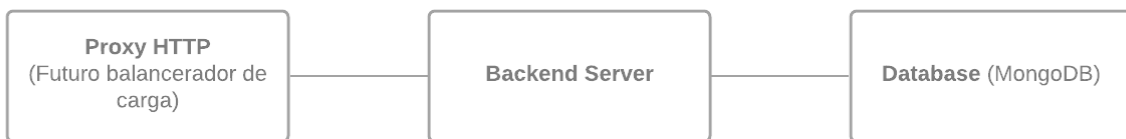


Laboratorio Virtual en IaaS

Introducción	1
Creación de las máquina virtuales	1
Configuración interna de las máquinas	4
Configuración de Interfaces de Red	5
Esquema de la red	6
Interfaces con sus IP	7

Introducción

En esta práctica vamos a crear una serie de máquinas virtuales en el servicio IaaS que nos ofrece la ULL. Dichas máquinas tendrán una función única y concreta, las cuales se pueden ver en el siguiente esquema:



Cada cuadro representa una máquina virtual, las tres estarán conectadas entre sí mediante la red, para así poder comunicarse y ofrecer todos los servicios.

Para que el IaaS detecte que las máquinas pertenecen a la asignatura es importante nombrarlas con "SyTW-" al principio. Como se verá a continuación, las máquinas no contienen ese nombre, pero han sido renombradas para su correcto funcionamiento.

Creación de las máquina virtuales

Máquina Proxy

Para crear esta máquina nos dirigimos a la página del IaaS. Entramos al apartado [Crear Máquina Virtual](#), ahí aplicaremos un nombre, seleccionamos la plantilla, la cantidad de memoria, CPU, etc.

Quedaría de la siguiente manera:

Configuración básica Redes Almacenamiento Revisar

1 2 3 4

* Nombre: STW_MV1

Descripción: PROXY HTTP (FUTURO BALANCEADOR)

* Clúster: Cluster-Rojo (KVMRojo)

* Fuente de aprovisionamiento: Plantilla

* Plantilla: debian-9

* Sistema operativo: Debian 9+

* Memoria (MiB): 4096

* Total de CPUs virtuales: 4

* Optimizado para: Servidor

☐ Iniciar máquina virtual en la creación

☐ Habilitar Cloud-init/Sysprep

Una vez creada la máquina virtual nos dirigimos a los ajustes de la misma. En el apartado ajustes de red vamos a crear otra interfaz llamada “Proxy-Backend” este nombre se debe a la conexión que va a realizar, del proxy al backend. Esta nos va a servir como segunda red totalmente cerrada (sin direccionamiento definido) que utilizaremos para nuestras redes internas, esta tendrá que ser del tipo (DOCINT1/DOCINT1) para que funcione correctamente.

Interfaces de red 2

nic1 (DOC1/DOC1)

PROXY-BACKEND (DOCINT1/DOCINT1)

Máquina Backend

En este paso efectuaremos los mismos que en el paso 2, está será la máquina para el backend, la diferencia es que crearemos dos interfaces de red totalmente privadas, una para conectarla con la máquina Proxy y la otra con la máquina BD.



Máquina BD

Esta tercera máquina se encarga de manejar la base de datos MongoDB, a la cual se conectará el backend para pedir los datos que necesite. Crearemos la máquina con los mismo ajustes que las anteriores, lo único que cambia son las interfaces de red, en este caso solo tendremos una de tipo DOCINT2.



Configuración interna de las máquinas

Una vez creadas las máquinas procedemos a realizar su configuración, uno de ellos es actualizar las librerías de linux:

```
sudo apt update
sudo apt upgrade
```

Una vez actualizada las librerías procedemos a hacer una serie de retoques:

1. Añadir una clave privada a las tres máquinas para poder conectarnos sin necesidad de introducir la contraseña (Se realiza en la máquina local el CMD si fuera Windows 10):

```
ssh-keygen
scp C:\Users\${ENV:USERNAME}\.ssh\yid_rsa.pub usuario@ipaddress:~
```

2. Actualizar el prompt de las máquinas virtuales:

```
sudo apt install zsh && sudo apt install curl && sudo apt install git

sh -c "$(curl -fsSL
https://raw.githubusercontent.com/robbyrussell/oh-my-zsh/master/tools/install.sh)"

git clone https://github.com/romkatv/powerlevel10k.git
${ZSH_CUSTOM}/themes/powerlevel10k

git clone https://github.com/zsh-users/zsh-syntax-highlighting.git
${ZSH_CUSTOM:~/.oh-my-zsh/custom}/plugins/zsh-syntax-highlighting

git clone https://github.com/zsh-users/zsh-autosuggestions
${ZSH_CUSTOM:~/.oh-my-zsh/custom}/plugins/zsh-autosuggestions
```

Editar el fichero .zshrc

Cambiamos donde pone: "robbyrussell" por "powerlevel10k/powerlevel10k"

Cambiamos la línea de plugins por esta:

plugins=(git zsh-syntax-highlighting zsh-autosuggestions)

```
source .zshrc
```

Completa el menú de configuración de powerlevel10k

Modificar el .bashrc añadiendo en la línea 2 lo siguiente:

```
exec zsh
```

Las terminales de las máquinas quedarían así:



Podemos ver como ha cambiado el prompt, a la derecha podemos ver un identificador de la maquina, asi podemos ver en todo momento en la máquina que estamos.

Configuración de Interfaces de Red

Para poder conectarnos entre todas las máquinas nos falta configurar las interfaces. Se hace mediante un fichero en **/etc/network/interfaces**:

Para la primera máquina tenemos que introducir lo siguiente:

```
# The secondary network interface PROXY -> Backend
auto ens3
iface ens3 inet static
address 172.16.12.1
netmask 255.255.255.0
```

En la segunda máquina:

```
# The secondary network interface Backend -> DB
auto ens3
iface ens3 inet static
address 172.16.13.2
netmask 255.255.255.0

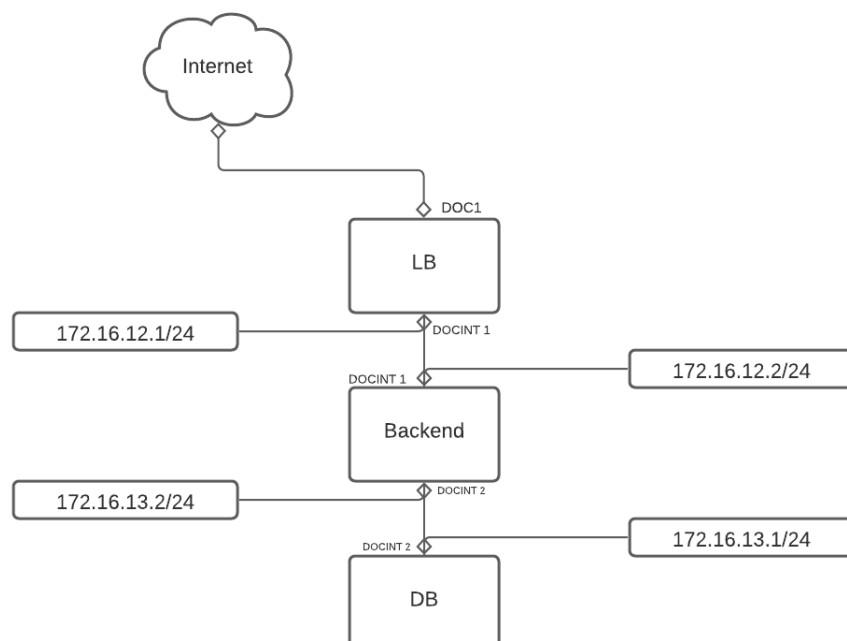
# The third network interface Backend -> Proxy
auto ens8
iface ens8 inet static
address 172.16.12.2
netmask 255.255.255.0
```

En la tercera máquina:

```
# The secondary network interface DB -> Backend
auto ens3
iface ens3 inet static
address 172.16.13.1
netmask 255.255.255.0
```

Esquema de la red

En el siguiente esquema podemos ver porque hemos usado esas direcciones IP en especial. En mi caso, elegí el rango de direcciones **172.16.12.X/24** a **172.16.13.X/24** para trabajar en este proyecto.

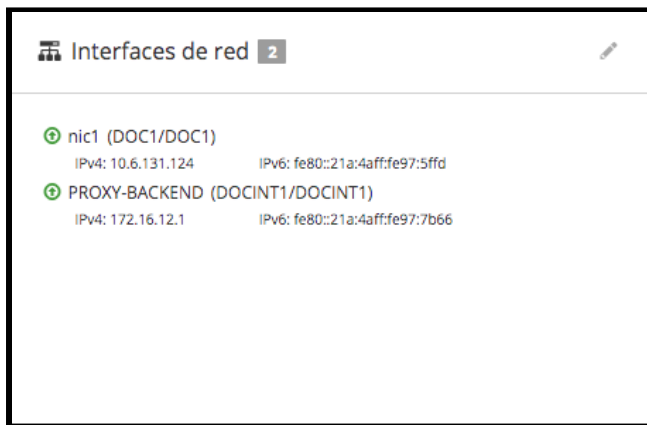


Las interfaz de red de la maquina LB junto con la de la máquina de Backend son de tipo DOCINT1, es decir, se conectan a una misma red, a la 172.16.12.0, a cada una de estas dos interfaces hay que añadirle una dirección ip de esta red, las que se ven en el esquema.

Lo mismo ocurre para las interfaces de Backend y DB, hay que hacer lo mismo, en este caso con la red 172.16.13.0, de tipo DOCINT2.

Interfaces con sus IP

A continuación podemos ver cuales son las direcciones IP asignada a cada interfaz, según muestra el IaaS:



Proxy



Backend



BD