

**UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE
TELECOMUNICACIÓN**

**COMPUTACIÓN EN NUBE Y VIRTUALIZACIÓN DE REDES Y
SERVICIOS**

**Trabajo Final: Despliegue automático de una aplicación escalable
sobre una nube OpenStack utilizando los servicios de orquestación
de Terraform**

Alumno: Alex David Segovia Ancajima

Curso: 2025-26

En el presente trabajo se explicará la solución de topología desplegada para una aplicación escalable sobre una nube OpenStack, utilizando Terraform como servicio de orquestación.

La topología a desplegar consta de varios componentes de cómputo y de red, entre los cuales se encuentran:

- Tres servidores web (escalables)
- Un sistema de base de datos
- Un sistema de almacenamiento de ficheros/objetos
- Dos redes internas
- Un router con reglas de firewall
- Un balanceador de carga
- Una máquina virtual de administración

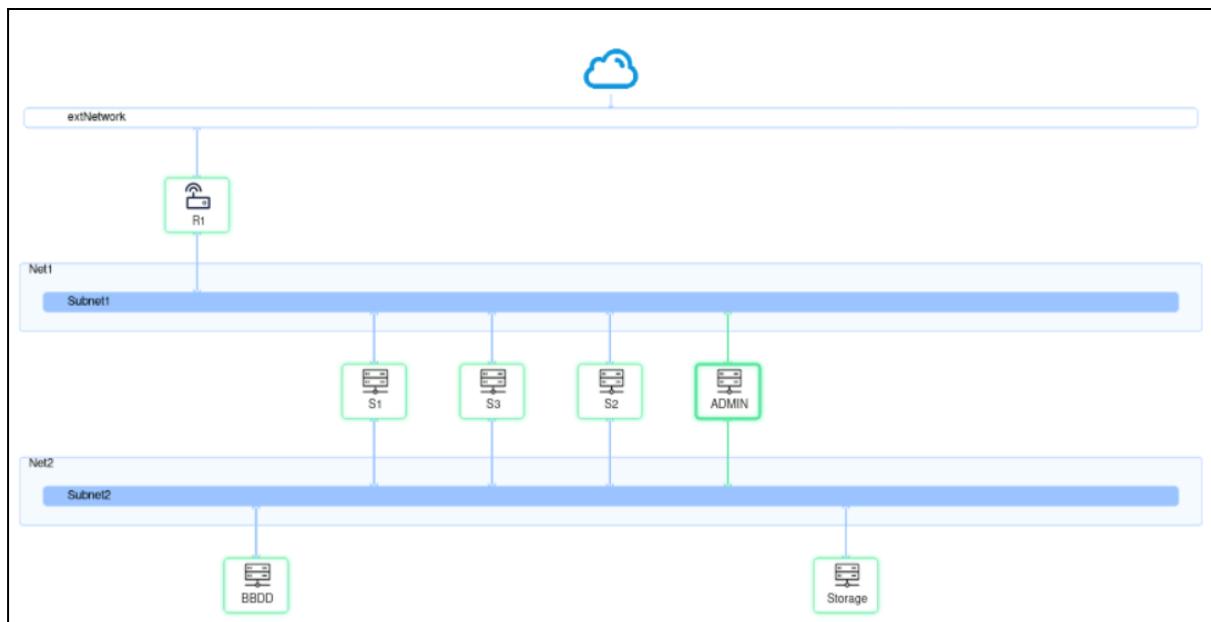


Figura 1: Topología desplegada

Para desplegar esta topología se utilizó el servicio de orquestación de Terraform y una estructura modular para la creación de los distintos recursos (máquinas virtuales, redes, router, firewall, balanceador de carga, entre otros). Cada módulo consta de un archivo “main.tf”, en el que se definen los recursos que se crearán en cada módulo; otro archivo llamado “variables.tf”, en el que se definen las variables a utilizar en los recursos de cada módulo y un archivo llamado “outputs.tf”, en el cual se especifica la información de cada recurso que se desea exponer, de tal forma que pueda ser utilizada por otros recursos. A continuación se muestra la estructura modular de los archivos de Terraform.

```

    ↴ modules
      ↴ database
        ↴ main.tf
        ↴ variables.tf
      ↴ firewall
        ↴ main.tf
        ↴ variables.tf
      ↴ image
        ↳ database_server_image.qcow2
        ↴ main.tf
        ↴ outputs.tf
        ↴ variables.tf
        ↳ web_server_image.qcow2
      ↴ load_balancer
        ↴ main.tf
        ↴ outputs.tf
        ↴ variables.tf
      ↴ network
        ↴ main.tf
        ↴ outputs.tf
        ↴ variables.tf
    ↴ router
      ↴ main.tf
      ↴ outputs.tf
      ↴ variables.tf
    ↴ security_group
      ↴ main.tf
      ↴ outputs.tf
    ↴ storage
      ↴ main.tf
      ↴ variables.tf
    ↴ vm_admin
      ↴ main.tf
      ↴ outputs.tf
      ↴ variables.tf
    ↴ web_server
      ↴ main.tf
      ↴ outputs.tf
      ↴ variables.tf
    ↴ main.tf
    ↴ variables.tf

```

Figura 2: Estructura de los scripts

Como se puede observar en la estructura, se crearon módulos para cada tipo de componente, de tal manera que se puede controlar cada uno de forma independiente y sin afectar a los demás. En ese sentido, las máquinas virtuales se separaron en módulos para la VM de administración, las VMs para los servidores web, la VM para la base de datos y la VM para el servidor de almacenamiento.

Máquina virtual de administración

Este módulo crea una máquina virtual que se encuentra conectada a las dos redes internas: Net1 y Net2. Además, es la única máquina virtual con dirección IP flotante, por lo que es la única a la que se puede acceder desde internet. Esta máquina virtual es el punto de entrada a la aplicación desplegada y a partir de la cual se puede acceder mediante SSH hacia los servidores web, el servidor de base de datos y el servidor de almacenamiento.

Además, es la única máquina virtual que tiene asociado un par de claves para poder acceder sin la necesidad de ingresar la contraseña. Cabe mencionar que este par de claves se creó con permisos de lectura para todos los usuarios, debido a que el usuario que lo crea es el root del nodo de Terraform, mientras que el que finalmente lo usa es el usuario de la máquina host; por lo que sin esta modificación en los permisos, no se podría utilizar el par de claves para acceder por SSH a la máquina virtual de administración. Por último, en el archivo sshd_config, se modificó el puerto

por defecto para SSH en esta máquina virtual, el cual pasó a ser el 2025 en lugar del puerto 22.

Servidores web

Este módulo crea una determinada cantidad de servidores web, los cuales se encuentran conectados a las redes Net1 y Net2. Estas máquinas virtuales no tienen asignada una dirección IP flotante, por lo que solo se puede acceder a ellas a través de la VM de administración.

Considero importante mencionar que estos servidores necesitan estar conectados a la red Net1, ya que el balanceador de carga se encuentra conectado a esa red y es necesario que agregue a los servidores web como miembros, por lo que deben encontrarse en la misma red. Asimismo, es importante que se encuentre conectada a la red Net2, ya que el servidor de base de datos y el servidor de almacenamiento se encuentran conectados a esa red y es necesario que los servidores web tengan acceso a los servicios que brindan.

Para reducir todo lo posible el tiempo de ejecución del cloud-init en los servidores web, se instalaron los paquetes de nginx y mariadb-client previamente, para luego descargar un snapshot y utilizarlo como imagen para desplegar finalmente cada servidor web.

Base de datos

Este módulo crea un servidor de base de datos, el cual se encuentra conectado a la red Net2. Esta red no se encuentra directamente conectada al router que brinda salida a internet, lo cual ofrece una mayor seguridad a la base de datos, ya que solo se podrá acceder a este servidor desde los servidores web y la máquina virtual de administración.

De forma similar a los servidores web, con la finalidad de reducir todo lo posible el tiempo de ejecución del cloud-init en el servidor de base de datos, se instaló el paquete de mariadb-server previamente. Luego se descargó un snapshot y se utilizó como imagen base para desplegar el servidor de base de datos.

Servidor de almacenamiento

Este módulo crea un servidor de almacenamiento, el cual se encuentra conectado a la red Net2. De forma similar a la base de datos, se encuentra protegida ya que no es accesible directamente desde internet, ya que la red no se encuentra conectada al router.

Redes internas

Este módulo crea dos redes internas: Net1 y Net2, así como sus respectivas subredes. Por un lado la red Net1 tiene dirección de red 10.1.1.0/24 y la red Net2

tiene dirección de red 10.1.2.0/24. Cada una tiene un pool de direcciones IP que se pueden asignar a las máquinas virtuales que se conecten.

Router

Este módulo crea un router que se encuentra conectado a la red Net1 y a la red externa. Este es el encargado de dar conexión a Internet y es el que realiza el SNAT y DNAT.

Firewall

Este módulo crea las reglas, políticas y un grupo de firewall que se aplican sobre el router. Esta es la capa de seguridad a nivel de red que tiene la topología desplegada, ya que se encarga de permitir el tráfico HTTP hacia la dirección IP flotante del balanceador de carga, el tráfico por el puerto 2025 de la dirección IP flotante de la máquina virtual de administración y permitir el cualquier tráfico proveniente de la red interna Net1 (tráfico hacia el exterior de la topología).

Balanceador de carga

Este módulo crea un balanceador de carga que se encuentra conectado a la red Net1. Este componente es externo al router o a las máquinas virtuales que se encuentran conectadas a la red Net1 y, así como la VM de administración, también tiene una dirección IP flotante, desde la cual se podrá acceder a la aplicación web. Para configurar esto, no solo se creó el balanceador de carga asociado a la red Net1, sino que también se creó un “listener”, el cual escucha las solicitudes que llegan a la dirección IP flotante por el puerto 80 (TCP) y se encuentra asociado al balanceador de carga creado previamente. Además, se crea un “pool”, el cual contiene como miembros a los servidores web y balancea la carga de las solicitudes HTTP entre ellos empleando el método Round Robin; para ello, este pool se asocia al listener creado previamente.

Grupos de seguridad

Este módulo crea un grupo de seguridad llamado “open” que estará asociado a todas las máquinas virtuales (servidores web, base de datos, almacenamiento de archivos y administración). En este grupo de seguridad se definen reglas que permiten todo el tráfico TCP entrante y saliente de las máquinas virtuales.

Imágenes

Este módulo se usó para crear las imágenes de los servidores web y la base de datos, en base a los archivos .qcow2 que se crearon previamente; estos archivos tienen instalados los paquetes que requieren los servidores web y la base de datos, de tal manera que se reduce al mínimo el tiempo de ejecución del cloud-init en el primer encendido de cada máquina virtual.

Pruebas de funcionamiento de la topología desplegada

Para realizar pruebas de funcionamiento de la topología desplegada, en primer lugar es necesario conocer las direcciones IP para cada máquina virtual y el balanceador de carga. Para esto se desplegó la topología con la cantidad de servidores web y regla de firewall para SSH por defecto (3 servidores web y permitir el acceso por SSH al puerto 2025).

```
alex.segoviaancajima@1117: $ openstack server list
+-----+-----+-----+-----+-----+
| ID   | Name | Status | Networks          | Image      | Flavor   |
+-----+-----+-----+-----+-----+
| e255a8c8-441a-4992-9fc5-9acc70851666 | S2    | ACTIVE | Net1=10.1.1.28; Net2=10.1.2.23 | web_server_image | m1.smaller |
| 93877bd2-449e-45af-bfc7-b70f7f70f3ec | S1    | ACTIVE | Net1=10.1.1.40; Net2=10.1.2.49 | web_server_image | m1.smaller |
| 982c4afe-9826-47c3-9409-716516e8c1e1 | S3    | ACTIVE | Net1=10.1.1.73; Net2=10.1.2.66 | web_server_image | m1.smaller |
| fa5061de-5188-4bb8-b075-87ce12b4f28 | BBDD  | ACTIVE | Net2=10.1.2.71                  | database_server_image | m1.smaller |
| c17c2225-612a-424f-87c1-82856f4180c1 | ADMIN | ACTIVE | Net1=10.0.10.187, 10.1.1.7; Net2=10.1.2.77 | jammy-server-cloudimg-amd64-vnx | m1.smaller |
| daa74cac-8ebf-4779-9d86-f250c7c8dbc0 | Storage | ACTIVE | Net2=10.1.2.48                  | jammy-server-cloudimg-amd64-vnx | m1.smaller |
+-----+-----+-----+-----+-----+
alex.segoviaancajima@1117: $ openstack floating ip list
+-----+-----+-----+-----+-----+
| ID       | Floating IP Address | Fixed IP Address | Port | Floating Network | Project |
+-----+-----+-----+-----+-----+
| 65486750-82ad-462f-ad38- | 10.0.10.194        | 10.1.1.61         |      | 2378c1c8-f8aa-4901-a5e2- | 827eed76241c4de6af4b4f417e6384 |
| eafb3093ce2b |                   |                   |      | 0d70c071b671 | bb |
| 8ca55c60-d1f9-4c6e-a656- | 10.0.10.187        | 10.1.1.7          |      | 2378c1c8-f8aa-4901-a5e2- | 827eed76241c4de6af4b4f417e6384 |
| 8555520983b8 |                   |                   |      | 42a0ecb58c6b | bb |
+-----+-----+-----+-----+-----+
```

Figura 3: Direcciones IP de las máquinas virtuales y el balanceador de carga

Se puede observar que solamente existen dos direcciones IP flotantes, de las cuales una corresponde a la VM de administración y la otra corresponde al balanceador de carga.

En la siguiente tabla se muestra el resumen de direcciones IP (estas direcciones cambiarán si se vuelve a desplegar la topología, por lo que solo se anotarán para realizar las pruebas de funcionamiento).

| | Net1 | | Net2 |
|----------------------|------------|-------------|------------|
| | IP privada | IP flotante | IP privada |
| ADMIN | 10.1.1.7 | 10.0.10.187 | 10.1.2.77 |
| S1 | 10.1.1.40 | - | 10.1.2.49 |
| S2 | 10.1.1.28 | - | 10.1.2.23 |
| S3 | 10.1.1.73 | - | 10.1.2.66 |
| BBDD | - | - | 10.1.2.71 |
| Storage | - | - | 10.1.2.48 |
| Load balancer | 10.1.1.61 | 10.0.10.194 | - |

Tabla 1: Resumen de direcciones IP de la topología desplegada

La primera prueba realizada fue la del acceso por SSH a la VM de administración. Cabe destacar que la conexión por SSH solo funciona en el puerto 2025, por lo que se debe especificar este al momento de realizar la conexión. Caso contrario, fallará

por dos motivos: la regla de firewall que abrió el puerto 2025 y no el puerto 22, y la configuración del archivo `sshd_config` en la VM de administración, reemplazando el puerto 22 por el puerto 2025.

```
alex.segoviaancajima@1117: $ ssh -i /mnt/tmp/openstack_lab-antelope_4n_classic_ovs-v04/shared/keys/admin.pem root@10.0.10.187
ssh: connect to host 10.0.10.187 port 22: Connection timed out
alex.segoviaancajima@1117: $ ssh -i /mnt/tmp/openstack_lab-antelope_4n_classic_ovs-v04/shared/keys/admin.pem root@10.0.10.187 -p 2025
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-71-generic x86_64)

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

System information as of Sun Nov 16 16:55:24 UTC 2025

System load: 0.0           Processes:      88
Usage of /: 52.4% of 2.73GB Users logged in:    0
Memory usage: 39%          IPv4 address for ens3: 10.1.1.7
Swap usage:  0%           IPv4 address for ens4: 10.1.2.77

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Sun Nov 16 16:37:07 2025 from 10.0.10.1
root@ADMIN:~#
```

Figura 4: Prueba de conexión por SSH hacia la VM de administración

La segunda prueba realizada fue la del acceso por SSH desde la máquina virtual de administración hacia los servidores web, el servidor de base de datos y el servidor de almacenamiento. En este caso no se creó un par de claves para estos servidores, por lo que se estableció la conexión por medio de contraseña.

```
root@ADMIN:~# ssh root@10.1.1.40
root@10.1.1.40's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-71-generic x86_64)

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

System information as of Sun Nov 16 16:58:51 UTC 2025

System load: 0.0           Processes:      89
Usage of /: 73.2% of 2.73GB Users logged in:    0
Memory usage: 39%          IPv4 address for ens3: 10.1.1.40
Swap usage:  0%           IPv4 address for ens4: 10.1.2.49

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

270 updates can be applied immediately.
182 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '24.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Nov 16 16:58:51 2025 from 10.1.1.7
root@Servidor-1:~#
```

Figura 5: Conexión por SSH hacia el servidor web 1

```

root@ADMIN:~# ssh root@10.1.1.28
root@10.1.1.28's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-71-generic x86_64)

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

 System information as of Sun Nov 16 17:00:46 UTC 2025

System load: 0.00390625      Processes:          94
Usage of /: 73.2% of 2.73GB  Users logged in:    0
Memory usage: 48%           IPv4 address for ens3: 10.1.1.28
Swap usage:  0%             IPv4 address for ens4: 10.1.2.23

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

270 updates can be applied immediately.
182 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '24.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Nov 16 16:59:04 2025 from 10.1.1.7
root@Servidor-2:~# 
```

Figura 6: Conexión por SSH hacia el servidor web 2

```

root@ADMIN:~# ssh root@10.1.1.73
root@10.1.1.73's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-71-generic x86_64)

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

 System information as of Sun Nov 16 17:01:17 UTC 2025

System load: 0.0          Processes:          90
Usage of /: 73.2% of 2.73GB  Users logged in:    0
Memory usage: 38%          IPv4 address for ens3: 10.1.1.73
Swap usage:  0%            IPv4 address for ens4: 10.1.2.66

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

270 updates can be applied immediately.
182 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '24.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Nov 16 16:59:16 2025 from 10.1.1.7
root@Servidor-3:~# 
```

Figura 7: Conexión por SSH hacia el servidor web 3

```

root@ADMIN:~# ssh root@10.1.2.71
root@10.1.2.71's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-71-generic x86_64)

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

 System information as of Sun Nov 16 17:02:06 UTC 2025

 System load:  0.00244140625   Processes:          93
 Usage of /:   72.6% of 2.73GB  Users logged in:     0
 Memory usage: 58%            IPv4 address for ens3: 10.1.2.71
 Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

270 updates can be applied immediately.
182 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Sun Nov 16 17:02:06 2025 from 10.1.2.77
root@BBDD:~#

```

Figura 8: Conexión por SSH hacia el servidor de base de datos

```

root@ADMIN:~# ssh root@10.1.2.48
root@10.1.2.48's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-71-generic x86_64)

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

 System information as of Sun Nov 16 17:02:17 UTC 2025

 System load:  0.0          Processes:          87
 Usage of /:   51.8% of 2.73GB  Users logged in:     0
 Memory usage: 39%           IPv4 address for ens3: 10.1.2.48
 Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Sun Nov 16 17:02:17 2025 from 10.1.2.77
root@storage:~#

```

Figura 9: Conexión por SSH hacia el servidor de almacenamiento

La tercera prueba que se realizó fue la de la verificación del balanceo de carga. Para esto se utilizó el servicio de OpenStack: Octavia, con el cual se desplegó un平衡ador de carga que tiene como miembros a los servidores web y una dirección IP flotante. En ese sentido, al utilizar el comando curl sobre esa dirección

IP flotante en repetidas ocasiones, el balanceador de carga utiliza el método Round Robin para dirigir las consultas hacia cada servidor web, tal y como se muestra en la siguiente captura de pantalla.

```
alex.segoviaancajima@1117: $ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-1</h1></body></html>
alex.segoviaancajima@1117: $ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-3</h1></body></html>
alex.segoviaancajima@1117: $ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-2</h1></body></html>
alex.segoviaancajima@1117: $ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-1</h1></body></html>
alex.segoviaancajima@1117: $ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-3</h1></body></html>
alex.segoviaancajima@1117: $ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-2</h1></body></html>
alex.segoviaancajima@1117: $ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-1</h1></body></html>
alex.segoviaancajima@1117: $ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-3</h1></body></html>
alex.segoviaancajima@1117: $ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-2</h1></body></html>
alex.segoviaancajima@1117: $ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-1</h1></body></html>
alex.segoviaancajima@1117: $ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-2</h1></body></html>
alex.segoviaancajima@1117: $ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-1</h1></body></html>
alex.segoviaancajima@1117: $ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-3</h1></body></html>
alex.segoviaancajima@1117: $ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-2</h1></body></html>
alex.segoviaancajima@1117: $
```

Figura 10: Consultas a la aplicación web con evidencia del balanceo de carga

La cuarta prueba que se realizó fue la de escalar la cantidad de servidores web que se desplegaron inicialmente. Para esto se modificó la variable “server_count”, la cual tiene un valor por defecto de 3. En esta ocasión se le asignó un valor de 5, por lo que se desplegaron 5 servidores web en total (3 servidores web ya existentes y 2 servidores web nuevos).

```
root@terraform:~/shared# terraform apply -var="server_count=5"
module.security_group_open.openstack_networking_secgroup_v2.my_security_group: Refreshing state... [id=abfa8175-c6da-4024-b64b-a26b8a7db605]
module.net2.openstack_networking_network_v2.net: Refreshing state... [id=13768f9d-5b8f-4150-99f7-c40c27ea21b3]
data.openstack_networking_network_v2.ExtNet: Reading...
module.web_server_image.openstack_images_image_v2.image: Refreshing state... [id=49ff48f0-049c-4094-97d7-07dc28a586ad]
module.database_server_image.openstack_images_image_v2.image: Refreshing state... [id=2ea5842b-8d9d-46bc-8754-65095e02985a]
module.net1.openstack_networking_network_v2.net: Refreshing state... [id=a7c8acce-7de4-4ff3-9b14-8a235ce227d7]
module.security_group_open.openstack_networking_secgroup_rule_v2.security_group_rule_egress: Refreshing state... [id=2a617b43-f3b1-4cc6-a5c0-2d4e42aad28b]
module.security_group_open.openstack_networking_secgroup_rule_v2.security_group_rule_ingress: Refreshing state... [id=46d3e36b-9c98-4be7-a8a8-e3c74a099653]
module.net2.openstack_networking_subnet_v2.subnet: Refreshing state... [id=a43f09c6-9f5a-42ac-bd97-2724bf218a2]
```

Figura 11: Despliegue escalable de servidores web

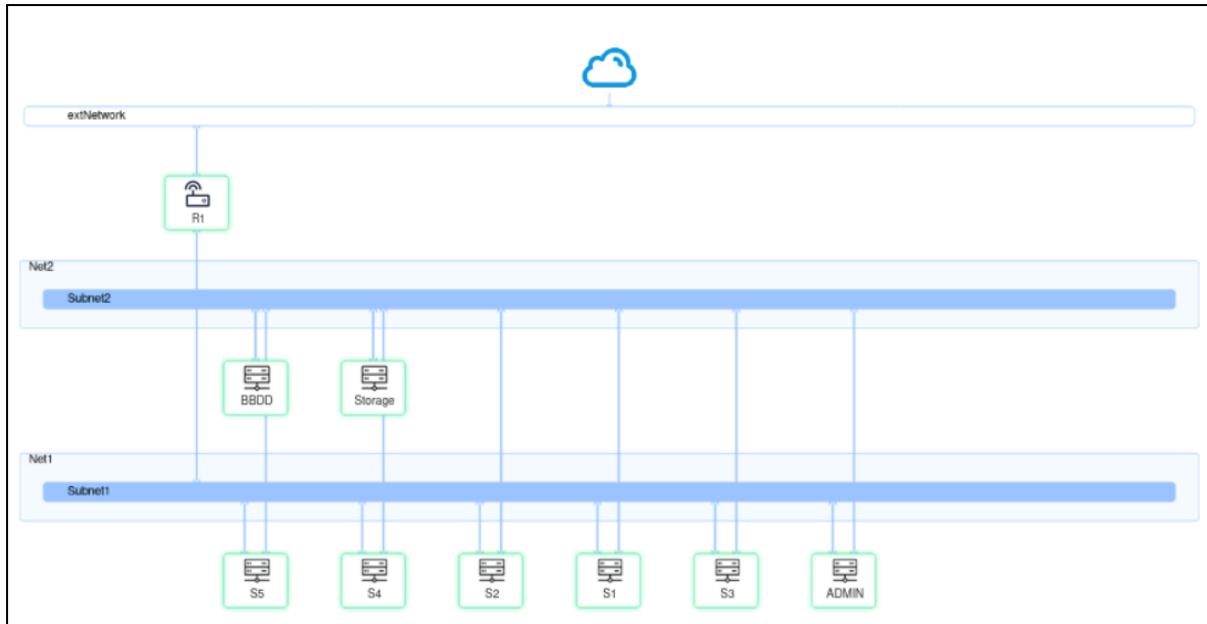


Figura 12: Topología del despliegue con 5 servidores web

Cabe mencionar que los servidores web nuevos que se agregan a la topología también se agregan como miembros al balanceador de carga. Por lo tanto, al utilizar el comando curl sobre la dirección IP flotante del balanceador de carga, se evidencia que también dirige las consultas hacia los nuevos servidores web.

```

alex.segoviaancajima@l117:~$ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-1</h1></body></html>
alex.segoviaancajima@l117:~$ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-5</h1></body></html>
alex.segoviaancajima@l117:~$ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-4</h1></body></html>
alex.segoviaancajima@l117:~$ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-3</h1></body></html>
alex.segoviaancajima@l117:~$ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-2</h1></body></html>
alex.segoviaancajima@l117:~$ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-1</h1></body></html>
alex.segoviaancajima@l117:~$ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-5</h1></body></html>
alex.segoviaancajima@l117:~$ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-4</h1></body></html>
alex.segoviaancajima@l117:~$ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-3</h1></body></html>
alex.segoviaancajima@l117:~$ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-2</h1></body></html>
alex.segoviaancajima@l117:~$ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-1</h1></body></html>
alex.segoviaancajima@l117:~$ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-5</h1></body></html>
alex.segoviaancajima@l117:~$ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-4</h1></body></html>
alex.segoviaancajima@l117:~$ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-3</h1></body></html>
alex.segoviaancajima@l117:~$ curl 10.0.10.194
<html><body><h1>Bienvenido al Servidor-2</h1></body></html>

```

Figura 13: Consultas a la aplicación web con evidencia del balanceo de carga

La quinta prueba que se realizó fue la de que los servidores web se puedan conectar al servidor de base de datos. Si bien es cierto la aplicación web no utiliza la base de datos, sus servidores sí pueden conectarse como clientes mediante la

dirección IP, el usuario y la contraseña creados. En las siguientes capturas de pantalla se muestra que cada servidor web puede conectarse a la base de datos.

```
root@Servidor-1:~# mysql -h 10.1.2.71 -u web_server -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 39
Server version: 10.6.22-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> █
```

Figura 14: Conexión a la base de datos desde el servidor web 1

```
root@Servidor-2:~# mysql -h 10.1.2.71 -u web_server -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 40
Server version: 10.6.22-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> █
```

Figura 15: Conexión a la base de datos desde el servidor web 2

```
root@Servidor-3:~# mysql -h 10.1.2.71 -u web_server -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 41
Server version: 10.6.22-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> █
```

Figura 16: Conexión a la base de datos desde el servidor web 3

La sexta prueba que se realizó fue la de verificar las políticas de firewall, específicamente las externas; es decir, la que permite el tráfico HTTP (puerto 80) de la dirección IP flotante del balanceador de carga y la que permite el tráfico hacia el puerto 2025 de la dirección IP flotante de la VM de administración (puerto que será utilizado para la conexión por SSH).

```
alex.segoviaancajima@l117:~$ nc -zv -w 2 10.0.10.194 80
Connection to 10.0.10.194 80 port [tcp/http] succeeded!
alex.segoviaancajima@l117:~$ nc -zv -w 2 10.0.10.187 2025
Connection to 10.0.10.187 2025 port [tcp/*] succeeded!
```

Figura 17: Verificación de las políticas de Firewall

La séptima prueba que se realizó fue la de bloquear el acceso por SSH a la VM de administración. Esto se realizó modificando la política de firewall que permite el tráfico hacia el puerto 2025. En este caso, se modificó la variable “ssh_action”, la cual tiene un valor por defecto de “allow”, para permitir el tráfico que ingrese del exterior hacia el puerto 2025. En ese sentido, se le asignó un valor de “deny” para que en lugar de permitir el tráfico por el puerto 2025, lo prohíba. De esta forma, aunque se intentó realizar la conexión por SSH hacia la VM de administración, no fue posible, ya que esta solicitud de conexión no llegó a la máquina virtual.

```
root@terraform:~/shared# terraform apply -var="ssh_action=deny"
module.security_group_open.openstack_networking_secgroup_v2.my_security_group: Refreshing state... [id=abfa8175-c6da-4024-b64b-a26b8a7db605]
data.openstack_networking_network_v2.ExtNet: Reading...
module.net2.openstack_networking_network_v2.net: Refreshing state... [id=13768f9d-5b8f-4150-99f7-c40c27ea21b3]
module.net1.openstack_networking_network_v2.net: Refreshing state... [id=a7c8acce-7de4-4ff3-9b14-8a235ce227d7]
module.database_server_image.openstack_images_image_v2.image: Refreshing state... [id=2ea5842b-8d9d-46bc-8754-65095e02985a]
module.web_server_image.openstack_images_image_v2.image: Refreshing state... [id=49ff48f0-049c-4094-97d7-07dc28a586ad]
module.net1.openstack_networking_subnet_v2.subnet: Refreshing state... [id=e2b0c4fc-ba71-4122-806e-f1c3f2bef9f4]
module.net2.openstack_networking_subnet_v2.subnet: Refreshing state... [id=a43f09c6-9f5a-42ac-bd97-2724bfff218a2]
module.r1.openstack_networking_port_v2.port: Refreshing state... [id=733e4c9b-2cd9-4d2f-af61-9f0edb0b1ad6]
data.openstack_networking_network_v2.ExtNet: Read complete after 0s [id=2378c1c8-f8aa-4901-a5e2-0d70c071b671]
module.r1.openstack_networking_router_v2.router: Refreshing state... [id=83c5a431-615f-4ffe-8f2b-05d6be148f73]
```

Figura 18: Despliegue con la política de Firewall que bloquea la conexión SSH

```
alex.segoviaancajima@1117: $ ssh -i /mnt/tmp/openstack_lab-antelope_4n_classic_ovs-v04/shared/keys/admin.pem root@10.0.10.187 -p 2025
ssh: connect to host 10.0.10.187 port 2025: Connection timed out
alex.segoviaancajima@1117: $
```

Figura 19: Intento de conexión por SSH hacia la VM de administración