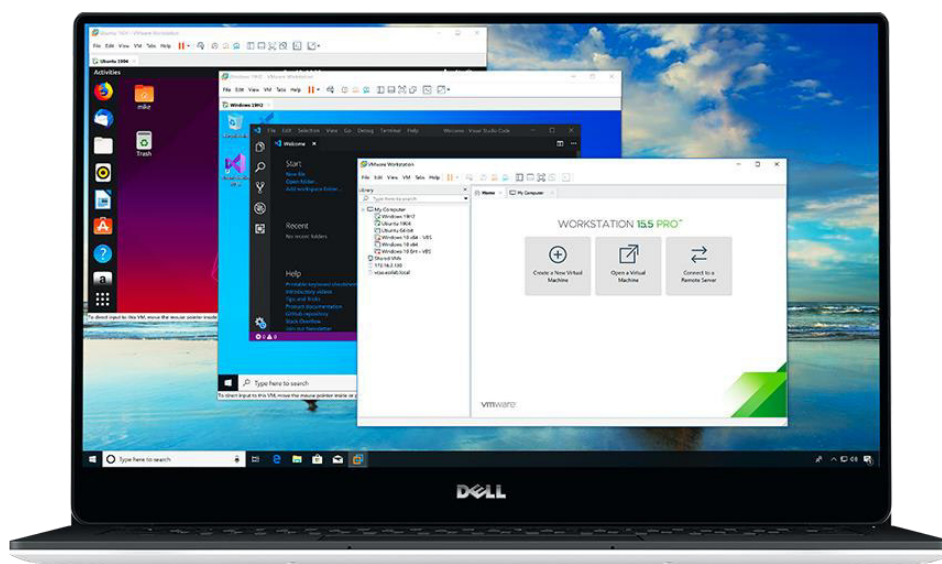
### Virtualización

La virtualización es un concepto que comenzó a implementarse en los años 60 mediante la empresa IBM, ya que intentaban mejorar el uso del CPU con la intención de poder realizar diversas tareas de forma simultánea. Esto optimizó costos, pues era más económico tener un solo equipo y dividir los recursos para las diferentes tareas que tener varios equipos con una sola tarea en específico. Es así que este concepto hace referencia a la creación de un entorno virtual de un sistema operativo con los recursos que están relacionados al *hardware*; esto quiere decir que se utiliza los recursos de la

máquina física (RAM, CPU, Disco Duro) para que sean ocupados dentro de una máquina virtual [1].

El funcionamiento de la virtualización está basado en dos elementos de suma importancia que son: la máquina virtual y el hipervisor, a continuación, se define cada uno de estos componentes.

La máquina virtual es un entorno virtual el cual es creado por un *software*, estas máquinas pueden ejecutar sistemas operativos como por ejemplo Windows, Linux y Mac OS, y sus recursos son entregados por la máquina anfitriona. El hipervisor es un proceso que permite la virtualización en un *host* determinado mediante el manejo compartido de recursos de cómputo, es decir que crea las máquinas virtuales asignando los recursos que esta necesita con ayuda de los recursos que tiene el *hardware* [2]. Gracias a un hipervisor se puede tener en funcionamiento varios sistemas operativos tal como se representa en la Figura 1.1.



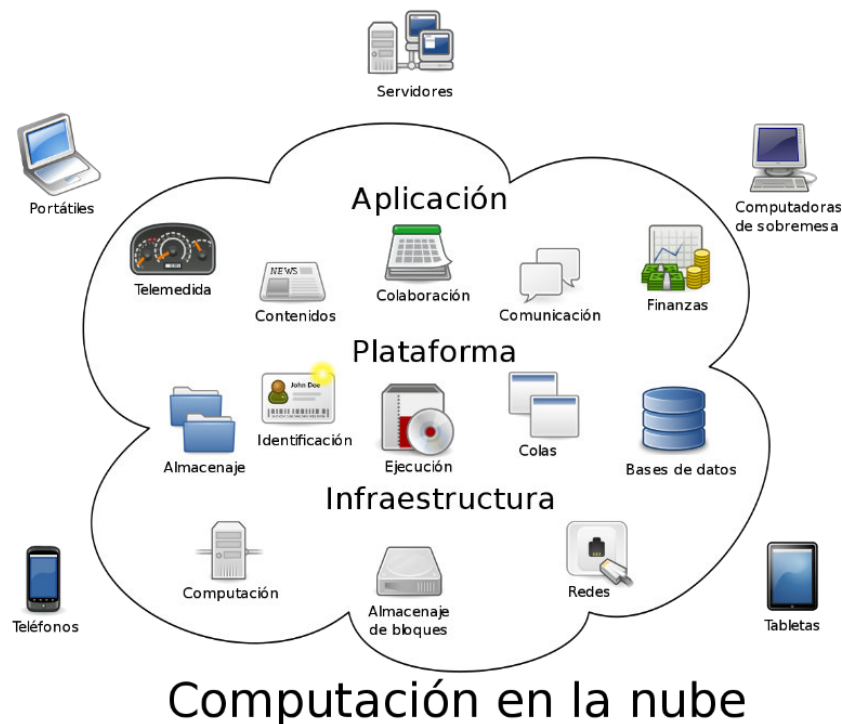
**Figura 1.1** Virtualización de Sistemas Operativos [1]

### ***Cloud Computing***

Mediante el desarrollo tecnológico y el avance de la creación de redes, *hardware* y *software* informático han hecho que la computación en la nube sea una realidad. Cuando se habla de computación en la nube, se refiere a un modelo tecnológico el cual puede ofrecer varios servicios por medio de Internet. La primera *Cloud Computing* accesible

para el público fue creada por Amazon en el año 2006 y se denominó Amazon Web Service (AWS) [3].

Hoy en día existen múltiples proveedores de *Cloud Computing* ya sean de paga o gratuitos y estos ofrecen gran variedad de servicios dependiendo las necesidades de los usuarios, es por ello la necesidad de realizar un estudio de algunas plataformas de *Cloud Computing* como OpenStack, OpenNebula y Google Cloud puesto que estas plataformas son utilizadas en la actualidad, además de ser herramientas de código abierto. En la Figura 1.2 se puede visualizar como es el funcionamiento de *Cloud Computing*.



**Figura 1.2** Computación en la nube

## 2 METODOLOGÍA

Para iniciar se investigó información acerca de *Cloud Computing*, además se detalló sus características, tipos de nubes y modelos de servicio en páginas web, artículos de revista y cualquier otro medio de investigación; con ello se obtuvo el conocimiento previo del funcionamiento de las *Cloud Computing*. Además, se investigó varias plataformas de *Cloud Computing* para entender cómo funcionan.

Al tener varias plataformas de *Cloud Computing* que utilicen *software open source*, se realizó una comparación de algunas plataformas. De esta manera se seleccionó una *cloud* la cual debe ser implementada para así entender más a detalle su funcionamiento.

Una vez determinada la *Cloud Computing*, se procede a la implementación de la misma, para lo cual se utilizó un hipervisor que permita al usuario ejecutar máquinas virtuales y en dicha máquina se implementó la *Cloud Computing*. Una vez instalado se procede a entender el funcionamiento de ciertas opciones que posee la plataforma para desarrollar las pruebas.

Para las pruebas de funcionamiento de la *Cloud Computing*, se inició con la parte de *networking* por lo que se comenzó con la implementación de redes. Además, se realizó la creación de instancias de máquinas virtuales las cuales van a estar conectadas a las subredes creadas.

### 3 RESULTADOS

En primer lugar, se presenta la investigación de *Cloud Computing* y sus características, una vez comprendido este concepto, se estudian tres plataformas de *Cloud Computing* que son OpenStack, OpenNebula y Google Cloud señalando sus características y los componentes que poseen para su funcionamiento. Por otra parte, se comparan las tres plataformas de *Cloud Computing* y se selecciona una para su implementación. Con la *Cloud Computing* implementada se tiene servicios como: la creación de máquinas virtuales las cuales son gestionadas por la *Cloud Computing*, además de que se tiene la creación de redes de datos y con varias instancias de máquinas virtuales.

#### 3.1 Investigación sobre *Cloud Computing*, sus modelos de servicio y sus diferentes plataformas.

##### *Cloud Computing*

*Cloud Computing* o también conocido como computación en la nube es un modelo que permite prestar diferentes servicios o recursos como aplicaciones, servidores, redes, máquinas virtuales, entre otros a través de Internet. Esto permite al usuario tener un acceso remoto a los diferentes servicios que requiera desde cualquier dispositivo permitiendo que la computadora no tenga fuertes cargas de trabajo al momento de abrir una aplicación y esto a su vez permite que haya mayor agilidad en el trabajo [4].

El funcionamiento de *Cloud Computing* está basado en el uso de un servidor remoto el cual servirá para conectar los dispositivos como computadoras, celulares del usuario con los recursos que el necesite. Además de beneficiar a los usuarios también beneficia a los proveedores ya que podrán tener un mayor número de servicios que podrán otorgar de forma rápida y eficiente [5].

Entre sus características esenciales están:

- Accesibilidad por medio de Internet: Todos los recursos están disponibles mediante Internet y podrán ser accesibles desde cualquier lugar y dispositivo.
- Mayor Elasticidad: Esto tiene como referencia la escalabilidad ya que mediante *Cloud Computing* se podrá seguir expandiéndose según las necesidades que tenga el cliente.

- Servicio Bajo Demanda: Se tiene libertad para utilizar los servicios que se requieran además de que se adapta fácilmente a las necesidades por lo que no es necesario que se tenga que requerir aumento de la capacidad a un proveedor.
- Agrupación de Recursos: Los recursos están agrupados por lo que múltiples usuarios con distintos dispositivos pueden acceder a los mismos, y los recursos de estos estarán aislados de los demás usuarios.
- Servicio Medido: El uso de los recursos se encuentra monitoreado, es decir que será controlado el consumo de cómputo para así tener una transparencia entre el usuario y el proveedor con los servicios.

En cuanto a los tipos de nubes se puede identificar tres, las cuales se pueden observar en la Figura 3.1 y que se diferencian:

- Nube Pública: Una nube pública puede ofrecer el acceso a los recursos a cualquier usuario con acceso a Internet. El almacenamiento al igual que los servidores y aplicaciones van a ser proporcionados por el proveedor [6]. Este tipo de nube es más económico en su consumo, sin embargo, se debe tener confianza al proveedor ya que ellos se encargarán del mantenimiento y seguridad.
- Nube Privada: Una nube privada a diferencia de la pública va a ofrecer su propia infraestructura de la nube a una cantidad limitada de usuarios dentro de su organización por lo que serán estos los que estén a cargo de la seguridad y de igual manera del mantenimiento [6].
- Nube Híbrida: La nube híbrida es la combinación de la nube privada y pública esto quiere decir que aprovecha las mejores características de ambas nubes por ejemplo en seguridad y en ahorro al consumir, de esta manera las empresas optimizan y mejoran sus recursos existentes [6].



**Figura 3.1** Tipos de Nubes

Además de los tipos de nubes que se puede implementar, también se encuentran los modelos de servicio en los cuales se puede entender de mejor manera como se encuentra organizada y administrada la información. Existen tres modelos de servicio que son: IaaS, SaaS y PaaS.

- IaaS: IaaS (*Infrastructure as a Service*) o en español Infraestructura como Servicio, es un modelo de servicio en donde una organización va a contratar una infraestructura. Esto quiere decir que un proveedor alquila una infraestructura como por ejemplo servidores, redes, virtualización y almacenamiento para que el usuario pueda implementar un *software* dependiendo de sus necesidades. Además de que va a tener un control total sobre la nube es decir control sobre las aplicaciones, sistema operativo y datos [7].

Entre los ejemplos de IaaS se tiene: Google Cloud, OpenStack, OpenNebula, Amazon Web Services.

- SaaS: SaaS (*Software as a Service*) o en español *Software* como Servicio va a permitir acceder a un *software* el cual está listo para ser utilizado y que se ejecuta en una nube. Aquí el usuario que accede a estas aplicaciones no tiene la posibilidad de gestionar nada de la infraestructura, el proveedor será el encargado de toda la administración. Por otro lado, el usuario lo único que podrá hacer es consumir el servicio y para lo cual lo podrá hacer mediante cualquier dispositivo que posea con conexión a Internet [7].

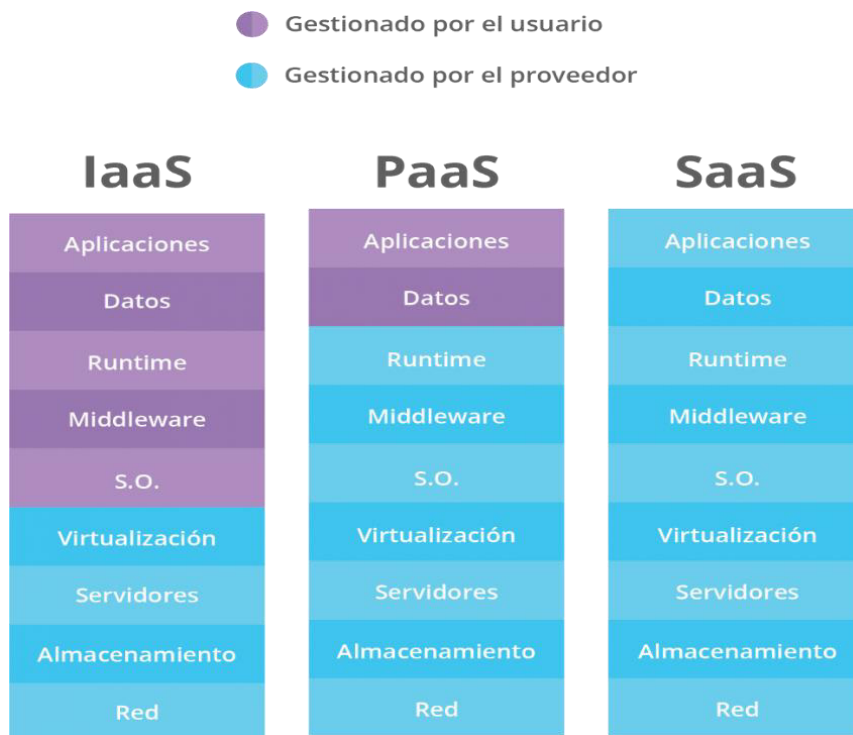
Entre los ejemplos de este modelo de servicio están: Dropbox, Office 365, Google Docs, Outlook.



- PaaS: PaaS (*Platform as a Service*) o en español Plataforma como Servicio permite al usuario poder crear y ejecutar sus propias aplicaciones, el proveedor se encargará de brindarle la plataforma, los recursos que necesite y además de la gestión y mantenimiento. El usuario tendrá únicamente el control de las aplicaciones y de los datos. A diferencia de IaaS, en este modelo de servicio no se tiene control del servidor, y a diferencia de SaaS, se tiene el control a las aplicaciones [8].

Entre los ejemplos de PaaS se tiene: OpenShift, Google App Engine, Oracle Cloud Platform, IBM Cloud Foundry.

En la Figura 3.2 se puede visualizar las gestiones que tiene tanto el usuario como el proveedor dependiendo del modelo de servicio ya que como se explicó anteriormente en cada modelo se va a poseer un control diferente de los servicios que se ofrecen.



**Figura 3.2** Modelos de servicio [9]

## Plataformas

### OpenStack

OpenStack es una plataforma de *Cloud Computing* que proporciona un modelo de servicio IaaS (Infraestructura como Servicio). Es un *software* libre que permite crear nubes tanto privadas, nubes públicas e híbridas, esto permite tener el control de ciertos recursos como redes, almacenamiento, creación de instancias y todo esto mediante una interfaz gráfica. Cabe recalcar que OpenStack solo funciona en sistemas Operativos de Linux por lo que se puede utilizar de manera ilimitada y gratuita [10].

OpenStack nace en el año 2010 por una empresa llamada Rackspace en conjunto con la NASA, y se funda debido a la necesidad que tenía la NASA para sus proyectos de crear redes de una manera flexible. Esta plataforma se encuentra bajo la licencia de Apache 2.0 por lo que su código es abierto. En la actualidad se tiene muchos colaboradores de alrededor de 179 países, además de muchas organizaciones registradas y proyectos activos [11].

Entre sus características están:

- Costes: OpenStack es gratuito y de código libre por lo que los usuarios no van a depender de algún proveedor ya que al tener una licencia Apache todo lo que utilice el usuario será gratuito, además de que la plataforma no tiene una versión de pago [12].
- Autonomía: El usuario no va a necesitar ayuda de algún administrador para poder lanzar instancias, ya que lo podrá hacer de una manera autónoma y de forma fácil haciendo que la plataforma sea muy sencilla de utilizar [12].
- Escalable: Los recursos que el usuario necesite como máquinas virtuales, redes, seguridad y entre otros, la plataforma te los proporciona de una manera rápida para que así el usuario pueda seguir creando sus propios proyectos según sus necesidades [12].
- Flexibilidad: Esto tiene que ver con que el código sea abierto ya que los usuarios pueden adaptarlo a sus necesidades haciendo que un grupo determinado de usuarios se beneficien con las modificaciones realizadas ya que estos se pueden compartir con la comunidad [12].

OpenStack posee una arquitectura modular y en sus inicios tenía dos módulos fundamentales para funcionar de manera correcta los cuales son Nova y Swift,

actualmente estos módulos han aumentado y oficialmente existen 7 módulos obligatorios que se deben instalar en conjunto con la *Cloud Computing* y estos son:

- Cinder: Este módulo se encarga del almacenamiento en bloque, es decir que funciona como un disco duro y todo esto gracias a la virtualización lo que permite que se puedan alojar las máquinas virtuales. Además, permite crear copias de seguridad o *Snapshots* para restaurar datos o instancias en volúmenes de almacenamiento. Volúmenes se refiere a un almacenamiento a nivel de bloque los cuales van a poseer las máquinas virtuales que están instaladas en la nube, además de que van a permitir al usuario gestionar el almacenamiento de acuerdo a sus necesidades [13].
- Nova: Es el módulo más importante de OpenStack, y el primero con el cual inició dicha *Cloud Computing*. Se la considera como el motor o el cerebro de OpenStack debido a que este controla la Infraestructura y es utilizado para la gestión y suministro de instancias, es decir la creación y eliminación de máquinas virtuales. Gracias a su desarrollo ahora este módulo se puede encargar de gestionar grandes grupos de máquinas virtuales [13].
- Neutron: Este módulo también es importante, pues permite que todas las instancias se comuniquen y así puedan intercambiar información. Además, se encarga de la gestión de redes tales como dirección IP, subredes, VLANs y entre otros. Otra función de este módulo es el de generar un *firewall* permitiendo seguridad en los proyectos. En otras palabras, es encargado de todo lo que tiene que ver con la conectividad por lo que ayudará a los usuarios al momento de crear sus redes y conectarlas con las instancias utilizando componentes como *routers* y *switches* [13].
- Keystone: Este módulo es el encargado de autenticación de usuarios permitiendo un acceso diferente a cada usuario para los servicios que ofrece OpenStack. Es quien autoriza o prohíbe a todos los componentes a que se realicen ciertas acciones que tengan que ver con lo que solicite el *Cloud Computing*. Este módulo utiliza un sistema de *Tokens* el cual es utilizado como un *ticket* el cual brinda dicho módulo después de colocar las credenciales correctas para poder realizar algo como por ejemplo eliminar una máquina virtual. De esta manera se tiene cierta seguridad para las peticiones del sistema [14].

- Glance: Este módulo se encarga de ofrecer y de almacenar las imágenes de los Sistemas Operativos para poder crear las máquinas virtuales. Ofrece flexibilidad de tal manera que el usuario pueda acceder a dichas imágenes en cualquier instante. Además, se puede utilizar el componente Swift como un servicio de almacenamiento para las imágenes [14].
- Horizon: Este módulo es encargado de proporcionar una interfaz gráfica para que los usuarios puedan acceder, de esta manera se puede simplificar el uso de la plataforma haciéndola más sencilla de manejar. Esta interfaz permitirá comunicarse con todos los servicios de OpenStack, además de poder observar un monitoreo de lo que el usuario realice en sus proyectos [15].
- Swift: Este módulo se comporta como un sistema de almacenamiento, es decir que almacena todos los datos que exista en el *software* para que así siempre estén disponibles al momento de necesitarlos. Este módulo también puede funcionar con otro módulo como Glance el cual contiene las imágenes de los Sistemas Operativos. Este módulo también es un sistema escalable y con tolerancia a fallos por lo que siempre los datos van a estar disponible para el usuario [15].

## **OpenNebula**

OpenNebula es una plataforma de código libre bajo una licencia de Apache y que posee un modelo de servicio IaaS. Esta plataforma se encarga de la creación y gestión de nubes tanto públicas, privadas o híbridas dependiendo de la necesidad del usuario, además OpenNebula se encarga del manejo de diferentes servicios como máquinas virtuales, almacenamiento, seguridad y configuraciones de red. Posee una interfaz gráfica que permitirá al usuario poder gestionar de mejor manera sus proyectos.

OpenNebula se desarrolla en el año 2008 en la Universidad Complutense de Madrid por un equipo el cual estaba en un trabajo de investigación sobre las plataformas IaaS además de que se buscaba soluciones en cuanto a la gestión de máquinas virtuales. Oficialmente en el año 2010 se funda C12G Labs el cual es una compañía encargada de gestionar y dar soporte a la plataforma de OpenNebula; ahora dicha compañía se llama OpenNebula Systems y está compuesto por muchas organizaciones las cuales han contribuido para poder desarrollar el proyecto [16].

Posee componentes básicos los cuales son necesarios para poder utilizar la plataforma los cuales son:

- Red Virtual: Tiene redes virtuales las cuales ayudan a que las máquinas virtuales tengan direcciones IP para cada red [16] .
- Clústeres: Es un pool de *hosts* los cuales van a compartir el almacenamiento al igual que las redes virtuales [16].
- Plantillas: Se tiene un lugar en donde se encuentran las plantillas de las máquinas virtuales [17].
- Máquinas virtuales: Se tiene cierto control al momento de lanzar una instancia de alguna plantilla como control en el despliegue de la instancia, su parada, su eliminación o incluso su clonación [17].
- Almacenamiento: El almacenamiento se refiere a las imágenes de las máquinas virtuales es decir del sistema operativo o incluso de datos que se encuentran alojadas en diferentes repositorios para que el usuario pueda acceder a ellos de una manera rápida y sencilla [17].
- API: La interfaz de programación de aplicaciones (API) va a proporcionar al usuario una interfaz para poder acceder a los servicios de la plataforma ya sea por medio de línea de comandos o por medio de una interfaz web que en este caso se llama Sunstone [17].

La interfaz gráfica web de este *software* se llama Sunstone el cual proporciona una vista muy adecuada para poder gestionar los proyectos que se realicen, además de que se podrá observar las imágenes para las máquinas virtuales y estas puedan convertirse en instancias. De esta manera se tiene un mejor control de todos los recursos que el usuario utiliza para así corregir errores [17].

## **Google Cloud**

Google Cloud Platform por sus siglas GCP, es una plataforma que permite crear una propia Infraestructura en la nube basándose en los tres modelos de servicio (IaaS, PaaS, SaaS). De manera general Google Cloud es un conjunto de servicios en la nube que buscan soluciones como el almacenamiento, *networking*, bases de datos los cuales son ideales para las empresas ya que podrán utilizar dichas herramientas dependiendo de la necesidad que estos tengan, de una manera sencilla y personalizada [18].

Esta plataforma posee un modelo de pago por uso por lo que se pagará dependiendo del consumo que tenga el usuario o la organización haciendo que se tenga un gran ahorro en los costos.

Entre sus principales características están:

- Seguridad: La seguridad es alta puesto ya que posee la infraestructura de Google, esto hace que los datos siempre estén disponibles y seguros, además existe una encriptación en los datos [18].
- Innovador: Esto tiene que ver con la inteligencia artificial pues la plataforma posee herramientas modernas para que se pueda trabajar de manera conjunta con *Machine Learning* haciendo que el usuario posea innovación en sus proyectos [18].
- *Open Source*: Ya que es de código abierto los usuarios pueden tener cierta personalización en las herramientas dependiendo de las necesidades del usuario o la organización [18].

Entre los servicios que ofrece la plataforma, las herramientas se pueden agrupar en 5 que son:

- *Networking*: Aquí se tendrán herramientas las cuales van a estar completamente relacionada con la red, desde la conexión de las máquinas virtuales hasta incluso reglas de *firewall*. Entre sus herramientas internas están *Content Delivery Network*, *Cloud Interconnect*, *Cloud Network*, *Cloud Domain Name System*, entre otros [19].
- *Computing*: Aquí se tienen ciertas herramientas que van a permitir diseñar aplicaciones como por ejemplo *App Engine*; otra herramienta es la de *Compute Engine* que en cambio ayuda a la gestión de máquinas virtuales en la nube y finalmente está *Kubernetes Engine* que es una mezcla de ambas herramientas aprovechando sus mejores ventajas para permitir construir *clusters* [19].
- *Storage*: Aquí se tienen a las herramientas de almacenamiento como por ejemplo *Cloud Storage* el cual se encarga de ofrecer almacenamiento y un acceso instantáneo a los datos desde donde sea. Además, están las herramientas *Cloud SQL* y *Cloud BigTable* que se encargan del almacenamiento de los datos tipo SQL, NOSQL [19].
- *Big Data*: Aquí se tiene ciertas herramientas para el procesamiento de *Big Data* en la nube como por ejemplo *Bigquery* el cual permite analizar grandes cantidades de información de una manera rápida y con resultados en tiempos cortos [19].

- *Machine Learning*: Dentro de las herramientas que posee la plataforma se encuentra *Cloud Machine Learning* el cual permite crear ciertos modelos de aprendizaje automático mediante patrones para así poder resolver las necesidades del usuario [19].

### **3.2 Determinación de una *Cloud Computing* que utilice *software open source* para su implementación.**

Una vez realizado el estudio de ciertas plataformas de *Cloud Computing*, se seleccionó una para su implementación y esta fue la de OpenStack.

Se seleccionó esta plataforma porque a comparación de OpenNebula y Google Cloud. OpenStack posee ciertas características que OpenNebula no considera, como por ejemplo un componente en específico para las autorizaciones de los usuarios, pues esto es de suma importancia al momento que los usuarios realicen ciertas acciones que tengan que ver con el sistema.

Ahora si se compara OpenStack con Google Cloud la diferencia es que esta plataforma tiene ciertos componentes que son de paga y aunque Google dé un bono para el uso de su nube, al momento de acabarse dicho bono el usuario puede quedarse sin poder implementar algún proyecto que esté realizando por lo que limita el uso completo de esta *Cloud Computing*.

Otra *Cloud* a tener en cuenta fue la de *Cloud Foundry* la cual es desarrollada por la empresa *VMware*, es una plataforma conocida y similar a OpenStack ya que poseen *software* de código abierto. Sin embargo, se tiene una diferencia, *Cloud Foundry* posee un modelo de servicio PaaS por lo que la *Cloud Computing* va a proveer una plataforma y el usuario tendrá la gestión de las aplicaciones y los datos.

En cambio, con OpenStack se posee un modelo de servicio IaaS en donde se provee una infraestructura y la gestión del usuario se basa en los datos, el sistema operativo, las aplicaciones y el tiempo de ejecución. Incluso la plataforma de *Cloud Foundry* puede ser alojada dentro de un servidor OpenStack [20].

Es por ello que se determinó la plataforma de OpenStack para ser implementada por lo que se investigó información para instalarla dentro de una máquina virtual, además se realizó la Tabla 3.1 para observar de mejor manera lo anteriormente explicado.

**Tabla 3.1** Comparación de Plataformas

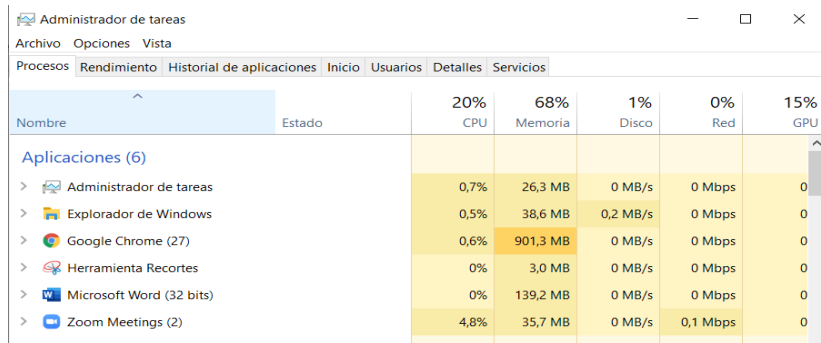
	<b>OpenStack</b>	<b>OpenNebula</b>	<b>Google Cloud</b>
<b>Código Abierto</b>	Si	Si	No
<b>Modelo de Servicio</b>	IaaS	IaaS	IaaS,PaaS,SaaS
<b>Uso</b>	Gratuito	Gratuito	Con herramientas de paga
<b>Componente de Autorización</b>	Si (Keystone)	No	Si
<b>Interfaz Gráfica</b>	Si (Horizon)	Si (Sunstone)	Si
<b>Facilidad de instalación de ISOs en una máquina virtual</b>	Fácil	Fácil	Difícil

Para la instalación de OpenStack se utilizó DevStack el cual es un conjunto de *scripts* que permiten instalar y abrir un entorno de OpenStack de manera fácil. Para poder implementar la *Cloud Computing* se utilizó una máquina virtual de Ubuntu específicamente la versión 20.04, debido a que esta versión es sugerida en la página de OpenStack ya que se ha realizado varias pruebas y su ejecución es fluida dando como resultado que la plataforma se encuentre estable.

### **3.3 Implementación de una *Cloud Computing* sobre una máquina virtual**

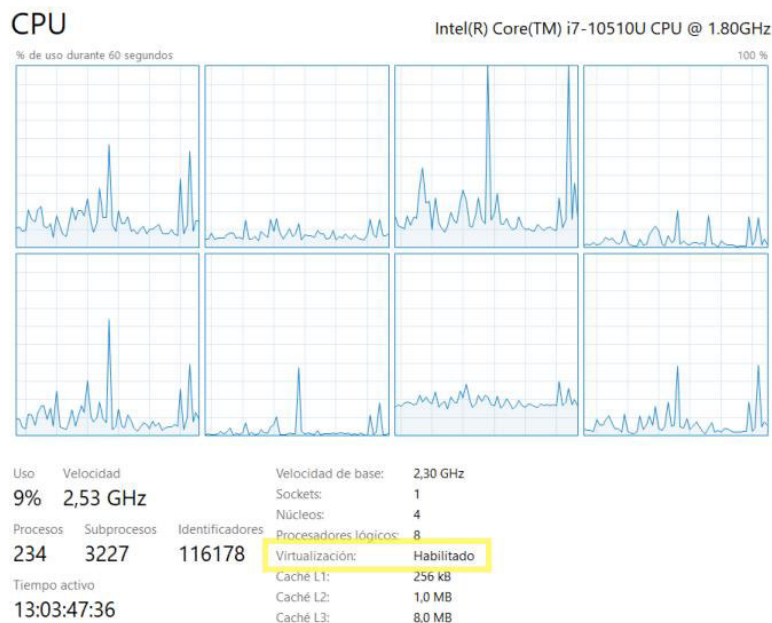
Antes de instalar la plataforma de OpenStack se debe verificar que la virtualización dentro del *host* esté habilitada y para poder comprobar lo antes mencionado se debe ejecutar el Administrador de Tareas. Dicho administrador se presenta en la Figura 3.3.





**Figura 3.3** Administrador de Tareas

Una vez ejecutado el Administrador de Tareas, se seleccionó la opción de Rendimiento. Por defecto en esta opción se presenta el rendimiento de la CPU, es aquí en donde se puede observar si se encuentra habilitada la virtualización. Como se puede observar en la Figura 3.4, la virtualización se encuentra habilitada por lo que se pudo continuar con la instalación de la plataforma de OpenStack.



**Figura 3.4** Virtualización Habilitada

## Instalación de OpenStack

Para iniciar con la instalación de OpenStack lo primero que se hizo fue agregar un usuario no *root* llamado "stack" y con sudo habilitado, por lo que se colocó el comando que se observa dentro de la Figura 3.5.

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

steven@steven-virtual-machine:~/Escritorio$ sudo useradd -s /bin/bash -d /opt/stack -m stack
[sudo] contraseña para steven:
steven@steven-virtual-machine:~/Escritorio$
```

**Figura 3.5** Agregación de un usuario

A continuación, se presenta en la Figura 3.6 los comandos con los cuales se procedió a brindar privilegios al usuario ya que se realizó algunos cambios en el sistema.

```
steven@steven-virtual-machine:~/Escritorio$ echo "stack ALL=(ALL) NOPASSWD: ALL" | sudo tee /etc/sudoers.d/stack
stack ALL=(ALL) NOPASSWD: ALL
steven@steven-virtual-machine:~/Escritorio$ sudo -u stack -i
stack@steven-virtual-machine:~$
```

**Figura 3.6** Permisos de super usuario

Después se procedió a instalar el paquete *git* con el comando que se puede visualizar en la Figura 3.7 ya que con este componente se puede clonar el repositorio para descargar el DevStack que posteriormente nos permitirá instalar OpenStack.

```
stack@steven-virtual-machine: ~
Desea continuar? [S/n] s
Des:1 http://ec.archive.ubuntu.com/ubuntu focal/main amd64 liberror-perl all 0.17029-1 [26,5 kB]
Des:2 http://ec.archive.ubuntu.com/ubuntu focal-updates/main amd64 git-man all 1:2.25.1-1ubuntu3.2 [884 kB]
Des:3 http://ec.archive.ubuntu.com/ubuntu focal-updates/main amd64 git amd64 1:2.25.1-1ubuntu3.2 [4.554 kB]
Descargados 5.465 kB en 9s (615 kB/s)
Seleccionando el paquete liberror-perl previamente no seleccionado.
(Leyendo la base de datos ... 189127 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../liberror-perl_0.17029-1_all.deb ...
Desempaquetando liberror-perl (0.17029-1) ...
Seleccionando el paquete git-man previamente no seleccionado.
Preparando para desempaquetar .../git-man_1%3a2.25.1-1ubuntu3.2_all.deb ...
Desempaquetando git-man (1:2.25.1-1ubuntu3.2) ...
Seleccionando el paquete git previamente no seleccionado.
Preparando para desempaquetar .../git_1%3a2.25.1-1ubuntu3.2_amd64.deb ...
Desempaquetando git (1:2.25.1-1ubuntu3.2) ...
Configurando liberror-perl (0.17029-1) ...
Configurando git-man (1:2.25.1-1ubuntu3.2) ...
Configurando git (1:2.25.1-1ubuntu3.2) ...
Procesando disparadores para man-db (2.9.1-1) ...
stack@steven-virtual-machine:~$
```

**Figura 3.7** Instalación del componente git

Una vez instalada la herramienta se procedió a clonar el repositorio DevStack. Devstack es un repositorio que contiene un *script* para poder instalar el OpenStack además de plantillas para poder configurarlo y utilizarlo, por lo que se utilizó los comandos que se pueden visualizar dentro de la Figura 3.8.

```
stack@steven-virtual-machine:~$ git clone https://opendev.org/openstack/devstack
Clonando en 'devstack'...
remote: Enumerating objects: 24547, done.
remote: Counting objects: 100% (24547/24547), done.
remote: Compressing objects: 100% (3032/3032), done.
remote: Total 47939 (delta 23940), reused 21515 (delta 21515), pack-reused 23392
Recibiendo objetos: 100% (47939/47939), 9.30 MiB | 8.38 MiB/s, listo.
Resolviendo deltas: 100% (34103/34103), listo.
stack@steven-virtual-machine:~$ cd devstack
stack@steven-virtual-machine:~/devstack$
```

**Figura 3.8** Clonación de Devstack

A continuación, se creó un archivo llamado “local.conf” en donde se colocaron ciertas configuraciones para poder instalarlo en el OpenStack. En la página de Devstack se puede encontrar un documento “local.conf” el cual es un archivo que contiene las configuraciones mínimas para poder instalar el OpenStack. Dichas configuraciones se copiaron al archivo creado. A continuación, se indica el archivo en la Figura 3.9. Este archivo es el que genera las claves para poder ingresar a OpenStack.

```
stack@steven-virtual-machine: ~/devstack
GNU nano 4.8 local.conf
# The ``localrc`` section replaces the old ``localrc`` configuration file.
# Note that if ``localrc`` is present it will be used in favor of this section.
[[local|localrc]]

# Minimal Contents
# -----

# While ``stack.sh`` is happy to run without ``localrc``, devlife is better when
# there are a few minimal variables set:

# If the ``*_PASSWORD`` variables are not set here you will be prompted to enter
# values for them by ``stack.sh`` and they will be added to ``local.conf``.
ADMIN_PASSWORD=nomoresecret
DATABASE_PASSWORD=stackdb
RABBIT_PASSWORD=stackqueue
SERVICE_PASSWORD=$ADMIN_PASSWORD

# ``HOST_IP`` and ``HOST_IPV6`` should be set manually for best results if
# the NIC configuration of the host is unusual, i.e. ``eth1`` has the default

^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Text ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar ^T Ortografía ^_ Ir a línea
```

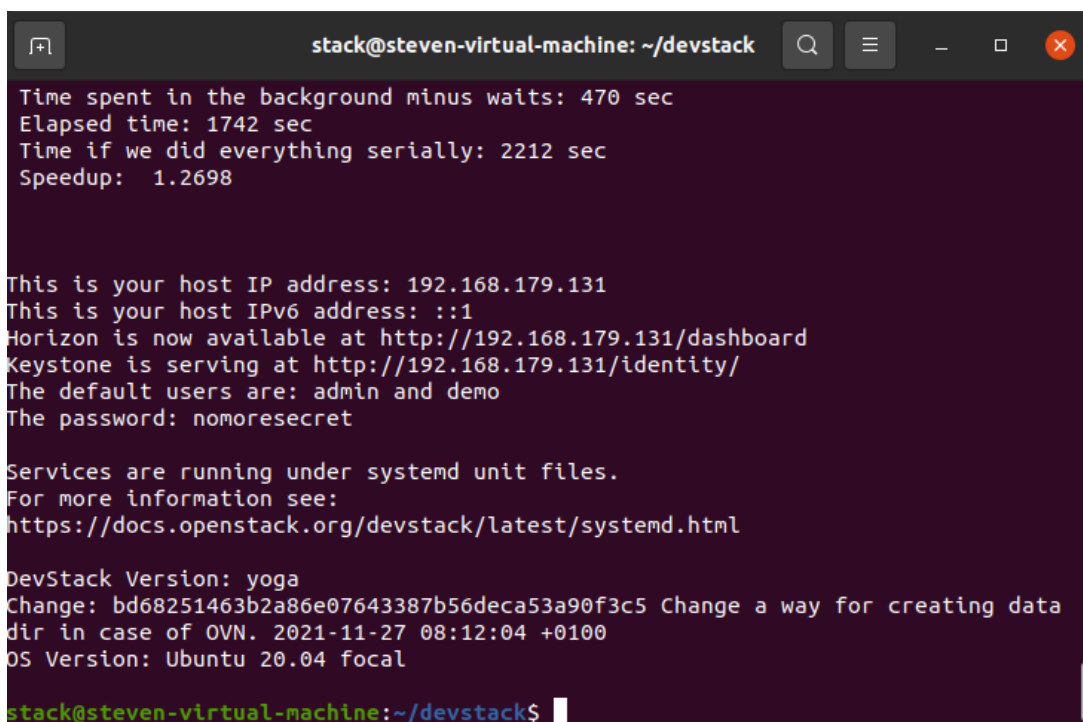
**Figura 3.9** Creación del archivo “local.conf”

Una vez creado el archivo, se comenzó con la instalación el cual tardó varios minutos para lo cual se colocó el comando que se puede ver en la Figura 3.10. Con dicho comando, el Devstack hace pruebas unitarias esto quiere decir que prueba cada módulo para que su funcionamiento sea el correcto.

```
stack@steven-virtual-machine:~/devstack$ ./stack.sh
```

**Figura 3.10** Ejecución del script stack.sh

Una vez finalizada la instalación en el terminal se presentará una dirección IP con la cual se puede acceder a la interfaz web del OpenStack, esta IP pertenece a una interfaz de red del *host*; además dará las credenciales necesarias para poder acceder y empezar a utilizar el OpenStack. Estas credenciales se pueden observar en la Figura 3.11.



```
stack@steven-virtual-machine: ~/devstack
Time spent in the background minus waits: 470 sec
Elapsed time: 1742 sec
Time if we did everything serially: 2212 sec
Speedup: 1.2698

This is your host IP address: 192.168.179.131
This is your host IPv6 address: ::1
Horizon is now available at http://192.168.179.131/dashboard
Keystone is serving at http://192.168.179.131/identity/
The default users are: admin and demo
The password: nomoresecret

Services are running under systemd unit files.
For more information see:
https://docs.openstack.org/devstack/latest/systemd.html

DevStack Version: yoga
Change: bd68251463b2a86e07643387b56deca53a90f3c5 Change a way for creating data
dir in case of OVN. 2021-11-27 08:12:04 +0100
OS Version: Ubuntu 20.04 focal

stack@steven-virtual-machine:~/devstack$
```

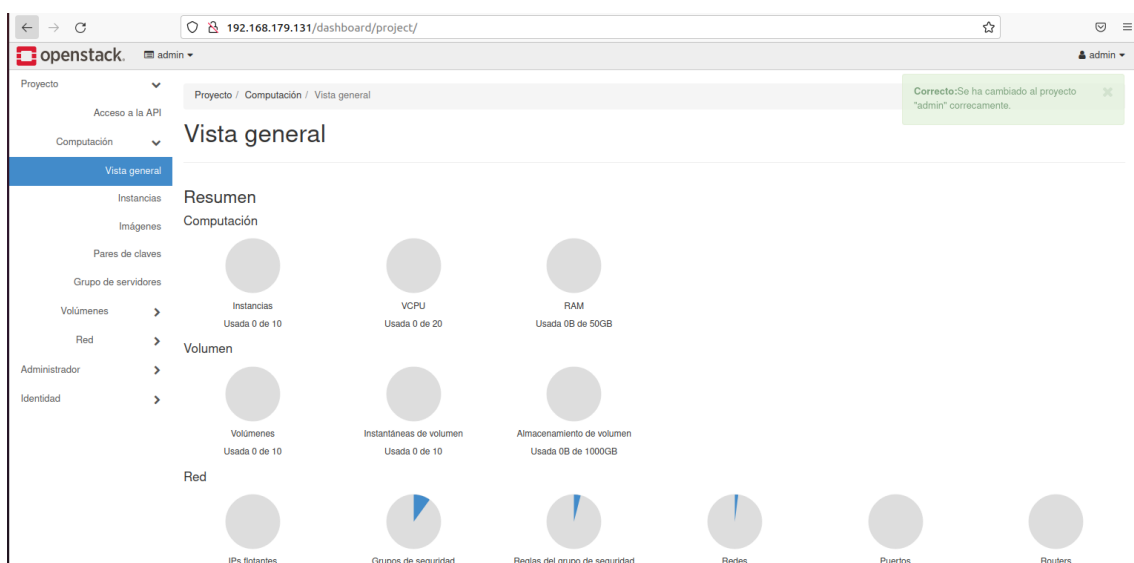
**Figura 3.11** Finalización de la instalación de OpenStack

Para su implementación se colocó el usuario y la contraseña que se presentó en la Figura 3.11, dichas credenciales son creadas gracias al módulo Keystone y servirán para acceder a OpenStack mediante la dirección IP que se presenta en la Figura 3.11. Además de esto se puede observar la versión que se está utilizando en OpenStack la cual es “Yoga”; es una versión la cual es estable y se encuentra en producción actualmente en OpenStack. A continuación, se puede observar el inicio de sesión de OpenStack dentro de la Figura 3.12.



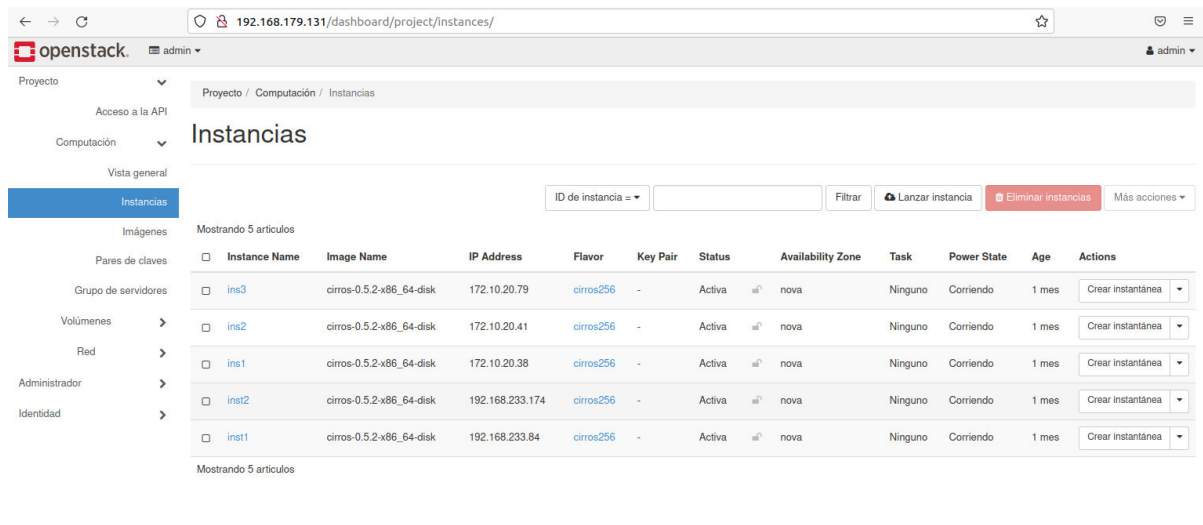
**Figura 3.12** Inicio de sesión de OpenStack

Una vez que se ingresa a la dirección IP suministrada por DevStack, se puede visualizar el *dashboard* de OpenStack. En primer lugar, se puede observar una vista general tanto de instancias, RAM, volúmenes, redes y más, tal como se observa en la Figura 3.13. Se debe tener en cuenta que se tiene ciertos límites como por ejemplo se puede utilizar máximo hasta 10 instancias, esto tiene que ver con las configuraciones de la máquina virtual ya que no se puede exceder de los valores del hardware del cliente, si se sobrepasa de estos valores vamos a generar sobre virtualización, en muchos casos el hipervisor elimina este tipo de problema limitando la capacidad de la máquina virtual.



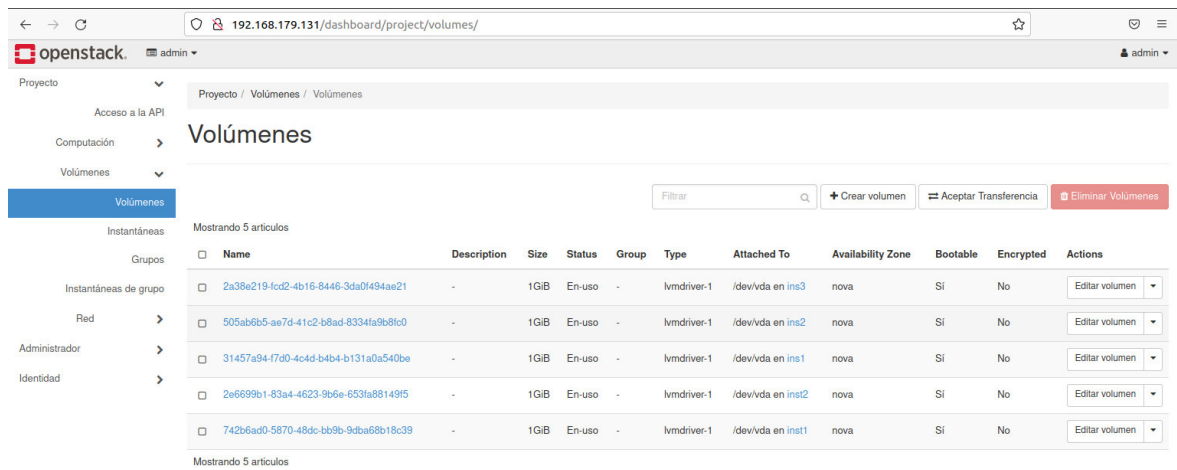
**Figura 3.13** Interfaz gráfica de OpenStack

Además de la vista general, hay más opciones que van a permitir crear instancias, redes y *routers*; por ejemplo, hay la opción de “Instancias” la cual se la puede observar en la Figura 3.14, y va a permitir gestionar todas las instancias que hayan sido creadas previamente.



**Figura 3.14** Vista de la opción “Instancias”

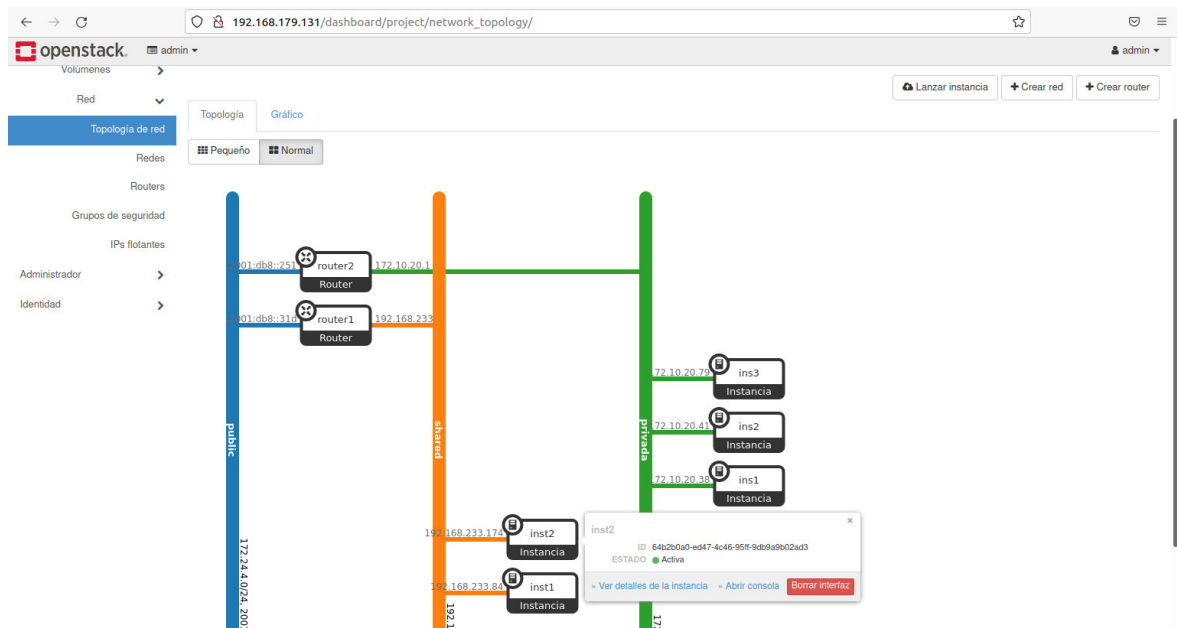
Existe otra opción llamada “volúmenes”, en donde se puede visualizar los volúmenes que se están utilizando en las instancias que se han creado. En la Figura 3.15 se observa los diferentes volúmenes utilizados.



**Figura 3.15** Vista de la opción “Volúmenes”

Otra de las opciones importantes es la de “Red” en donde se puede crear *routers*, además de crear las redes para conectar las instancias. Aquí también se puede observar

la topología de las redes de datos creadas. Como se observa en la Figura 3.16 se tiene una vista de la topología de una red que previamente se creó.



**Figura 3.16** Vista de la opción “Red”

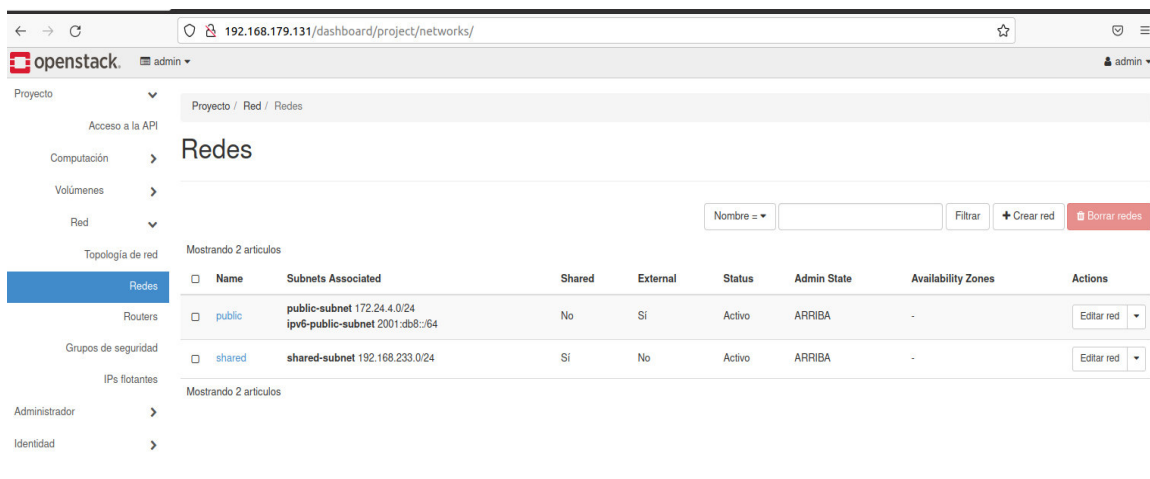
Una vez familiarizado con la Dashboard, se realizaron diferentes pruebas de funcionamiento para verificar el funcionamiento de la *Cloud Computing*.

### 3.4 Pruebas de funcionamiento de la *Cloud Computing*

Para la realización de pruebas de funcionamiento en la *Cloud Computing* implementada, en la parte de *networking* se crearon redes con *routers* y subredes. Además, se crearon instancias de máquinas virtuales.

#### Creación de redes

Dentro de la sección “Red” se encuentran varias opciones las cuales ayudarán para la creación de redes y *routers*. Una de estas opciones se llama “Redes”. Como se puede observar en la Figura 3.17, por defecto vienen creadas 2 redes, una red pública y una red compartida.



**Figura 3.17** Creación de redes

Al momento de colocar en “Crear red” se despliega una ventana en donde se solicita colocar un nombre a la red, además se debe marcar la casilla “Compartida”. La ventana de “Crear red” se puede observar en la Figura 3.18.

**Figura 3.18** Configuración de la red

Al colocar en “Siguiente” se despliega las configuraciones de la subred, en donde se debe colocar un nombre, la dirección de la red, la versión de IP ya que se puede tener IPv4 e IPv6, y finalmente la puerta de enlace. Para este caso se colocó la red 172.10.20.0/24 la cual se presenta en la Figura 3.19.



**Crear red**

Red Subred Detalles de Subred

**Nombre de subred**

**Origen Dirección de Red**

**Direcciones de red**

**Versión de IP**

**IP de la puerta de enlace**

Deshabilitar puerta de enlace

Crea una subred asociada a la red. Es necesario añadir una "dirección de red" y una "IP de la puerta de enlace" válidos. Si no añade una "IP de la puerta de enlace", el primer valor de la red se asignará por defecto. Si no quiere puerta de enlace, seleccione "Deshabilitar puerta de enlace". La configuración avanzada está disponible haciendo click en la pestaña "Detalles de subred".

Cancelar « Anterior Siguiente »

**Figura 3.19** Configuraciones de la subred

Al colocar en "Siguiente" se despliega los detalles de la subred. Aquí se coloca ciertos datos como el *pool* de asignación el cual permite mediante el DHCP poder asignar de manera dinámica a las instancias una dirección IP para este caso se asignó desde la dirección 172.10.20.10 hasta la 172.10.20.200; además se solicita el servidor DNS. Estas direcciones se pueden observar en la Figura 3.20.

**Crear red**

Red Subred Detalles de Subred

**Habilitar DHCP**

**Pools de asignación**

**Servidores DNS**

**Rutas de host**

Especificar atributos adicionales para la subred.

Cancelar « Anterior Crear

**Figura 3.20** Detalles de la subred mediante DHCP

Por último, se coloca en “Crear” y eso sería todo para crear una red. Para poder observar de mejor manera se colocó dentro de la opción “Topología de red”. Dicha topología se presenta en la Figura 3.21.

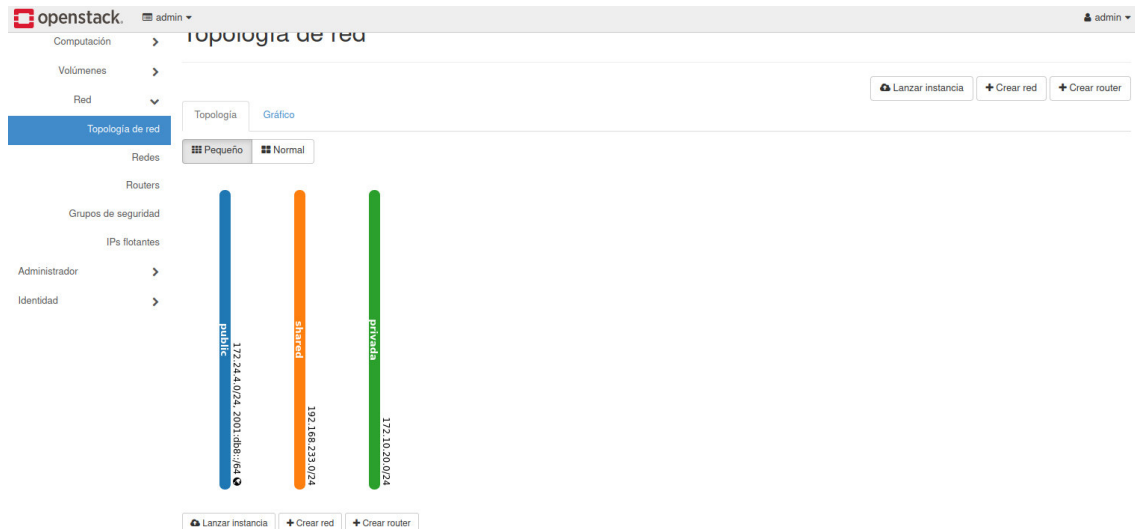


Figura 3.21 Vista de Topología de red creada

### Creación de *routers*

Para la creación del *router* se encuentra la opción “Crear *Router*”, se desplegará una ventana como se observa en la Figura 3.22. Se colocó un nombre al *router* y se añadió la red a la cual se va a conectar dicho *router*. Se debe dejar por defecto habilitado el SNAT ya que esto hace referencia a cuando un host desea una conexión con un host externo, el *router* cambia la dirección IP privada del host origen a una IP pública

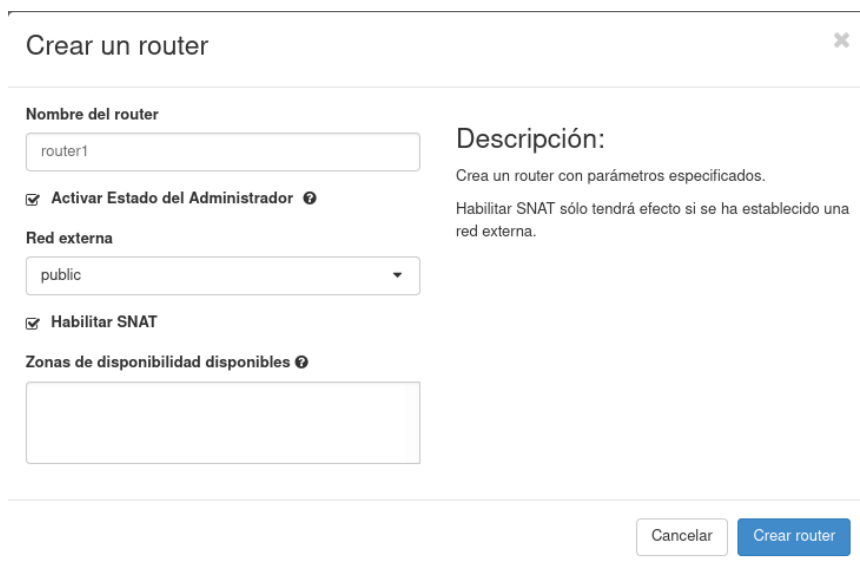


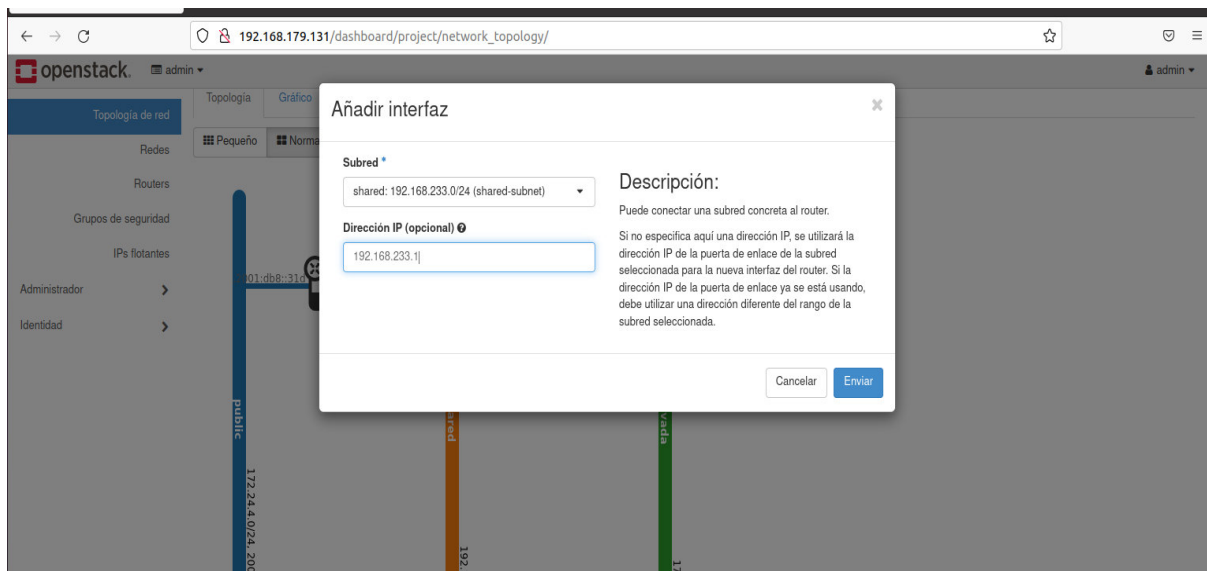
Figura 3.22 Creación de un router

Una vez creado el *router*, se puede observar en la topología de la Figura 3.23. Y para añadirle una interfaz, sólo se debe dar clic en el *router* y en la opción “añadir interfaz”.



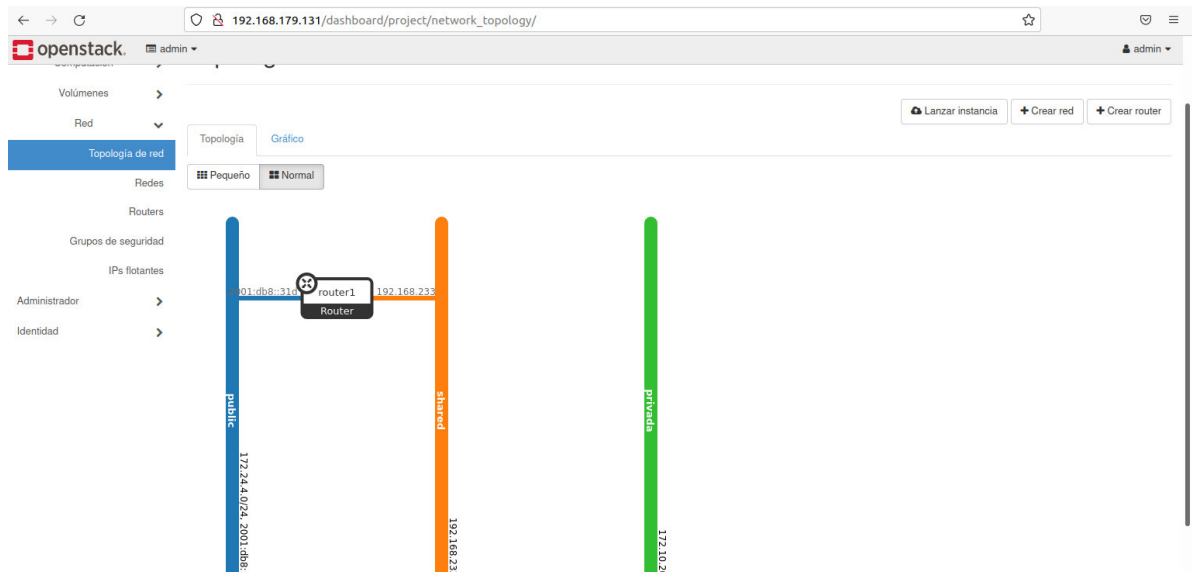
**Figura 3.23** Router creado

En este punto se solicita que se coloque una subred como se especifica en la Figura 3.24 y también la dirección IP, para este caso se conectó en la subred llamada “*shared*” y la dirección IP puede ser opcional ya que por defecto va a escoger la dirección de puerta de enlace que se colocó al momento de crear la red.



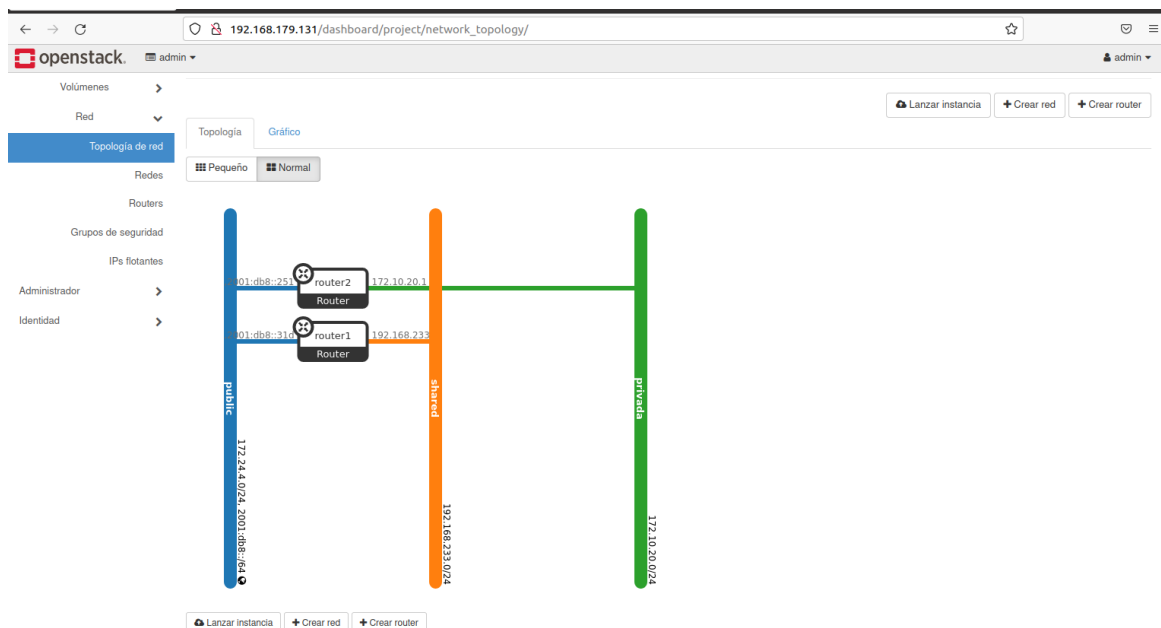
**Figura 3.24** Añadir interfaz a un *router*

Una vez finalizada las configuraciones para añadir la interfaz, se coloca en “Crear” para así visualizar el *router* con sus interfaces conectadas a las redes tal como se indica en la Figura 3.25.



**Figura 3.25** Router añadido con interfaces

Para poder conectar la otra red, se hace exactamente lo mismo creando un nuevo *router*, y de la misma manera conectando las interfaces. Una vez finalizado, la topología será más o menos como se observa en la **Figura 3.26**.



**Figura 3.26** Routers conectados a las redes

## Creación de instancias de máquinas virtuales

Para la creación de instancias se lo puede realizar con la opción llamada “Lanzar instancia” que se encuentra ubicada a lado de las opciones de “Crear red” Y “Crear router”. Se puede observar en la Figura 3.27.



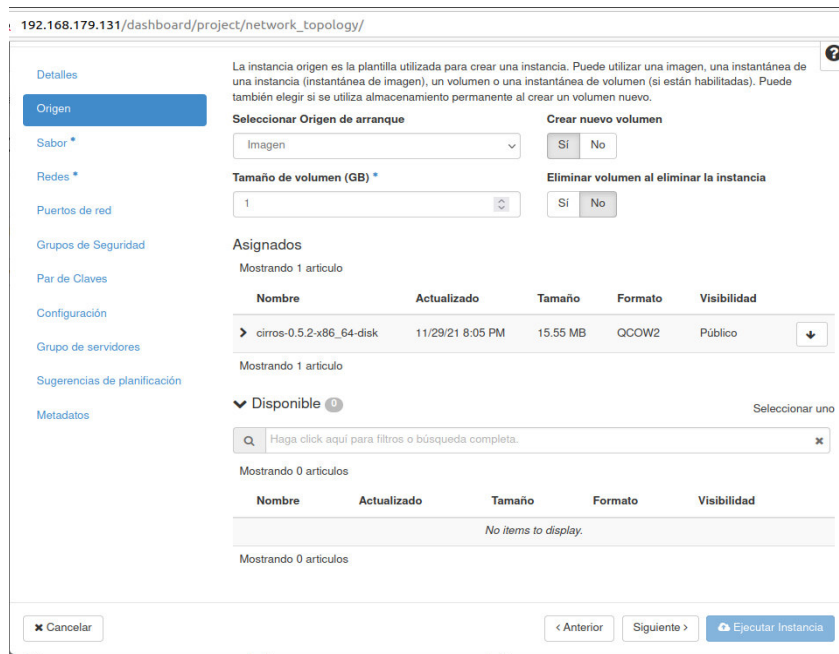
**Figura 3.27** Creación de una instancia

Al dar clic se desplegó una pestaña donde se podrá elegir ciertas configuraciones para la creación de la instancia como detalles, origen, sabor, redes, puertos de red y entre otras más. Todas estas configuraciones se pueden observar en la Figura 3.28.

The screenshot shows a web interface for creating a virtual machine instance. The title is 'Ejecutar Instancia'. On the left is a sidebar with a 'Detalles' tab selected, and other options like 'Origen', 'Sabor', 'Redes', etc. The main area contains a form with the following fields: 'Nombre del proyecto' (admin), 'Nombre de la instancia' (empty), 'Descripción' (empty), 'Zona de Disponibilidad' (nova), and 'Número' (1). A 'Total de Instancias (10 Max)' gauge shows 10% usage, with a legend indicating 0 actual, 1 added, and 9 remaining. At the bottom are 'Cancelar', '< Anterior', 'Siguiete >', and 'Ejecutar Instancia' buttons.

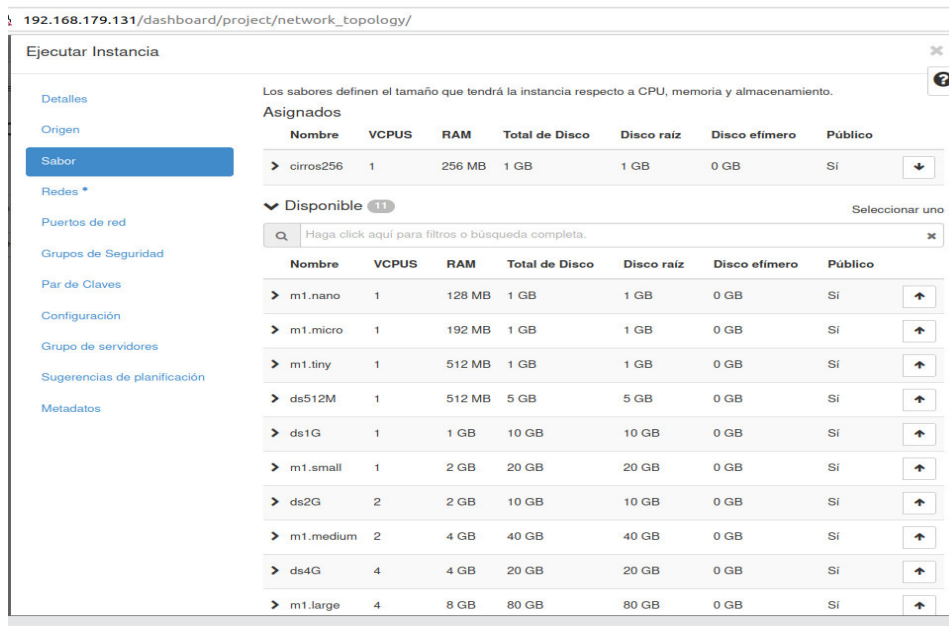
**Figura 3.28** Configuración de la instancia

Como se ve en la Figura 3.28 se colocó un nombre para la instancia y luego se colocó en “Siguiete”. Dentro de la opción “Origen” se asignó una plantilla por lo que se seleccionó la única que hay y se colocó en “Siguiete” tal como se observa en la Figura 3.29.



**Figura 3.29** Origen de la instancia

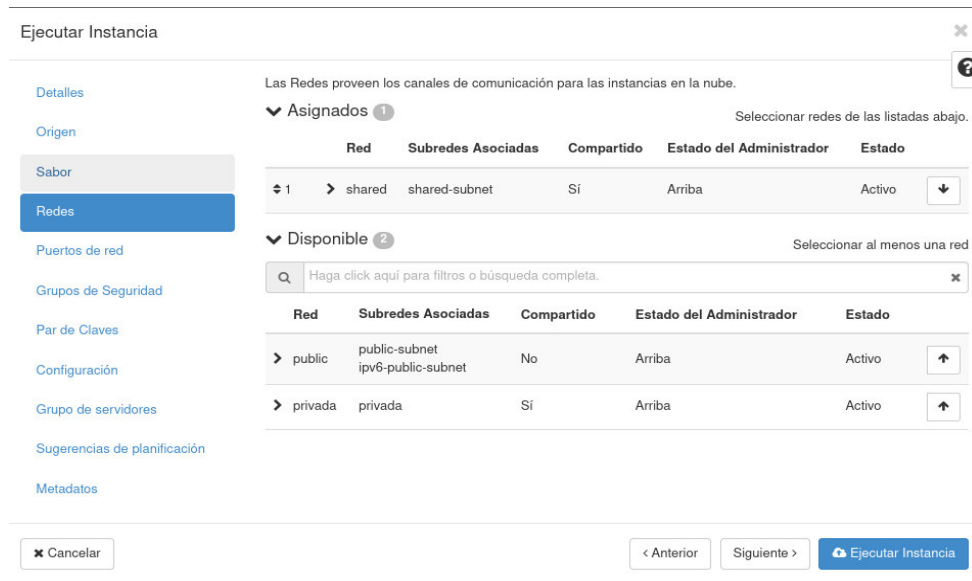
La siguiente opción que se observa en la Figura 3.30 es “Sabor”, que hace referencia al tamaño de la instancia tanto en RAM como en CPU y almacenamiento. Se puede escoger varias opciones dependiendo del uso que se desee. Para este caso se eligió un sabor llamado *cirros* con 256MB de RAM, y 1GB de almacenamiento.



**Figura 3.30** Sabor de la instancia

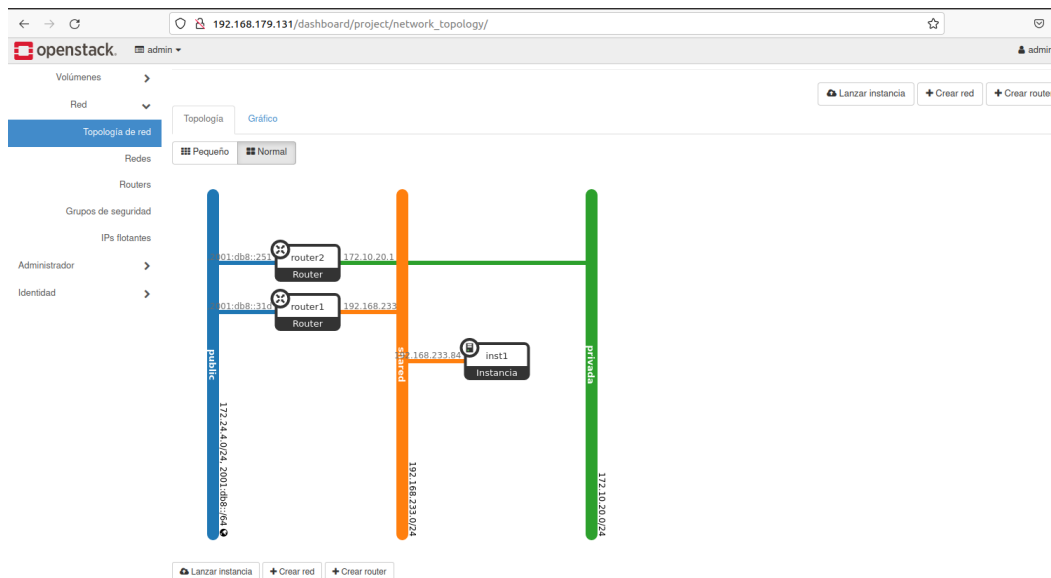
La siguiente opción es la de “Redes”. Aquí se puede seleccionar a qué red se desea que se conecte la instancia. Como se tiene 3 redes, deben salir las 3 redes para

conectarse. En la Figura 3.31 se puede observar que se asignó la red llamada “shared” para que la instancia se conecte a dicha red.



**Figura 3.31** Red de la instancia

Una vez asignada la red, ya se puede ejecutar la instancia. Por lo que se colocó en “Ejecutar instancia” y se pudo observar dentro de la topología de la Figura 3.32 a la instancia conectada a la red a la cual se le asignó.



**Figura 3.32** Instancia creada

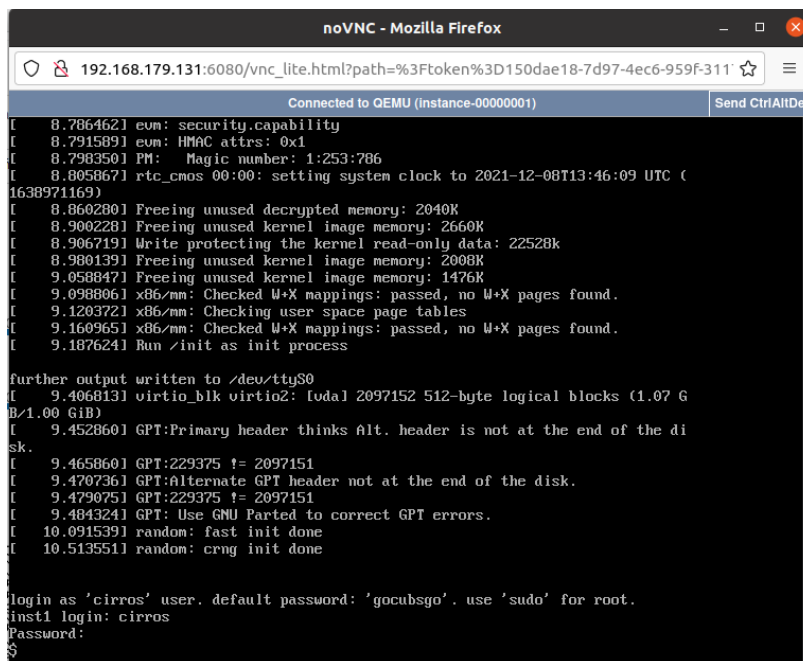
Para poder abrir la instancia creada se debe dar clic en la instancia y seguido dentro de la opción “Abrir consola”. Dentro de la Figura 3.33 se puede observar dicha opción además de la de “Borrar interfaz” que sirve para eliminar la instancia de la red. También

se puede observar el estado, cuando esta esté en activa, la consola se podrá abrir caso contrario no se abrirá.



**Figura 3.33** Abrir consola de la instancia

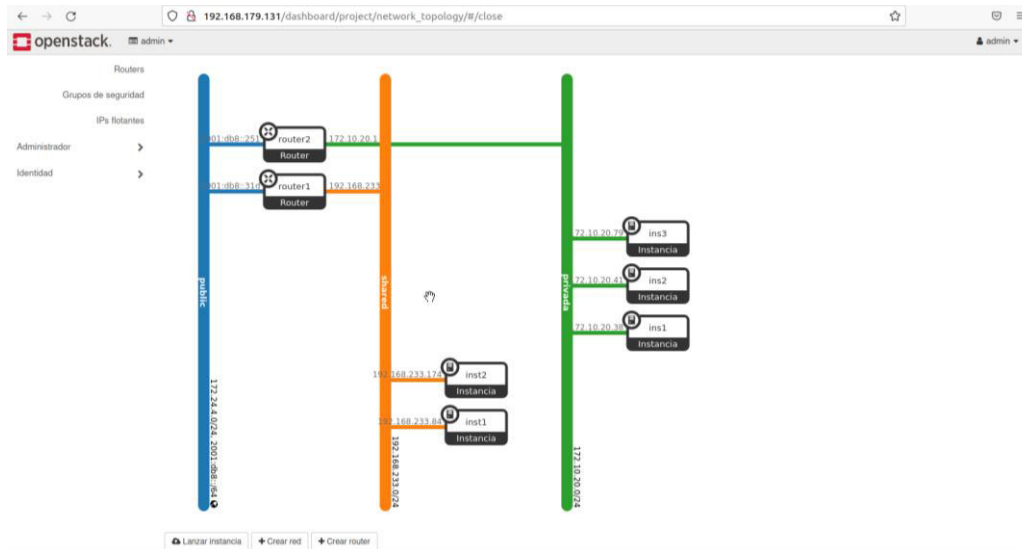
Una vez abierta la consola, solo se debe esperar a que cargue, al final dará ciertas credenciales para poder ingresar a la consola como se puede observar en la Figura 3.34 y eso sería todo para la creación de instancias.



**Figura 3.34** Consola abierta de la Instancia

Posteriormente, se crearon más instancias en las redes, de la misma manera como se creó la primera instancia, lo único en lo que varió es en la asignación de la red ya que algunas se conectaban a la red llamada “privada” y otras a la red llamada “shared”, sin embargo, su creación es igual. En la Figura 3.35 se puede observar la topología de una red con varias instancias creadas.



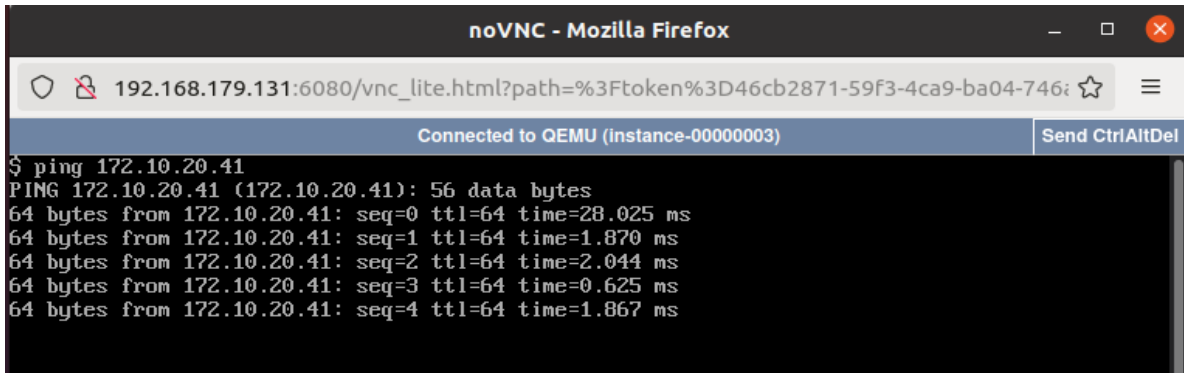


**Figura 3.35** Instancias creadas

Seguido se comprobó que exista comunicación entre las instancias por lo que se realizó una prueba de ping y para ello se abrieron las ventanas de comandos de 2 instancias de máquinas virtuales. En la Figura 3.36 se observa las instancias abiertas.

**Figura 3.36** Consola de 2 instancias de máquinas virtuales

Finalmente colocó la dirección IP de una instancia de máquina virtual en una consola y se verificó que el ping sea exitoso. La dirección IP que se colocó fue la 172.10.20.41 y la prueba de ping exitosa se puede verificar en la Figura 3.37.

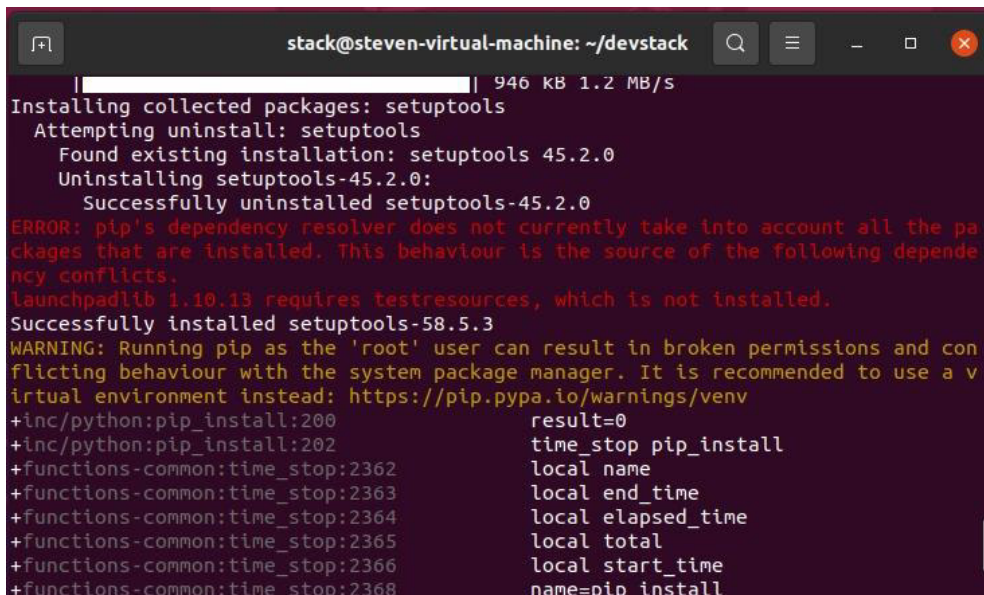


```
noVNC - Mozilla Firefox
192.168.179.131:6080/vnc_lite.html?path=%3Ftoken%3D46cb2871-59f3-4ca9-ba04-746:
Connected to QEMU (instance-00000003) Send CtrlAltDel
$ ping 172.10.20.41
PING 172.10.20.41 (172.10.20.41): 56 data bytes
64 bytes from 172.10.20.41: seq=0 ttl=64 time=28.025 ms
64 bytes from 172.10.20.41: seq=1 ttl=64 time=1.870 ms
64 bytes from 172.10.20.41: seq=2 ttl=64 time=2.044 ms
64 bytes from 172.10.20.41: seq=3 ttl=64 time=0.625 ms
64 bytes from 172.10.20.41: seq=4 ttl=64 time=1.867 ms
```

Figura 3.37 Ping exitoso

### Error al momento de Instalar OpenStack

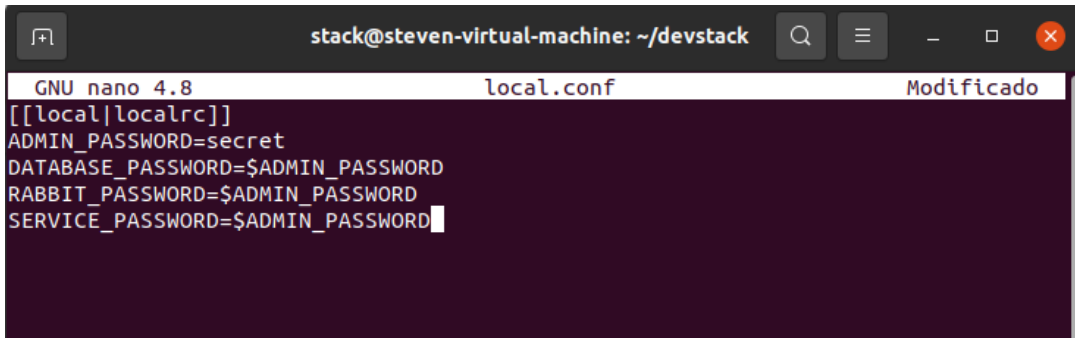
Un error que se presentó al momento de la instalación de OpenStack fue cuando se ejecutaba el *script* "stack.sh". Ahí se solicitaba que se debía descargar ciertos complementos para que pueda funcionar la *Cloud*, sin embargo, ese no era el error ya que se instalaban dichos complementos y aun así no funcionaba la instalación. En la Figura 3.38 se puede visualizar el error con el párrafo en color rojo.



```
stack@steven-virtual-machine: ~/devstack
Installing collected packages: setuptools
  Attempting uninstall: setuptools
    Found existing installation: setuptools 45.2.0
    Uninstalling setuptools-45.2.0:
      Successfully uninstalled setuptools-45.2.0
ERROR: pip's dependency resolver does not currently take into account all the pa
ckages that are installed. This behaviour is the source of the following depende
ncy conflicts:
launchpadlib 1.10.13 requires testresources, which is not installed.
Successfully installed setuptools-58.5.3
WARNING: Running pip as the 'root' user can result in broken permissions and con
flicting behaviour with the system package manager. It is recommended to use a v
irtual environment instead: https://pip.pypa.io/warnings/venv
+inc/python: pip_install:200 result=0
+inc/python: pip_install:202 time_stop pip_install
+functions-common: time_stop:2362 local name
+functions-common: time_stop:2363 local end_time
+functions-common: time_stop:2364 local elapsed_time
+functions-common: time_stop:2365 local total
+functions-common: time_stop:2366 local start_time
+functions-common: time_stop:2368 name=pip install
```

Figura 3.38 Error en la instalación de OpenStack

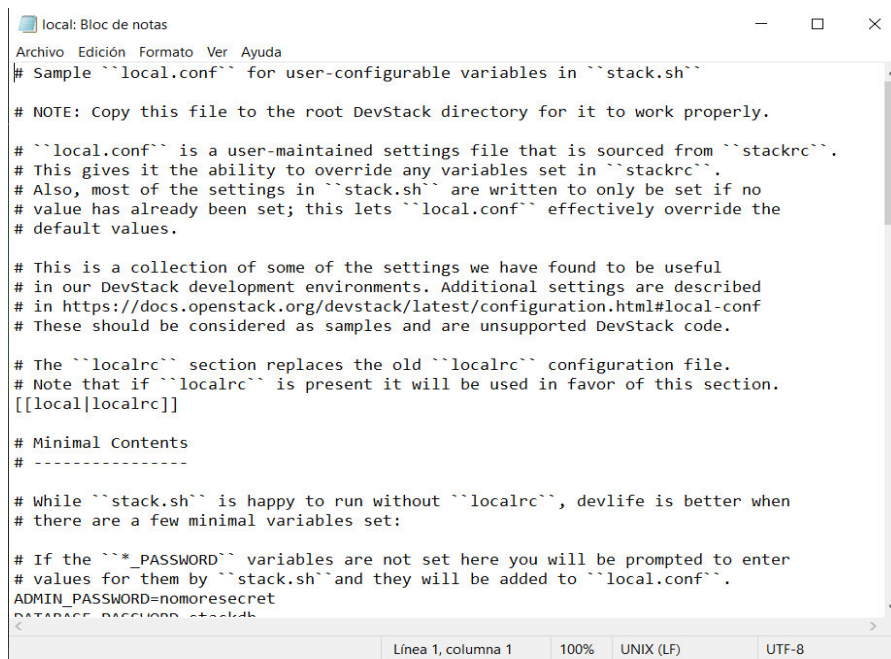
Este error se debía a que el archivo que previamente se creó (*local.conf*) estaba mal creado, es decir que las configuraciones que se colocó estaban mal por lo que al momento de instalar el OpenStack no iba a funcionar. Las palabras que se pueden observar en la Figura 3.39 corresponden a lo que se colocó en el archivo y seguido se procedió a instalar dando como resultado el error de la Figura 3.38.



```
stack@steven-virtual-machine: ~/devstack
GNU nano 4.8 local.conf Modificado
[[local|localrc]]
ADMIN_PASSWORD=secret
DATABASE_PASSWORD=$ADMIN_PASSWORD
RABBIT_PASSWORD=$ADMIN_PASSWORD
SERVICE_PASSWORD=$ADMIN_PASSWORD
```

**Figura 3.39** Archivo con configuraciones erróneas

Para solucionar el error lo que se hizo fue descargar un archivo llamado “local.conf” de la página de DevStack. Este archivo tenía todas las configuraciones mínimas que se requieren para instalar el OpenStack. Dentro de la Figura 3.40 se observa todas las configuraciones que se copió para colocarla dentro del archivo creado.



```
local: Bloc de notas
Archivo Edición Formato Ver Ayuda
# Sample ``local.conf`` for user-configurable variables in ``stack.sh``

# NOTE: Copy this file to the root DevStack directory for it to work properly.

# ``local.conf`` is a user-maintained settings file that is sourced from ``stackrc``.
# This gives it the ability to override any variables set in ``stackrc``.
# Also, most of the settings in ``stack.sh`` are written to only be set if no
# value has already been set; this lets ``local.conf`` effectively override the
# default values.

# This is a collection of some of the settings we have found to be useful
# in our DevStack development environments. Additional settings are described
# in https://docs.openstack.org/devstack/latest/configuration.html#local-conf
# These should be considered as samples and are unsupported DevStack code.

# The ``localrc`` section replaces the old ``localrc`` configuration file.
# Note that if ``localrc`` is present it will be used in favor of this section.
[[local|localrc]]

# Minimal Contents
# -----

# While ``stack.sh`` is happy to run without ``localrc``, devlife is better when
# there are a few minimal variables set:

# If the ``*_PASSWORD`` variables are not set here you will be prompted to enter
# values for them by ``stack.sh`` and they will be added to ``local.conf``.
ADMIN_PASSWORD=nomoresecret
DATABASE_PASSWORD=stacksh
```

**Figura 3.40** Archivo “local.conf” descargado

Todo el contenido del archivo de la Figura 3.40 se copió en el “local.conf” creado en la máquina virtual de Ubuntu tal como se detalla en la Figura 3.41. Una vez copiadas las configuraciones en el archivo se procedió a guardar dicho archivo y posteriormente se procedió a ejecutar el *script* “stack.sh”. Una vez ejecutado, solo fue cuestión de esperar a que terminara la instalación y como todo estaba correcto la instalación fue todo un éxito.

```
stack@steven-virtual-machine: ~/devstack
GNU nano 4.8 local.conf
# Sample ``local.conf`` for user-configurable variables in ``stack.sh``

# NOTE: Copy this file to the root DevStack directory for it to work properly.

# ``local.conf`` is a user-maintained settings file that is sourced from ``stack.sh``.
# This gives it the ability to override any variables set in ``stackrc``.
# Also, most of the settings in ``stack.sh`` are written to only be set if no
# value has already been set; this lets ``local.conf`` effectively override the
# default values.

# This is a collection of some of the settings we have found to be useful
# in our DevStack development environments. Additional settings are described
# in https://docs.openstack.org/devstack/latest/configuration.html#local-conf
# These should be considered as samples and are unsupported DevStack code.

# The ``localrc`` section replaces the old ``localrc`` configuration file.
# Note that if ``localrc`` is present it will be used in favor of this section.
[[local|localrc]]

# Minimal Contents

[ 100 líneas leídas ]
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Texto ^J Justificar ^G Posición
^X Salir ^R Leer fichero ^\ Reemplazar ^U Pegar ^T Ortografía ^_ Ir a línea
```

Figura 3.41 Archivo correcto

## 4 CONCLUSIONES

- El estudio de *Cloud Computing* ha permitido entender cómo funcionan los servicios dentro de la nube y sobre todo entender los diferentes tipos de nubes que se pueden implementar en las diversas plataformas.
- La plataforma de OpenStack resultó una herramienta apropiada y completa para realizar los diferentes servicios, además de que su interfaz fue sencilla de utilizar haciendo que la red de datos creada se pueda gestionar de manera conjunta con todas las instancias de máquinas virtuales para así verificar fácilmente errores que podrían suscitarse.
- Mediante la implementación de la plataforma OpenStack se logró comprender el uso de una *Cloud Computing*, además de todos los servicios que esta puede ofrecer a los usuarios para que realicen sus proyectos y satisfacer sus necesidades.
- La investigación realizada de las diferentes plataformas de *Cloud Computing* ha ayudado a diferenciar los diferentes modelos de servicio que se pueden tener, además de que permitió seleccionar una para su implementación para así desarrollar ciertos proyectos como la creación de instancias de máquinas virtuales y la creación de redes.
- Con el uso del módulo *Horizon* el cual proporciona la *dashboard* se pudo tener mayor facilidad para la creación de instancias, implementación de los *routers* y de las redes, además proporcionó un mayor control para la red pues permitió observar la topología de la red de modo que se podía gestionar las instancias y verificar que las direcciones estén correctas.
- Gracias al uso de las instancias y de su consola de comandos se pudo verificar que existía conectividad entre instancias y de igual forma se pudo verificar que las direcciones IP las cuales fueron asignadas mediante DHCP estén correctamente fijadas con las configuraciones que se realizaron previamente en la creación de las subredes.

## 5 RECOMENDACIONES

- Se recomienda utilizar la máquina virtual de Ubuntu 20.04 para la instalación de OpenStack ya que en esta versión se tiene mayor estabilidad y además la plataforma ha sido experimentada en dicha versión por lo que el funcionamiento será más fluido.
- Se debe tener en cuenta antes de instalar el OpenStack que la virtualización en el *host* esté activada caso contrario se debe activar dicha opción y para poder activarla se tiene que acceder a la BIOS y para ello va a depender de la marca del computador.
- Durante la instalación de OpenStack es importante colocar correctamente los datos dentro del archivo "local.conf" y para ello se puede encontrar un archivo de ejemplo dentro de la página de Devstack con todo lo que se debe copiar para que la instalación sea correcta.
- Se recomienda siempre verificar la versión de OpenStack que se instala puesto que a veces algunos comandos cambian dependiendo de la versión que se utilice.
- Es recomendable ingresar con las credenciales de "admin" para OpenStack ya que así se puede tener todo el control de la plataforma y de todas sus herramientas internas.
- Se recomienda utilizar un computador que tenga los recursos necesarios para poder simular una máquina virtual ya que así no se tendrá inconvenientes al momento de instalar y utilizar el OpenStack ya que debido a esto se puede tener lentitud al momento de realizar las pruebas de funcionamiento.
- Se debería tener una asignatura específica para el estudio de *Cloud Computing* dentro de la carrera ya que permitiría obtener gran conocimiento acerca de esta tecnología.

## 6 REFERENCIAS BIBLIOGRÁFICAS

- [1] VMware, «VMware,» [En línea]. Available: <https://www.vmware.com/latam/solutions/virtualization.html>. [Último acceso: 03 01 2022].
- [2] Nutanix, «Nutanix,» [En línea]. Available: <https://www.nutanix.com/es/info/hypervisor>. [Último acceso: 18 12 2021].
- [3] Three Points, «Three Points,» 19 05 2020. [En línea]. Available: <https://www.threepoints.com/int/cloud-computing-principales-proveedores-y-casos-de-exito>. [Último acceso: 19 12 2021].
- [4] FORLOPD, «FORLOPD,» 23 09 2021. [En línea]. Available: <https://forlopd.es/el-cloud-computing-caracteristicas-y-tipologia/>. [Último acceso: 10 11 2021].
- [5] EALDE, «EALDE Business School,» 30 03 2019. [En línea]. Available: <https://www.ealde.es/cloud-computing-caracteristicas-funcionamiento/>. [Último acceso: 11 11 2021].
- [6] F. Flores, «OpenWebinars,» 22 03 2021. [En línea]. Available: <https://openwebinars.net/blog/tipos-de-cloud-computing/>. [Último acceso: 13 11 2021].
- [7] A. B. Carrillo, «Blog de Viafirma,» 05 09 2019. [En línea]. Available: <https://www.viafirma.com/blog-xnoccio/es/tipos-servicios-cloud-computing-empresas/>. [Último acceso: 15 11 2021].
- [8] StackScale, «StackScale,» 04 08 2021. [En línea]. Available: <https://www.stackscale.com/es/blog/modelos-de-servicio-cloud/>. [Último acceso: 19 11 2021].
- [9] Alba Fernández, «AuraQuantic,» [En línea]. Available: <https://www.auraquantic.com/es/cloud-computing-y-las-diferencias-entre-iaas-paas-y-saas/>. [Último acceso: 20 11 2021].

- [10] «Red Hat,» [En línea]. Available: <https://www.redhat.com/es/topics/openstack>. [Último acceso: 24 11 2021].
- [11] «Rackspace Technology,» [En línea]. Available: <https://www.rackspace.com/es/library/what-is-openstack>. [Último acceso: 24 11 2021].
- [12] M. Serrano, «Virtualiza desde Zero,» 10 05 2021. [En línea]. Available: <https://virtualizadesdezero.com/que-es-openstack/>. [Último acceso: 24 11 2021].
- [13] «IONOS Digitalguide,» 24 08 2020. [En línea]. Available: <https://www.ionos.es/digitalguide/servidores/herramientas/que-es-openstack/>. [Último acceso: 26 11 2021].
- [14] «Aesis Technologies,» [En línea]. Available: <https://www.aesis.la/infraestructura-it-en-la-nube>. [Último acceso: 27 11 2021].
- [15] D. G. Orozco, «OpenStack: una alternativa de Infraestructura como servicio para instituciones de educación superior,» *Revista de Investigación en Tecnologías de la Información*, vol. 8, nº 15, pp. 1-7, 2020.
- [16] F. M. Villaverde, «OpenNebula y Hadoop: Cloud Computing con herramientas Open Source,» 2012.
- [17] A. P. Paniagua, «Adictos al trabajo,» 04 11 2015. [En línea]. Available: <https://www.adictosaltrabajo.com/2015/11/04/tutorial-de-opennebula-jugando-con-opennebula/>. [Último acceso: 26 11 2021].
- [18] «uCloudStore,» [En línea]. Available: <https://ucloudstore.com/google-cloud-platform/>. [Último acceso: 27 11 2021].
- [19] Ignacio Ordorica, «Incentro,» 19 08 2020. [En línea]. Available: <https://www.incentro.com/es-es/blog/stories/que-es-google-cloud-platform/>. [Último acceso: 27 11 2021].
- [20] L. Gracia, «Un poco de Java,» 25 11 2011. [En línea]. Available: <https://unpocodejava.com/2011/11/25/que-es-cloud-foundry/>. [Último acceso: 28 11 2021].



## 7 ANEXOS

### ANEXO I

#### CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 16 de febrero de 2022

De mi consideración:

Yo, FERNANDO VINICIO BECERRA CAMACHO, en calidad de Director del Trabajo de Integración Curricular titulado ESTUDIO E IMPLEMENTACIÓN DE UNA *CLOUD COMPUTING* asociado a la IMPLEMENTACIÓN DE *CLOUD COMPUTING* UTILIZANDO HERRAMIENTAS *OPEN SOURCE* elaborado por el estudiante STEVEN MAURICIO HERRERA HERNÁNDEZ de la carrera en TECNOLOGÍA SUPERIOR EN REDES Y TELECOMUNICACIONES, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 7%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin.

[https://epnecuador-my.sharepoint.com/:b/g/personal/fernando\\_becerrac\\_epn\\_edu\\_ec/EQpGS8LQsRRGnAFmEmEpvwQBnu5A\\_s6Z3CHLx-aPQeFzjA?e=V1Zqd2](https://epnecuador-my.sharepoint.com/:b/g/personal/fernando_becerrac_epn_edu_ec/EQpGS8LQsRRGnAFmEmEpvwQBnu5A_s6Z3CHLx-aPQeFzjA?e=V1Zqd2)

Atentamente,



**Fernando Vinicio Becerra Camacho**

**Profesor Ocasional a Tiempo Completo**

**ESFOT**

## ANEXO II

Evidencia practica enlace del video

