

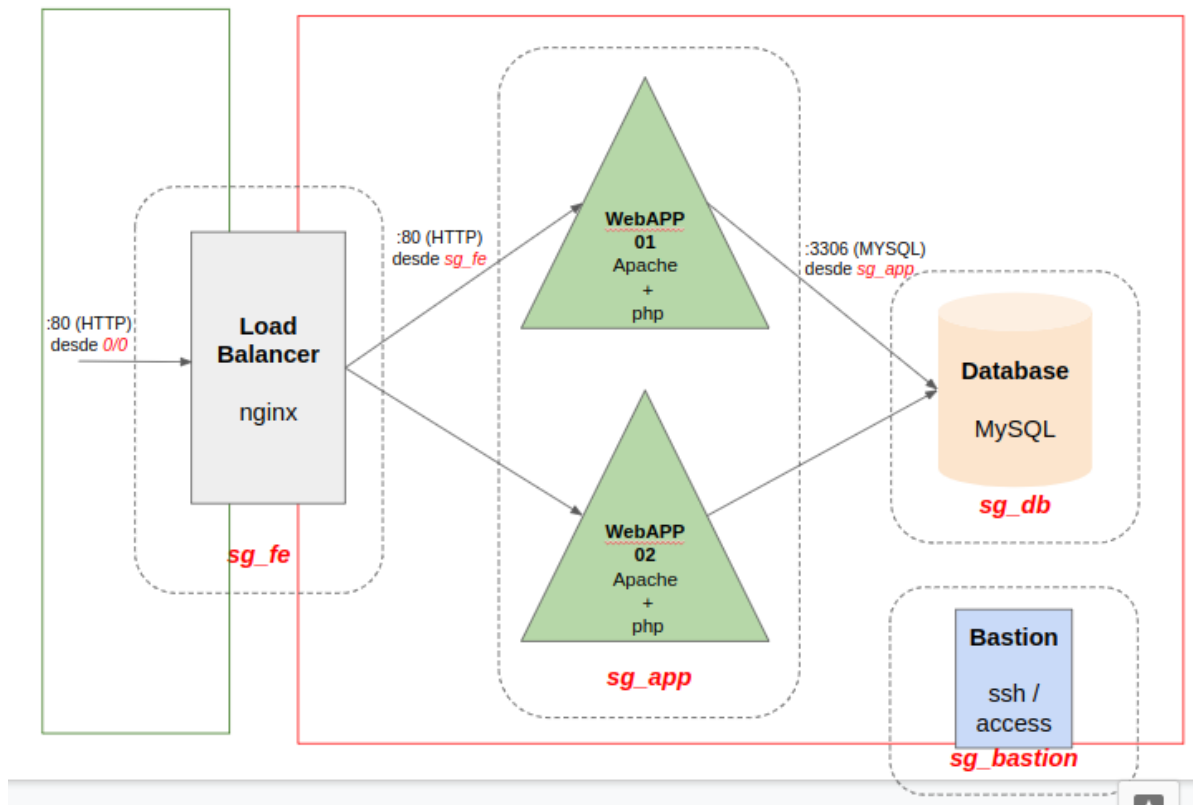
Teleinformática

Cloud Classic

Caso Práctico ++

Ricciardi Marcos

El objetivo a realizar, consiste en montar la siguiente infraestructura utilizando el servicio de Cloud provisto por la cátedra de teleinformática:



Para ello, una vez dentro de la plataforma de la Cloud, lo primero que se realizará será crear tantas instancias como sean necesarias, en nuestro caso serán cinco:

- Bastion
- Load Balancer
- WebApp 01
- WebApp 02
- DataBase

Cada instancia, como es de esperar, cumple con funciones diferentes, por lo que a la hora de crearlas se adecuarán las configuraciones según las necesidades en cada caso. Como es un caso de ejemplo bastante sencillo, las especificaciones para todas las instancias serán las mismas:

- 1 VCPU
- 2GB RAM
- 20GB Disco

Dicho conjunto de especificaciones se pueden encontrar a la hora de crear una nueva instancia, dentro de flavor como "m1.small"

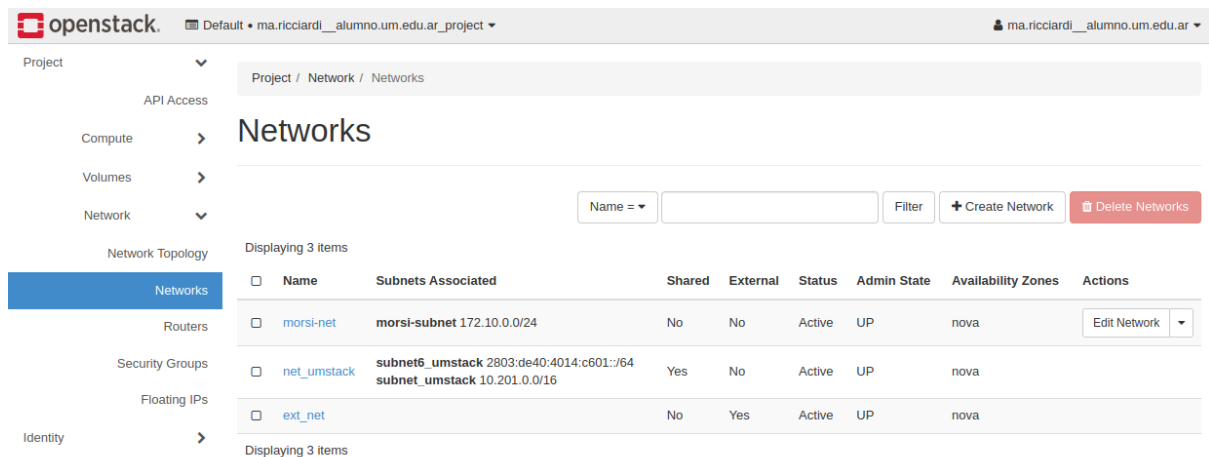
Los pasos para crear una nueva instancia son los mismos para todos los casos, salvo que en este caso, para ahorrar tiempo e instalaciones, se pueden elegir distribuciones de linux adecuadas a los requisitos de cada una de las instancias, por lo que será el único factor que diferirá entre cada una de las instancias. Todas tendrán como base Ubuntu 18.04.

Se explicará el proceso para crear una nueva instancia utilizando de ejemplo al Bastión, y de la misma manera se crearán las demás instancias, eligiendo la Imagen

Previo a la creación de las instancias es necesario crear una nueva red dentro de la infraestructura para separar las instancias propias de las de otros usuarios de la Cloud.

Creación de nueva red:

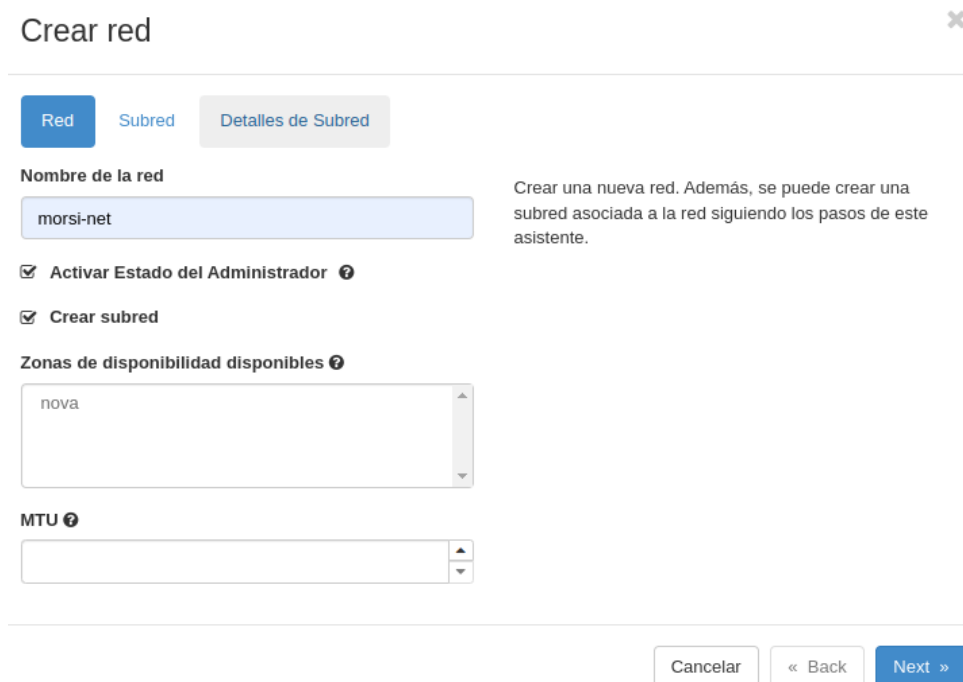
Dentro del dashboard, dentro de Networks, se encuentra la opción “Create Network”



Name	Subnets Associated	Shared	External	Status	Admin State	Availability Zones	Actions
morsi-net	morsi-subnet 172.10.0.0/24	No	No	Active	UP	nova	Edit Network
net_umstack	subnet6_umstack 2803:de40:4014:c601::/64 subnet_umstack 10.201.0.0/16	Yes	No	Active	UP	nova	
ext_net		No	Yes	Active	UP	nova	

Al hacer click en dicha opción se abrirá la siguiente ventana:

Aquí solo se definirá el nombre de la red



Crear red

Red Subred Detalles de Subred

Nombre de la red

morsi-net

☒ Activar Estado del Administrador

☒ Crear subred

Zonas de disponibilidad disponibles

nova

MTU

Cancelar « Back Next »

En la misma ventana, en la pestaña de subred, se deberá definir las direcciones de nuestra nueva red y el nombre de la misma:

Crear red

Red **Subred** Detalles de Subred

Nombre de subred

Direcciones de red ⓘ

Versión de IP

IP de la puerta de enlace ⓘ

☐ **Deshabilitar puerta de enlace**

Cancelar

« Back

Next »

Una vez creada la nueva red, es necesario crear un router para comunicar dicha red con el exterior

En el panel lateral de opciones, se encuentra la opción de “routers”:

openstack

Default • ma.ricciardi__alumno.um.edu.ar_project

ma.ricciardi__alumno.um.edu.ar

Project

API Access

Compute

Volumes

Network

Network Topology

Networks

Routers

Security Groups

Floating IPs

Identity

Project / Network / Routers

Routers

Router Name = Filter

+ Create Router

Delete Routers

Displaying 1 item

<input type="checkbox"/>	Name	Status	External Network	Admin State	Availability Zones	Actions
<input type="checkbox"/>	morsi-router	Active	ext_net	UP	nova	<div>Clear Gateway</div>

Displaying 1 item

Una vez dentro, se selecciona “create router”, lo que despliega la siguiente ventana:

Crear router

Nombre del router

morsi-router

Descripción:

Crea un enrutador con parámetros especificados.

☒ Activar Estado del Administrador ?

Red externa

ext_net

Zonas de disponibilidad disponibles ?

nova

Cancelar

Crear router

Solo resta elegir un nombre para el dispositivo y elegir la red externa a la cual se conectará dicho router.

Una vez creado el router, solo resta crear una nueva interfaz para unir dicho router con la red creada anteriormente. Para ello, dentro de la opción “network topology” se puede observar la topología de nuestra red, y a su vez al router agregado anteriormente se le puede añadir una nueva interfaz:

The screenshot shows the OpenStack Network Topology interface. The left sidebar contains a navigation menu with options: Project, API Access, Compute, Volumes, Network, Network Topology (selected), Networks, Routers, Security Groups, Floating IPs, and Identity. The main content area is titled 'Network Topology' and shows a diagram of the network topology. A router named 'morsi-router' is connected to a network labeled 'ext_net'. A modal window is open over the router, displaying its details:

- ID: ea7ae98e-d1be-4787-9d3d-c7682d4af7c5
- STATUS: Active
- Interfaces: 2985329f-be... (172.10.0.1 router_interface, Active), gateway4cd1... (None, None)
- Buttons: + Add Interface, View Router Details, Delete Interface, Delete Router

At the bottom of the modal, there are three buttons: Launch Instance, Create Network, and Create Router.

Al añadir una nueva interfaz se tiene la siguiente ventana:

Añadir interfaz

Subred *

morsi-net: 172.10.0.0/24 (morsi-subnet) ▼

Dirección IP (opcional) ?

Descripción:

Puede conectar una subred concreta al router.

Si no especifica aquí una dirección IP, se utilizará la dirección IP de la puerta de enlace de la subred seleccionada para la nueva interfaz del router. Si la dirección IP de la puerta de enlace ya se está usando, debe utilizar una dirección diferente del rango de la subred seleccionada.

Cancelar

Enviar

Solo hay que elegir con cual subred queremos conectar la nueva interfaz asociada al router, en este caso “morsi-net”.

Como requisito previo para poder gestionar de forma remota las instancias, es necesario crear un par de claves SSH, lo que en linux es muy sencillo.

Solo bastará con abrir una nueva terminal en nuestro equipo personal y ejecutar el comando “ssh-keygen” y listo. Esto generará automáticamente el par de claves, y si se desea se le puede colocar una palabra clave. El directorio será el definido por defecto: /home/user/.ssh/id_rsa.pub

Con estos requisitos previos ya cumplidos, ahora es posible crear correctamente las nuevas instancias y así asociarlas a la nueva red y cargar la clave SSH correspondiente.

Pasos para crear una nueva instancia:

En el panel lateral, dentro de “Compute”, disponemos de la opción “Instances”. Una vez dentro, hay que seleccionar “Launch Instance” para proceder a crear una nueva.

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
mricciardi-fe-01	srv-nginx-ubuntu1804	172.10.0.87, 192.168.3.72	m1.small	mr12	Active	nova	None	Running	1 week	Create Snapshot
mricciardi-wp-webapp02	srv-wordpress-ubuntu1804	172.10.0.150	m1.small	mr12	Active	nova	None	Running	1 week	Create Snapshot
mricciardi-wp-webapp01	srv-wordpress-ubuntu1804	172.10.0.178	m1.small	mr12	Active	nova	None	Running	1 week	Create Snapshot
mricciardi-wp-db	srv-mysql-ubuntu1804	172.10.0.76	m1.small	mr12	Active	nova	None	Running	1 week	Create Snapshot
mricciardi-bastion	ubuntu_minimal_1804	172.10.0.75, 192.168.3.68	m1.small	mr12	Active	nova	None	Running	1 week	Create Snapshot

En la ventana que se despliega, primero tenemos los detalles, donde se colocará el nombre de la instancia, en este caso “mricciardi-bastion”

Launch Instance

[Details](#)
[Source](#)
[Flavor *](#)
[Networks *](#)
[Network Ports](#)
[Security Groups](#)
[Key Pair](#)
[Configuration](#)
[Server Groups](#)
[Scheduler Hints](#)
[Metadata](#)

Please provide the initial hostname for the instance, the availability zone where it will be deployed, and the instance count. Increase the Count to create multiple instances with the same settings.

Project Name

Instance Name *

Description

Availability Zone

Count *

Total Instances (10 Max)

60%

5 Current Usage
1 Added
4 Remaining

✕ Cancel

< Back

Next >

Launch Instance

En “Source” se seleccionará la imagen a utilizar según las necesidades, en este caso se elegirá un Ubuntu 18.04 minimal, que alcanza para nuestro bastion.

Launch Instance

[Details](#)
[Source](#)
[Flavor *](#)
[Networks *](#)
[Network Ports](#)
[Security Groups](#)
[Key Pair](#)
[Configuration](#)
[Server Groups](#)
[Scheduler Hints](#)
[Metadata](#)

Instance source is the template used to create an instance. You can use an image, a snapshot of an instance (image snapshot), a volume or a volume snapshot (if enabled). You can also choose to use persistent storage by creating a new volume.

Select Boot Source

Create New Volume

Yes No

Allocated

Displaying 0 items

Name	Updated	Size	Type	Visibility
Select an item from Available items below				

Displaying 0 items

Available 12

Select one

Displaying 12 items

Name	Updated	Size	Type	Visibility
flatcar_prod	6/8/22 8:43 PM	343.12 MB	QCOW2	Public
srv-mysql-ubuntu1804	6/23/22 4:06 PM	1.66 GB	QCOW2	Public
srv-nginx-ubuntu1804	6/23/22 4:21 PM	1.43 GB	QCOW2	Public
srv-wordpress-ubuntu1804	6/23/22 4:21 PM	1.61 GB	QCOW2	Public
ubuntu_1804	6/6/22 1:44 PM	369.81 MB	QCOW2	Public
ubuntu_2004	6/6/22 1:56 PM	567.63 MB	QCOW2	Public
ubuntu_2204	6/6/22 1:57 PM	597.63 MB	QCOW2	Public
ubuntu_minimal_1804	6/6/22 1:59 PM	191.81 MB	QCOW2	Public

Además, en Select Boot Source, seleccionar Imagen y NO crear un nuevo volumen.

En la sección “Flavor” simplemente elegimos la opción “m1.small”

Launch Instance

Details

Source

Flavor *

Networks *

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

Flavors manage the sizing for the compute, memory and storage capacity of the instance.

Allocated

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
Select an item from Available items below						
▼ Available 7						
Click here for filters or full text search.						
Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
> m1.tiny	1	512 MB	1 GB	1 GB	0 GB	Yes
> m1.c1m1d20	1	1 GB	20 GB	20 GB	0 GB	Yes
> m1.small	1	2 GB	20 GB	20 GB	0 GB	Yes
> m1.c2m2d200	2	2 GB	200 GB	200 GB	0 GB	Yes
> m1.medium	2	4 GB	40 GB	40 GB	0 GB	Yes
> m1.large	4	8 GB	80 GB	80 GB	0 GB	Yes
> m1.xlarge	8	16 GB	160 GB	160 GB	0 GB	Yes

Cancel

< Back

Next >

Launch Instance

En Networks, asociamos la red creada anteriormente “morsi-net” a la nueva instancia:

Launch Instance

Details

Source

Flavor *

Networks

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

Networks provide the communication channels for instances in the cloud.

▼ Allocated 1

Select networks from those listed below.

Network	Subnets Associated	Shared	Admin State	Status
< 1 > morsi-net	morsi-subnet	No	Up	Active

▼ Available 1

Select at least one network

Click here for filters or full text search.

Network	Subnets Associated	Shared	Admin State	Status
> net_umstack	subnet6_umstack subnet_umstack	Yes	Up	Active

Cancel

< Back

Next >

Launch Instance

Launch Instance

Finalmente así quedan todas nuestras instancias creadas:

Instances

Displaying 5 items

Si bien ya disponemos de todas nuestras instancias, al formar parte de una red privada, es necesario asociar una IP flotante a nuestro Bastión para poder acceder a este de forma remota, así como debemos asociar una IP flotante al load balancer para poder acceder al sitio a través de internet.

Para esto en la opciones de cada instancia, se encuentra la opción de “Asociar IP flotante”.

Para el bastión:

Gestionar asociaciones de IP flotantes

Dirección IP *

192.168.3.68

Seleccione la dirección IP que quiere asociar con una determinada instancia o puerto.

Puerto a asociar *

mricciardi-bastion: 172.10.0.75

Cancelar

Asociar

Para el Load Balancer:

Gestionar asociaciones de IP flotantes

Dirección IP *

192.168.3.72

Seleccione la dirección IP que quiere asociar con una determinada instancia o puerto.

Puerto a asociar *

mricciardi-fe-01: 172.10.0.87

Cancelar

Asociar

Ahora lo que resta para poder proceder con la configuración de nuestras instancias, es definir los grupos de seguridad para cada una de ellas, según las acciones que sean necesarias en cada una.

Para configurar el grupo de seguridad se procede de la siguiente manera:

openstack. Default • ma.ricciardi_alumno.um.edu.ar_project

Project / Network / Security Groups

Security Groups

Filter [+ Create Security Group](#) [Delete Security Groups](#)

Displaying 5 items

<input type="checkbox"/>	Name	Security Group ID	Description	Actions
<input type="checkbox"/>	default	29054d73-446f-43cd-b8e7-95027457c37d	Default security group	Manage Rules
<input type="checkbox"/>	sg_app	8e092b54-130c-4be8-9957-0f58b0bcb71		Manage Rules
<input type="checkbox"/>	sg_bastion	7a518242-7a19-401d-9cd6-b91bed1f112b		Manage Rules
<input type="checkbox"/>	sg_db	854c2010-66d7-4b8e-8685-4e5cd3c5b72f		Manage Rules
<input type="checkbox"/>	sg_fe	cfb5add5-6f7c-4b8d-af35-b4e919462110		Manage Rules

Displaying 5 items

Dentro de Network, está la opción de “Security Groups”, donde se procederá a crear cada grupo de seguridad al hacer click en “Create Security Group”, lo cual despliega la siguiente ventana:

Create Security Group ✕

Name *

Description

Description:

Security groups are sets of IP filter rules that are applied to network interfaces of a VM. After the security group is created, you can add rules to the security group.

[Create Security Group](#)

En la ventana anterior, solo se crea el grupo de seguridad y se coloca el nombre deseado, luego corresponde agregar las reglas deseadas a cada grupo.

Bastión:

El bastión será nuestro único punto de acceso mediante internet a la configuración de esta y las demás instancias, por lo que el grupo de seguridad asociado a esta instancia debe permitir en particular el tráfico entrante desde cualquier IP a través del puerto 22 (SSH). Por lo que la configuración será la siguiente:

Manage Security Group Rules: sg_bastion (7a518242-7a19-401d-9cd6-b91bed1f112b)

[+ Add Rule](#) [Delete Rules](#)

Displaying 4 items

<input type="checkbox"/>	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Description	Actions
<input type="checkbox"/>	Egress	IPv4	Any	Any	0.0.0.0/0	-	-	Delete Rule
<input type="checkbox"/>	Egress	IPv6	Any	Any	:::0	-	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	ICMP	Any	0.0.0.0/0	-	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	22 (SSH)	0.0.0.0/0	-	-	Delete Rule

Displaying 4 items

Load Balancer:

Para el caso del balanceador de carga (frontend), es necesario poder acceder al mismo desde cualquier dirección IP a través del puerto 80 (HTTP) para que el sitio sea accesible a través de internet, y además se debe permitir el acceso al puerto 22 (SSH), pero como el requisito es que las demás instancias sean accesibles mediante el bastión, el puerto 22 de esta instancia se asocia con el grupo de seguridad correspondiente al bastión, de esta manera se evita colocar direcciones IP y no es necesario preocuparse por cambios en las direcciones o por si se agregan nuevas instancias.

Manage Security Group Rules: sg_fe (cfb5add5-6f7c-4b8d-af35-b4e919462110)

+ Add Rule

Delete Rules

Displaying 4 items

<input type="checkbox"/>	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Description	Actions
<input type="checkbox"/>	Egress	IPv4	Any	Any	0.0.0.0/0	-	-	Delete Rule
<input type="checkbox"/>	Egress	IPv6	Any	Any	:::0	-	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	22 (SSH)	-	sg_bastion	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	80 (HTTP)	0.0.0.0/0	-	-	Delete Rule

Displaying 4 items

Web Apps:

Las aplicaciones web serán accesibles desde internet a través del load balancer, por lo que de la misma manera que el caso anterior, las instancias que corran las apps, estarán asociadas directamente al grupo de seguridad del frontend, de manera que no es necesario especificar ninguna dirección IP y a medida que se agreguen mas apps, con asociarlas al grupo de seguridad correspondiente será suficiente. Además para la configuración por SSH, como ya se explicó anteriormente, se asociará el puerto 22 al grupo de seguridad del bastión.

Manage Security Group Rules: sg_app (8e092b54-130c-4be8-9957-0f58b0bcfb71)

+ Add Rule

Delete Rules

Displaying 4 items

<input type="checkbox"/>	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Description	Actions
<input type="checkbox"/>	Egress	IPv4	Any	Any	0.0.0.0/0	-	-	Delete Rule
<input type="checkbox"/>	Egress	IPv6	Any	Any	:::0	-	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	22 (SSH)	-	sg_bastion	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	80 (HTTP)	-	sg_fe	-	Delete Rule

Displaying 4 items

DataBase:

La base de datos será accedida por las apps, de manera que directamente el puerto 3306 (MYSQL) será asociado al grupo de seguridad de las WebApps, de forma que no sea necesario especificar las IP de cada instancia, solo con que la instancia forme parte del grupo de seguridad de las WebApps, ya podrán acceder a la base de datos por el puerto correspondiente. Además para la configuración por SSH, como ya se explicó anteriormente, se asociará el puerto 22 al grupo de seguridad del bastión.

Manage Security Group Rules: sg_db (854c2010-66d7-4b8e-8685-4e5cd3c5b72f)

[+ Add Rule](#) [Delete Rules](#)

Displaying 4 items

<input type="checkbox"/>	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Description	Actions
<input type="checkbox"/>	Egress	IPv4	Any	Any	0.0.0.0/0	-	-	Delete Rule
<input type="checkbox"/>	Egress	IPv6	Any	Any	:::/0	-	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	22 (SSH)	-	sg_bastion	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	3306 (MYSQL)	-	sg_app	-	Delete Rule

Displaying 4 items

En todos los casos anteriores, para administrar cada grupo de seguridad, basta con entrar a “Manage Rules” de cada grupo de seguridad deseado y con pulsar “Add rule” se podrá agregar la regla deseada.

Una vez creados los grupos de seguridad, volvemos a instancias y asociamos a cada una el grupo correspondiente:

[Disassociate Floating IP](#)
[Attach Interface](#)
[Detach Interface](#)
[Edit Instance](#)
[Attach Volume](#)
[Detach Volume](#)
[Update Metadata](#)
[Edit Security Groups](#)

Edit Instance

[Information](#) [Security Groups](#)

Add and remove security groups to this instance from the list of available security groups.

Warning: If you change security groups here, the change will be applied to all interfaces of the instance. If you have multiple interfaces on this instance and apply different security groups per port, use “Edit Port Security Groups” action instead.

All Security Groups

default	+
sg_bastion	+
sg_db	+
sg_app	+

Instance Security Groups

sg_fe	-
-------	---

[Cancel](#) [Save](#)

Hecho esto, ya es posible proceder con la configuración de cada una de las instancias

El primer paso será acceder al bastión mediante SSH extendiendo nuestra clave para poder acceder a las demás instancias, de la siguiente manera:

```
marcos@mr-arch:~$ ssh -A ubuntu@192.168.3.68
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-1116-kvm x86_64)

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

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

12 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

New release '20.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Tue Jul 19 04:31:37 2022 from 192.168.3.252
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@mr Ricciardi-bastion:~$
```

Una vez dentro del bastión empezamos configurando la base de datos, previo acceso a la misma por SSH:

```
ubuntu@mr Ricciardi-bastion:~$ ssh ubuntu@172.10.0.76
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-180-generic x86_64)

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

System information as of Tue Jul 19 04:34:01 UTC 2022

System load:  0.05          Processes:            84
Usage of /:   11.0% of 19.21GB Users logged in:          0
Memory usage: 24%          IP address for ens3: 172.10.0.76
Swap usage:   0%

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

11 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

*** System restart required ***
Last login: Thu Jul 14 03:53:02 2022 from 172.10.0.75
ubuntu@mr Ricciardi-wp-db:~$
```

En caso de no tener MYSQL instalado, se procede a instalarlo junto a un editor de texto en caso de ser necesario:

```
sudo apt update && sudo apt-get install -y  
mysql-server vim-tiny
```

Poner Base de datos disponible en la RED:

```
sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf  
#bind-address          = 127.0.0.1  
bind-address           = 0.0.0.0  
sudo systemctl restart mysql
```

```
ubuntu@mr Ricciardi-wp-db:~$ sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf  
ubuntu@mr Ricciardi-wp-db:~$ sudo systemctl restart mysql  
ubuntu@mr Ricciardi-wp-db:~$ sudo mysql  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 2  
Server version: 5.7.38-0ubuntu0.18.04.1 (Ubuntu)  
  
Copyright (c) 2000, 2022, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> CREATE DATABASE wordpress;  
Query OK, 1 row affected (0.00 sec)  
  
mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER  
-> ON wordpress.*  
-> TO wordpress@'%'  
-> IDENTIFIED BY 'telewordpress';  
Query OK, 0 rows affected, 1 warning (0.00 sec)  
  
mysql> █
```

Luego procedemos a configurar cada una de las WebApps, dos en este caso:

En caso de no tenerlo instalado se instala wordpress, php, apache y demás junto a un editor de texto:

```
sudo apt-get update && sudo apt-get install -y wordpress php libapache2-mod-php php-mysql  
vim-tiny
```

Luego se crea el archivo de configuración para cada web app:

```
sudo vi /etc/apache2/sites-available/wordpress.conf
```

Alias /blog /usr/share/wordpress

```
<Directory /usr/share/wordpress>
    Options FollowSymLinks
    AllowOverride Limit Options FileInfo
    DirectoryIndex index.php
    Order allow,deny
    Allow from all
</Directory>
<Directory /usr/share/wordpress/wp-content>
    Options FollowSymLinks
    Order allow,deny
    Allow from all
</Directory>
```

```
#sudo vi /etc/wordpress/config-<IP_INSTANCIA_WEBAPP>.php
```

```
sudo vi /etc/wordpress/config-10.201.0.XX.php
```

```
<?php
define('DB_NAME', 'wordpress');
define('DB_USER', 'wordpress');
define('DB_PASSWORD', 'telewordpress'); #password DB
define('DB_HOST', '10.201.Y.Z'); #Reemplazar por la
IP_INSTANCIA_DB
define('DB_COLLATE', 'utf8_general_ci');
define('WP_CONTENT_DIR',
'/usr/share/wordpress/wp-content');
?>
```

Activar Sitio wordpress y módulo de rewrite

```
sudo a2ensite wordpress && sudo a2enmod rewrite
```

Reiniciar apache

```
sudo service apache2 restart
```


WebApp01:

```
ubuntu@mr Ricciardi-bastion:~$ ssh ubuntu@172.10.0.178
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-180-generic x86_64)

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

System information as of Tue Jul 19 04:42:17 UTC 2022

System load:  0.0               Processes:           90
Usage of /:   10.0% of 19.21GB   Users logged in:    0
Memory usage: 16%              IP address for ens3: 172.10.0.178
Swap usage:   0%

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

https://ubuntu.com/blog/microk8s-memory-optimisation

11 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

*** System restart required ***
Last login: Tue Jul 12 16:02:17 2022 from 172.10.0.75
ubuntu@mr Ricciardi-wp-webapp01:~$ cat /etc/apache2/sites-available/wordpress.conf
Alias /blog /usr/share/wordpress
<Directory /usr/share/wordpress>
    Options FollowSymLinks
    AllowOverride Limit Options FileInfo
    DirectoryIndex index.php
    Order allow,deny
    Allow from all
</Directory>
<Directory /usr/share/wordpress/wp-content>
    Options FollowSymLinks
    Order allow,deny
    Allow from all
</Directory>
ubuntu@mr Ricciardi-wp-webapp01:~$ cat /etc/wordpress/config-172.10.0.178.php
<?php
define('DB_NAME', 'wordpress');
define('DB_USER', 'wordpress');
define('DB_PASSWORD', 'telewordpress'); #password DB
define('DB_HOST', '172.10.0.76'); #Reemplazar por la IP_INSTANCIA_DB
define('DB_COLLATE', 'utf8_general_ci');
define('WP_CONTENT_DIR', '/usr/share/wordpress/wp-content');
?>

ubuntu@mr Ricciardi-wp-webapp01:~$
```

WebApp02:

```
ubuntu@mr Ricciardi-bastion:~$ ssh ubuntu@172.10.0.150
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-180-generic x86_64)

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

System information as of Tue Jul 19 04:48:55 UTC 2022

System load:  0.0               Processes:            90
Usage of /:   9.0% of 19.21GB   Users logged in:     0
Memory usage: 13%              IP address for ens3: 172.10.0.150
Swap usage:   0%

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

https://ubuntu.com/blog/microk8s-memory-optimisation

11 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

*** System restart required ***
Last login: Thu Jul 14 03:50:40 2022 from 172.10.0.75
ubuntu@mr Ricciardi-wp-webapp02:~$ cat /etc/apache2/sites-available/wordpress.conf
Alias /blog /usr/share/wordpress
<Directory /usr/share/wordpress>
    Options FollowSymLinks
    AllowOverride Limit Options FileInfo
    DirectoryIndex index.php
    Order allow,deny
    Allow from all
</Directory>
<Directory /usr/share/wordpress/wp-content>
    Options FollowSymLinks
    Order allow,deny
    Allow from all
</Directory>

ubuntu@mr Ricciardi-wp-webapp02:~$ cat /etc/wordpress/config-172.10.0.150.php
<?php
define('DB_NAME', 'wordpress');
define('DB_USER', 'wordpress');
define('DB_PASSWORD', 'telewordpress'); #password DB
define('DB_HOST', '172.10.0.76'); #Reemplazar por la IP_INSTANCIA_DB
define('DB_COLLATE', 'utf8_general_ci');
define('WP_CONTENT_DIR', '/usr/share/wordpress/wp-content');
?>
```

Load Balancer:

Finalmente resta configurar el balanceador de carga.

En caso de no tener instalado Nginx, y un editor de texto:

```
sudo apt-get update && sudo apt-get install nginx vim-tiny curl
```

Primero eliminar el sitio por defecto de Nginx:

```
sudo rm /etc/nginx/sites-enabled/default
```

Configurar Nginx como proxy reverso con múltiples WebApps para este caso:

```
ubuntu@mr Ricciardi-bastion:~$ ssh ubuntu@172.10.0.87
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-180-generic x86_64)

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

System information as of Tue Jul 19 04:56:55 UTC 2022

System load:  0.3               Processes:            87
Usage of /:   9.2% of 19.21GB   Users logged in:     0
Memory usage: 15%              IP address for ens3: 172.10.0.87
Swap usage:   0%

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

https://ubuntu.com/blog/microk8s-memory-optimisation

11 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

New release '20.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Tue Jul 19 04:56:50 2022 from 172.10.0.75
ubuntu@mr Ricciardi-fe-01:~$ cat /etc/nginx/conf.d/lb.conf
access_log /var/log/nginx/access.log;
upstream 192.168.3.72 {
server 172.10.0.150;
server 172.10.0.178;
}

# This server accepts all traffic to port 80 and passes it to the upstream.
# Notice that the upstream name and the proxy_pass need to match.

server {
listen 80;
location / {
proxy_pass http://192.168.3.72;
}
}
ubuntu@mr Ricciardi-fe-01:~$
```

sudo vi /etc/nginx/conf.d/lb.conf

```
upstream 192.168.3.72 {  
    server 172.10.0.150;  
    server 172.10.0.178;  
}  
  
# This server accepts all traffic to port 80 and passes it  
to the upstream.  
# Notice that the upstream name and the proxy_pass need to  
match.  
  
server {  
    listen 80;  
    location / {  
        proxy_pass http://192.168.3.72;  
    }  
}
```

Es importante remarcar que la dirección de la IP flotante asociada al Load Balancer se redireccione a las diferentes webapps, ya que si se utiliza otro nombre como variable, puede generar conflictos a la hora de intentar acceder al sitio web

Una vez cargada la configuración, se reinicia el servicio:

sudo service nginx restart

Como pasos finales, resta crear nuevamente en las webapps los archivos de configuración con la dirección de la IP flotante del Load Balancer para poder acceder al sitio a través de la IP flotante y que cargue correctamente

```
ubuntu@mr Ricciardi-wp-webapp01:~$ cd /etc/wordpress/  
ubuntu@mr Ricciardi-wp-webapp01:/etc/wordpress$ ls  
config-172.10.0.178.php  config-192.168.3.72.php  config-backend.php  htaccess  
ubuntu@mr Ricciardi-wp-webapp01:/etc/wordpress$ cat config-192.168.3.72.php  
<?php  
define('DB_NAME', 'wordpress');  
define('DB_USER', 'wordpress');  
define('DB_PASSWORD', 'telewordpress'); #password DB  
define('DB_HOST', '172.10.0.76'); #Reemplazar por la IP_INSTANCIA_DB  
define('DB_COLLATE', 'utf8_general_ci');  
define('WP_CONTENT_DIR', '/usr/share/wordpress/wp-content');  
?>  
ubuntu@mr Ricciardi-wp-webapp01:/etc/wordpress$
```

```
ubuntu@mr Ricciardi-wp-webapp02:~$ cd /etc/wordpress/  
ubuntu@mr Ricciardi-wp-webapp02:/etc/wordpress$ ls  
config-172.10.0.150.php  config-192.168.3.72.php  config-backend.php  htaccess  
ubuntu@mr Ricciardi-wp-webapp02:/etc/wordpress$ cat config-192.168.3.72.php  
<?php  
define('DB_NAME', 'wordpress');  
define('DB_USER', 'wordpress');  
define('DB_PASSWORD', 'telewordpress'); #password DB  
define('DB_HOST', '172.10.0.76'); #Reemplazar por la IP_INSTANCIA_DB  
define('DB_COLLATE', 'utf8_general_ci');  
define('WP_CONTENT_DIR', '/usr/share/wordpress/wp-content');  
?>  
ubuntu@mr Ricciardi-wp-webapp02:/etc/wordpress$
```

Una vez agregados los archivos de configuración a cada webapp asociados a la IP flotante del load balancer, lo único que resta es eliminar la base de datos creada al principio y recrearla nuevamente, hecho esto quedaría por concluida la configuración de nuestra infraestructura.

Al acceder a la dirección 192.168.3.72/blog a través del navegador y realizar la configuración correspondiente del sitio el resultado obtenido es el siguiente:

