

# ***MICROSOFT AZURE***



**Realizado por:**

**Baeza Álvarez, Jesús  
Ruano Narváez, Raúl**

# ÍNDICE

1- INTRODUCCIÓN.

2- OBJETIVO DEL TRABAJO.

3- MICROSOFT AZURE, ¿QUÉ ES?.

4- MICROSOFT AZURE, ¿QUÉ NOS OFRECE?

5- ¿CÓMO FUNCIONA?

6- PROCESO DE INSTALACIÓN

7- CONFIGURACIÓN DEL BALANCEADOR DE CARGA USANDO AZURE

7.1- CREACIÓN DEL BALANCEADOR DE CARGA

7.2- CREACIÓN DE UNA RED VIRTUAL

7.3- CREACIÓN DE MÁQUINAS VIRTUALES

7.4- CONFIGURACIÓN EXTRA TRAS LA CREACIÓN DE LAS MÁQUINAS

7.5- CREACIÓN DE REGLAS NGS

7.6- CREACIÓN DE DIRECCIONES DE BACK-END

7.7- CREACIÓN DE UN SONDEO DE ESTADO

7.8- CREACIÓN DE UNA REGLA DE LOAD BALANCER

8- PRUEBA DEL BALANCEADOR

9- CONCLUSIONES Y OPINIÓN

10- BIBLIOGRAFÍA

# 1- INTRODUCCIÓN

La realización de este trabajo forma parte de la evaluación teórica de la asignatura SWAP. De entre todos los temas propuestos, hemos elegido el de Microsoft Azure ya que nos ha parecido interesante y nos llamó la atención el sistema de virtualización que emplea dicho sistema de Microsoft.

El desarrollo del trabajo seguirá un orden cronológico, es decir, se explicará el objetivo del trabajo, explicación de Microsoft Azure, proceso de instalación, configuración, etc...

## 2- OBJETIVO DEL TRABAJO

Se utilizará Microsoft Azure para la configuración de un balanceador de carga. A dicho balanceador estarán asociadas 2 máquinas virtuales (creadas bajo la herramienta de virtualización que ofrece Microsoft Azure) que correrán con Ubuntu Server.

Se pretende que se establezca un reparto de carga entre las 2 máquinas mediante la configuración y el uso del balanceador.

## 3- MICROSOFT AZURE, ¿QUÉ ES?

Azure es un conjunto integral de servicios en la nube que los desarrolladores y los profesionales de las tecnologías de la información (TI) utilizan para crear, implementar y administrar aplicaciones a través de una red global de centros de datos. Herramientas integradas, DevOps y un marketplace ayudan a crear de manera eficaz cualquier cosa, desde aplicaciones móviles sencillas hasta soluciones orientadas a Internet.

En cuanto a las características principales de esta plataforma se encuentran:

- **Proceso**: el servicio de proceso de Windows Azure ejecuta aplicaciones basadas en Windows Server. Estas aplicaciones se pueden crear mediante .NET Framework en lenguajes como C# y Visual Basic, o implementar sin .NET en C++, Java y otros lenguajes.
- **Almacenamiento**: objetos binarios grandes (blobs) proporcionan colas para la comunicación entre los componentes de las aplicaciones de Windows Azure y ofrece un tipo de tablas con un lenguaje de consulta simple.
- **Servicios de infraestructura**: posibilidad de desplegar de una forma sencilla máquinas virtuales con Windows Server o con distribuciones de Linux.

- **Controlador de tejido**: Windows Azure se ejecuta en un gran número de máquinas. El trabajo del controlador de tejido es combinar las máquinas en un solo centro de datos de Windows Azure formando un conjunto armónico. Los servicios de proceso y almacenamiento de Windows Azure se implementan encima de toda esta eficacia de procesamiento.
- **Red de entrega de contenido (CDN)**: el almacenamiento en caché de los datos a los que se accede frecuentemente cerca de sus usuarios agiliza el acceso a esos datos.
- **Connect**: organizaciones interactúan con aplicaciones en la nube como si estuvieran dentro del propio firewall de la organización.
- **Administración de identidad y acceso**: La solución Active Directory permite gestionar de forma centralizada y sencilla el control de acceso y la identidad. Esta solución es perfecta para la administración de cuentas y la sincronización con directorios locales.

## 4- MICROSOFT AZURE, ¿QUE NOS OFRECE?

- **IaaS** (Infraestructura como servicio): servicios orientados a que el usuario tenga el control total de la infraestructura virtual. Aquí incluimos todo lo relacionado con servidores (máquinas virtuales) donde escoger sistema operativo (Windows Server, Linux, Oracle, Open Logic, etc.), número de núcleos de procesamiento, tamaño de la RAM o discos virtuales. Azure cuenta con una larga lista de máquinas virtuales ya creadas en la galería, tanto servidores de Sharepoint, de desarrollo con Visual Studio y la mayoría de server Enterprise de la distribuciones de Linux en Ubuntu, CentOS u Oracle.
- **PaaS** (Plataforma como servicio): en este nivel nos encontramos una plataforma ya creada que Azure gestiona por nosotros, escalando y desplegando según las necesidades de nuestras aplicaciones. Así, ojo, nos encontramos con sorpresa como que podemos instalar un Wordpress en Azure o cualquier CMS (Drupal, Joomla, etc..) y frameworks (Django, CakePHP, etc..). Choca con la idea esperada de un ecosistema cerrado de Microsoft basado en .Net, ASP o SQL Server. Alejado de la realidad
- **SaaS** (Software como Servicio): servicios donde la infraestructura y la plataforma están ocultas bajo una capa de abstracción. El cliente consume directamente las aplicaciones en formato servicios. Aquí nos encontramos

una tremenda base de datos en la nube SQL Azure, servicios de Big Data como Hadoop o suyo propio como HDInsight integrado con LINQ y Hive, Biztalk como servicio de comunicaciones, servicios de Directorio Activo o los Service Bus para difundir material audiovisual.

Podemos ver algo más extenso que esto que hemos explicado anteriormente con una simple foto, donde veremos las funcionalidades que nos proporciona Microsoft Azure.

## FUNCIONALIDADES



### MÁQUINAS VIRTUALES

Aplicaciones en la nube y máquinas virtuales Windows y Linux de gran capacidad.



### APLICACIONES WEB Y MÓVIL

Plataforma de creación de apps para cualquier dispositivo.



### ALMACENAMIENTO EN LA NUBE

Gran capacidad de almacenaje de archivos, datos y discos.



### STORSIMPLE

Almacenamiento en la nube híbrido para empresas.



### BASE DE DATOS

Base de datos SQL relacional administrada y NoSQL administrada.



### APRENDIZAJE AUTOMÁTICO

Potentes análisis predictivos basados en la nube.



### ANÁLISIS

Análisis distribuido que facilita los macrodatos y procesamiento de datos en tiempo real



### INTERNET DE LAS COSAS (IOT)

Conecta, supervisa y controla millones de activos de IoT.



### RED VIRTUAL

Aprovisiona redes privadas, equilibra cargas y hospeda dominios DNS en Azure.



### SERVICIOS MULTIMEDIA Y CDN

Codifica, almacena y transmite contenidos de audio y vídeo.



### BACKUP

Copias de seguridad en los servidores de la nube.



### ADMINISTRACIÓN

Autenticación de accesos, sincronización de directorios, gestión de usuarios e identidad.



### VISUAL STUDIO

Entorno de desarrollo de aplicaciones móviles y web en Azure.



### FACTURACIÓN

Detalles de consumo de tus suscripciones a recursos Microsoft Azure.

## 5-¿CÓMO FUNCIONA?

En el portal Microsoft Azure existen diferentes servicios de infraestructura y de plataforma para que puedas “montar” los servicios que necesites de manera sencilla, con unos cuantos clics.

En tu portal dispones de un botón [Crear un recurso +] y a partir de ahí, sólo tienes que elegir la región de los datacenter donde estará tu servicio y, a continuación, el tipo de servicio con sus características. Por ejemplo, en el caso de querer crear una máquina virtual podrías seleccionar el tipo de máquina como Windows Server 2016 Datacenter, a continuación, rellenar todos sus características (espacio de disco, RAM...). En este proceso de creación, se define un nombre de usuario y contraseña específica para poder iniciar sesión en esta máquina virtual.

Entre los servicios, dispones de infraestructuras (IaaS: almacenamiento, redes, máquinas virtuales...) y plataformas (PaaS: bases de datos de alta disponibilidad SQL, CMS para desarrollo de web, backend para aplicaciones móviles...). Son compatibles con todo tipo de tecnología: bases de datos Oracle, Linux, php, iOS, MySQL, Android, php...

Estos servicios están garantizados con una disponibilidad del 99.99%, y en caso de fallo en disponibilidad superior, Microsoft se compromete a indemnizar por los daños. Además, cuenta con todas las certificaciones en materia seguridad y protección de datos.

Usando este servicio de cloud computing tenemos la posibilidad de si necesitamos mejorar nuestros sistemas tan solo deberíamos acceder al sistema para mejorar las máquinas que necesitemos añadiendo más o mejorando las que ya tenemos, con esto nos ahorramos mucho tiempo y espacio, ya que tampoco tenemos que tener los equipos físicos a nuestra disposición. Destacar que todo este sistema de cloud computing es de pago y para hacer uso de máquinas virtuales, bases de datos y demás hay que pagar una cuota que ellos especifican en su web.

## 6- PROCESO DE INSTALACIÓN

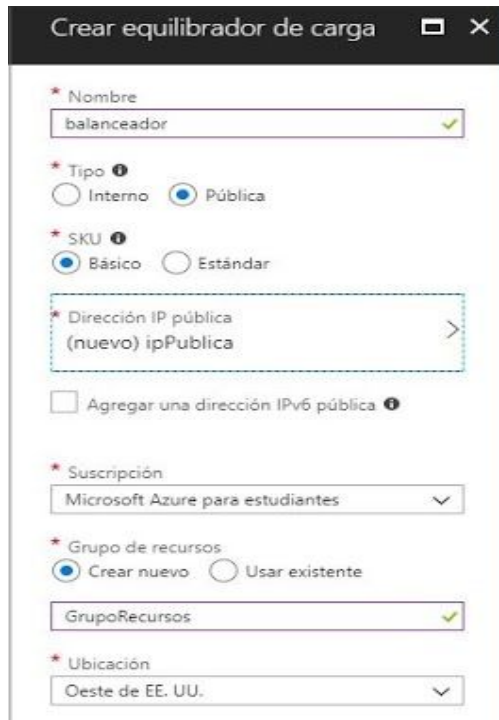
La licencia de Microsoft Azure tiene un coste gratuito de uso. Sólo se paga por los recursos que se usen. Gracias al convenio de DreamSpark de Microsoft con la UGR hemos podido obtener una licencia de 200\$ para uso de recursos de Microsoft Azure.

Como hemos comentado antes, Microsoft Azure es un servicio en la nube y por lo tanto no se necesita ningún tipo de instalación software.

Para acceder a este servicio sólo es necesario ir a la web de Azure e iniciar sesión (<https://azure.microsoft.com/es-es/>).

# 7- CONFIGURACIÓN DEL BALANCEADOR DE CARGA USANDO AZURE

Para realizar la configuración se ha seguido el tutorial disponible en la documentación oficial de Azure.



The screenshot shows the 'Crear equilibrador de carga' (Create Load Balancer) form. The fields are filled as follows: Name: 'balanceador'; Type: 'Pública' (Public); SKU: 'Básico' (Basic); Public IP address: '(nuevo) ipPublica' (new ipPublica); Subscription: 'Microsoft Azure para estudiantes'; Resource Group: 'GrupoRecursos'; Location: 'Oeste de EE. UU.' (West US). The 'Agregar una dirección IPv6 pública' (Add a public IPv6 address) checkbox is unchecked.

## 7.1 - Creación del balanceador de carga

En primer lugar se debe de crear una instancia pública de Balanceador de Carga. Para ello se selecciona la opción “Equilibrador de carga” del panel de Azure y se configura indicando el nombre del balanceador, el tipo de balanceador (público/privado), dirección IP (asignada automáticamente), creación de un grupo de recursos y otros aspectos.

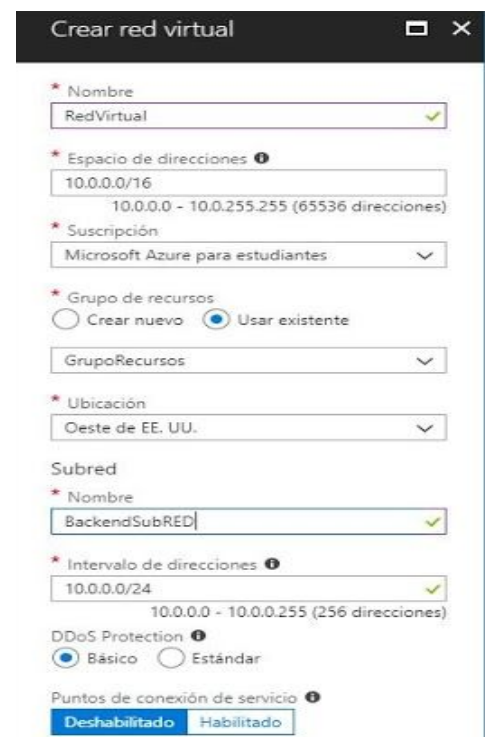
Una vez rellenados estos campos tendríamos creado el balanceador de carga.

A continuación se configuran los servidores back-end. Para ello se crea una red virtual y 2 máquinas virtuales.

## 7.2- Creación de una red virtual

Para crear una red virtual, Microsoft Azure ofrece la posibilidad de hacerlo mediante el apartado “Redes” que se encuentra en el panel. Una vez ahí se selecciona “Red Virtual” y se configura dicha red.

Para la configuración se deben de rellenar unos campos especificando el nombre, el grupo de recursos a la que se asocia y el nombre de la subred.



The screenshot shows the 'Crear red virtual' (Create Virtual Network) form. The fields are filled as follows: Name: 'RedVirtual'; Address space: '10.0.0.0/16'; Subscription: 'Microsoft Azure para estudiantes'; Resource Group: 'GrupoRecursos'; Location: 'Oeste de EE. UU.' (West US). Under 'Subred' (Subnet), the Name is 'BackendSubRED'. The 'Intervalo de direcciones' (Address range) is '10.0.0.0/24'. The 'DDoS Protection' (DDoS Protection) is set to 'Básico' (Basic). The 'Puntos de conexión de servicio' (Service connection points) are set to 'Deshabilitado' (Disabled).

### 7.3- Creación de máquinas virtuales

Azure permite virtualizar distintos sistemas (Windows Server, Ubuntu Server, SQL Server, RedHat Linux...). Nosotros hemos decidido virtualizar 2 máquinas con Ubuntu Server para realizar el balanceo de carga.

Para seleccionar el sistema a virtualizar debemos seleccionar en el apartado “Nuevo” de Azure “Proceso” y una vez ahí escoger Ubuntu Server.

Para realizar la configuración debemos de seguir 4 pasos:

1. Indicar el nombre de la máquina, el nombre de usuario, el grupo de recursos, etc...
2. Seleccionar el tamaño de la máquina virtual (en este punto hay que tener en cuenta que cada configuración tiene un coste).
3. Especificar los valores de la configuración de la máquina virtual, entre estos valores se encuentran: nombre del conjunto de disponibilidad, elección de subred, ip, firewall, etc ...
4. El último paso consiste simplemente en revisar la configuración y aceptar dicha configuración.

Una vez finalizados estos 4 pasos tendremos disponible la máquina. Hay que repetir dichos pasos de configuración para cada una de las máquinas que se quieran virtualizar.

The screenshot shows the 'Crear máquina virtual' (Create Virtual Machine) wizard in the Azure portal, specifically the 'Básico' (Basic) step. The wizard is divided into two main sections: a left sidebar with step indicators and a main configuration area on the right.

**Left Sidebar (Step Indicators):**

- 1 Básico** (Selected): Definir configuración básica
- 2 Tamaño**: Elegir tamaño de máquina virt...
- 3 Configuración**: Configurar características opc...
- 4 Resumen**: Ubuntu Server 17.10

**Main Configuration Area (Básico):**

- \* Nombre**: Ubuntu-Server1 (Validated with a green checkmark)
- Tipo de disco de máquina virtual**: SSD (Selected from a dropdown)
- \* Nombre de usuario**: jesra (Validated with a green checkmark)
- \* Tipo de autenticación**: Clave pública SSH (Selected, with a 'Contraseña' button also visible)
- \* Clave pública SSH**: ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQC02M9q50Q9kTUHGmsWDI61ZVUS09 (Validated with a green checkmark)
- Suscripción**: Microsoft Azure para estudiantes (Selected from a dropdown)
- \* Grupo de recursos**:   
 ☐ Crear nuevo ☒ Usar existente   
 GrupoRecursos (Selected from a dropdown)
- \* Ubicación**: Oeste de EE. UU. (Selected from a dropdown)



## **7.4- Configuración extra tras la creación de las máquinas**

Una vez terminado el proceso de creación de las máquinas, hay que instalar LAMP en cada una de las máquina.

Para ello se selecciona la máquina y mediante la terminal que ofrece Azure se instala. Para acceder a cada máquina desde la terminal se hace mediante conexión ssh (previamente durante la configuración de la máquina se ha creado una contraseña ssh mediante el comando “ssh-keygen -t rsa -b 2048”).

## **7.5- Creación de reglas NSG**

Llegados a este punto de la configuración se debe de realizar una configuración de reglas de NSG. Estas siglas se refieren a los recursos Network Security Group asociados al grupo del balanceador de carga.

Estas reglas permitirán las conexiones entrantes que usen HTTP y RDP.

La conexión RDP, conocida como Remote Desktop Protocol, es un protocolo desarrollado por Microsoft que permite la comunicación en la ejecución de una aplicación entre una terminal y un servidor Windows.

Para llevar a cabo la configuración de estas reglas , primero se hace sobre HTTP y debemos situarnos en el panel de Azure sobre “Todos los recursos” y seleccionar el grupo de seguridad del grupo de recursos del balanceador. Una vez ahí hay que ir a “Reglas de seguridad de entrada” y agregar una nueva regla. Hay que rellenar algunos campos para permitir conexiones HTTP y abrir el puerto 80, para ello se siguen unos pasos muy intuitivos que aparecen en Azure.

Para configurar la regla para RDP se siguen los mismo pasos pero abriendo el puerto 3389.

Agregar regla de seguridad de entrada ✕  
Ubuntu-Server1-nsg

✈️ Básica

\* Origen ⓘ  
Service Tag

\* Etiqueta de servicio de origen ⓘ  
Internet

\* Intervalos de puertos de origen ⓘ  
\*

\* Destino ⓘ  
Any

\* Intervalos de puertos de destino ⓘ  
80 ✓

\* Protocolo  
Any TCP UDP

\* Acción  
Permitir Denegar

\* Prioridad ⓘ  
100 ✓

\* Nombre  
PuertoHTTP ✓

Descripción  
Apertura del puerto 80 para dar acceso al HTTP

## **7.6- Creación de un grupo de direcciones de back-end**

Para distribuir el tráfico a las máquinas virtuales debe de existir un grupo de direcciones de back-end, que contiene las direcciones IP de las tarjetas de interfaz de red virtual conectadas al balanceador de carga. Por ello se debe de crear el grupo de direcciones de back-end para incluir las 2 máquinas virtuales creadas.

Para crear este grupo hay que situarse en el recurso asociado al balanceador en el panel de Azure y darle a “Agregar grupo back-end”.

A dicho grupo hay que realizarle la siguiente configuración: añadir nombre, asociarlo al conjunto de disponibilidad creado anteriormente y asociar las 2 máquinas (IP's).

Inicio > Balanceador - Grupos de back-end > Agregar grupo back-end

### Agregar grupo back-end

Balanceador

\* Nombre  
GrupoDisponibilidad ✓

Versión de IP  
**IPv4** IPv6

Asociado a ⓘ  
Conjunto de disponibilidad ▼

Conjunto de disponibilidad ⓘ  
GrupoDisponibilidadBackend  
Número de máquinas virtuales: 2 ▼

Configuraciones IP de red de destino  
Solo se pueden elegir las máquinas virtuales con el conjunto de disponibilidad actual. Una vez elegida una VM, puede seleccionar una configuración IP de red relacionada con esta.

Máquina virtual: Ubuntu-Server1 Configuración IP de red: ubuntu-server1656/ipconfig1 (10.0.0.6)	🗑️
Máquina virtual: Ubuntu-Server2 Configuración IP de red: ubuntu-server2210/ipconfig1 (10.0.0.7)	🗑️

+ Agregar una configuración IP de red de destino

## **7.7- Creación de un sondeo de estado**

Para permitir que la instancia del balanceador de carga supervise el estado de la aplicación se usa un sondeo de estado. El sondeo de estado agrega o quita de forma dinámica las máquinas virtuales de la rotación del balanceador de carga en base a su respuesta a las comprobaciones de estado.

Para configurar el sondeo se debe de agregar esta opción en el recurso asociado al balanceador.

Una vez añadido se debe de configurar el sondeo con los siguientes valores:

- *myHealthProbe*: como nombre del sondeo de estado.
- **HTTP**: en tipo de protocolo.
- *80*: en número de puerto.
- *15*: como número de **Intervalo**, en segundos, entre los intentos de sondeo.
- *2*: como número de **Umbral incorrecto** o errores de sondeo consecutivos que deben producirse para que una máquina virtual se considere que no funciona de manera incorrecta.

**Agregar sondeo de estado**  
Balanceador

\* Nombre  
SondeoDeEstado ✓

Versión de IP  
IPv4

Protocolo  
**HTTP** TCP

\* Puerto  
80

\* Ruta de acceso ⓘ  
Healthprobe.aspx ✓

\* Intervalo ⓘ  
15 segundos ✓

\* Umbral incorrecto ⓘ  
2 errores consecutivos ✓

## 7.8- Creación de una regla de Load Balancer

Las reglas de Load Balancer se utilizan para definir cómo se distribuye el tráfico a las máquinas virtuales. Se define la configuración de IP front-end para el tráfico entrante y el grupo IP de back-end para recibir el tráfico, junto con el puerto de origen y destino requeridos.

Para crear una regla Load Balancer debemos de seleccionar en el recurso del balanceador “Reglas de equilibrio de carga”.

Para configurarlo se deben de usar los siguientes valores:

- *myHTTPRule*: como nombre de la regla de equilibrio de carga.
- **TCP**: en tipo de protocolo.
- **80**: en número de puerto.
- **80**: como puerto de back-end.
- *myBackendPool*: como nombre del grupo back-end.
- *myHealthProbe*: como nombre del sondeo de estado.

ReglaHTTP
Balanceador

Guardar
Descartar
Eliminar

\* Nombre
ReglaHTTP

\* Versión de IP
☒ IPv4
☐ IPv6

\* Dirección IP de front-end ⓘ
104.210.50.253 (LoadBalancerFrontEnd)

Protocolo
☒ TCP
☐ UDP

\* Puerto
80

\* Puerto back-end ⓘ
80

Grupo de back-end ⓘ
GrupoDisponibilidad (2 máquinas virtuales)

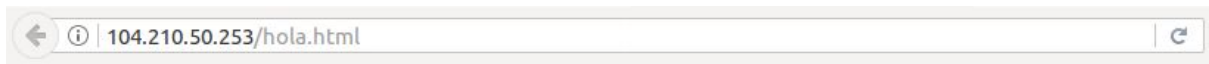
Sondeo de estado ⓘ
SondeoDeEstado (HTTP:80)

Persistencia de la sesión ⓘ
Ninguno

Tiempo de espera de inactividad (minutos) ⓘ
 4

## 8- PRUEBA DEL BALANCEADOR

Tras la configuración del balanceador, este ya funciona. A continuación se muestran 2 capturas de pantalla en las que se conecta desde el navegador a la ip del balanceador y se muestra el archivo “hola.html”. Dicho archivo muestra la máquina en la que se encuentra.



Estas en la maquina 1

*Máquina 1*



Estas en la maquina 2

*Máquina 2*

**Pruebas sometiendo a alta carga el servidor** (en la 1º imagen no tiene carga, en la 2º imagen se le está enviando carga, se puede apreciar el cambio en el valor del uso de la CPU). Para ello hemos usado Apache Benchmark y el comando “`ab -n 1000 -c 10 http://104.210.50.253/hola.html`”

```
jesra@Ubuntu-ServerAB: ~
jesra@Ubuntu-ServerAB:~$

CPU[ ] 0.7% Tasks: 44, 53 thr: 1 running
Mem[ ] 419M/3.34G Load average: 0.20 0.19 0.08
Swp[ ] 0K/0K Uptime: 1 day, 07:04:12

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
34536 jesra 20 0 16804 3492 2896 R 0.7 0.1 0:02.33 htop
1572 root 20 0 230M 40600 9080 S 0.0 1.2 8:16.56 python
1583 root 20 0 230M 40600 9080 S 0.0 1.2 1:28.73 python
25024 mysql 20 0 1076M 132M 17092 S 0.0 3.9 0:33.00 /usr/s
26047 root 20 0 308M 34684 28732 S 0.0 1.0 0:03.15 /usr/s
25036 mysql 20 0 1076M 132M 17092 S 0.0 3.9 0:01.65 /usr/s
25041 mysql 20 0 1076M 132M 17092 S 0.0 3.9 0:02.12 /usr/s
1428 root 20 0 461M 18188 10180 S 0.0 0.5 0:01.05 /usr/s
1187 root 20 0 5096 900 836 S 0.0 0.0 0:04.33 /usr/s
35225 www-data 20 0 308M 9136 3128 S 0.0 0.3 0:00.21 /usr/s
35230 www-data 20 0 308M 9380 3236 S 0.0 0.3 0:00.20 /usr/s
1186 unsd 20 0 15052 1780 1620 S 0.0 0.1 0:05.27 /usr/s
35193 www-data 20 0 308M 9136 3128 S 0.0 0.3 0:00.29 /usr/s
35248 www-data 20 0 308M 9380 3232 S 0.0 0.3 0:00.19 /usr/s
35222 www-data 20 0 308M 9136 3128 S 0.0 0.3 0:00.22 /usr/s
35242 www-data 20 0 308M 9132 3124 S 0.0 0.3 0:00.20 /usr/s
35219 www-data 20 0 308M 9136 3128 S 0.0 0.3 0:00.22 /usr/s
F1:help F2:Setup F3:Search F4:Filter F5:Free F6:SortBy F7:Nice F8:Nice F9:All

jesra@Ubuntu-Server2: ~
jesra@Ubuntu-Server2:~$

CPU[ ] 1.3% Tasks: 44, 45 thr: 1 running
Mem[ ] 393M/3.34G Load average: 0.21 0.21 0.08
Swp[ ] 0K/0K Uptime: 06:37:57

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
33225 jesra 20 0 16796 3420 2932 R 0.0 0.1 0:00.69 htop
1264 mysql 20 0 1076M 131M 16368 S 0.0 3.8 0:06.96 /usr/s
1519 root 20 0 226M 36152 9080 S 0.0 1.0 0:17.45 python
1463 root 20 0 461M 18720 10856 S 0.0 0.5 0:00.18 /usr/s
1494 mysql 20 0 1076M 131M 16368 S 0.0 3.8 0:00.86 /usr/s
33201 jesra 20 0 103M 4180 3140 S 0.0 0.1 0:00.01 sshd:
1587 mysql 20 0 1076M 131M 16368 S 0.0 3.8 0:00.66 /usr/s
1589 mysql 20 0 1076M 131M 16368 S 0.0 3.8 0:00.75 /usr/s
32393 www-data 20 0 308M 9140 3124 S 0.0 0.3 0:00.83 /usr/s
```

```
jesra@Ubuntu-ServerAB: ~
jesra@Ubuntu-ServerAB:~$ ab -n 100000 -c 10 http://104.210.50.253/hola.html
This is ApacheBench, Version 2.3 <Revision: 1796539> $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 104.210.50.253 (be patient)
Completed 10000 requests
Completed 20000 requests
Completed 30000 requests
Completed 40000 requests
Completed 50000 requests
Completed 60000 requests
Completed 70000 requests
Completed 80000 requests
Completed 90000 requests
Completed 100000 requests

jesra@Ubuntu-Server1: ~
CPU: [|||||] 43.2% Tasks: 45, 53 thr: 7 running
Mem: [|||||] 421M/3.34G Load average: 1.20 0.40 0.15
Swp: [|||||] 0K/0K Uptime: 1 day, 07:04:42

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
35248 www-data 20 0 308M 9140 3128 R 4.7 0.3 0:00.81 /usr/s
35219 www-data 20 0 308M 9140 3128 R 4.7 0.3 0:00.84 /usr/s
35243 www-data 20 0 308M 9136 3124 S 4.7 0.3 0:00.82 /usr/s
35193 www-data 20 0 308M 9140 3128 S 4.1 0.3 0:00.91 /usr/s
35222 www-data 20 0 308M 9140 3128 S 4.1 0.3 0:00.84 /usr/s
35470 www-data 20 0 308M 9128 3128 R 4.1 0.3 0:00.52 /usr/s
35215 www-data 20 0 308M 9384 3236 S 4.1 0.3 0:00.84 /usr/s
35485 www-data 20 0 308M 9128 3128 R 4.1 0.3 0:00.14 /usr/s
35225 www-data 20 0 308M 9140 3128 R 3.4 0.3 0:00.65 /usr/s
35201 www-data 20 0 308M 9140 3128 S 3.4 0.3 0:00.80 /usr/s
1572 root 20 0 230M 40600 9080 S 2.0 1.2 8:16.82 python
34538 jesra 20 0 16804 3492 2996 R 0.7 0.1 0:02.44 htop
1187 root 20 0 5096 900 836 S 0.7 0.0 0:04.34 /usr/t
1583 root 20 0 230M 40600 9080 S 0.0 1.2 1:28.75 python
35242 www-data 20 0 308M 9136 3124 S 0.0 0.3 0:00.65 /usr/s
1186 unsd 20 0 15052 1780 1620 S 0.0 0.1 0:05.28 /usr/s
34486 jesra 20 0 103M 4232 3204 S 0.0 0.1 0:00.03 sshd:

F1 Help F2 Setup F3 Search F4 Filter F5 Free F6 Sorts F7 Vics F8 Vics F9 All

jesra@Ubuntu-Server2: ~
CPU: [|||||] 47.0% Tasks: 45, 45 thr: 1 running
Mem: [|||||] 394M/3.34G Load average: 0.35 0.24 0.10
Swp: [|||||] 0K/0K Uptime: 06:38:28

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
33125 www-data 20 0 308M 9140 3124 S 4.0 0.3 0:01.12 /usr/s
33551 www-data 20 0 0 0 0 Z 4.0 0.0 0:00.47 /usr/s
33308 www-data 20 0 308M 9140 3124 S 4.0 0.3 0:01.15 /usr/s
33316 www-data 20 0 308M 9140 3124 S 4.0 0.3 0:01.12 /usr/s
33319 www-data 20 0 308M 9384 3236 S 4.0 0.3 0:01.12 /usr/s
33560 www-data 20 0 308M 9128 3124 S 4.0 0.3 0:00.23 /usr/s
33374 www-data 20 0 308M 9140 3124 S 4.0 0.3 0:00.69 /usr/s
33568 www-data 20 0 308M 9128 3124 S 4.0 0.3 0:00.07 /usr/s
33567 www-data 20 0 308M 9128 3124 S 3.3 0.3 0:00.14 /usr/s
33546 www-data 20 0 308M 9128 3124 S 2.6 0.3 0:00.48 /usr/s
```

## 9- CONCLUSIONES Y OPINIÓN

Tras haber probado la configuración de un balanceador de carga con Microsoft Azure, podemos afirmar que la herramienta de Microsoft es interesante para su uso como herramienta de configuración para el balanceo.

Este servicio ofrece una gran cantidad de opciones y su configuración es sencilla, además Microsoft ofrece una gran cantidad de tutoriales y manuales PDF para la correcta configuración de sus servicios disponibles en Azure.

Si en algún momento tenemos que recomendar su uso, diríamos si.



# 10- BIBLIOGRAFÍA

## **Microsoft Azure, ¿qué es?**

<https://azure.microsoft.com/es-es/>

<https://azure.microsoft.com/es-es/overview/what-is-azure/>

[https://es.wikipedia.org/wiki/Microsoft\\_Azure#Servicio\\_de\\_Windows\\_Azure](https://es.wikipedia.org/wiki/Microsoft_Azure#Servicio_de_Windows_Azure)

<http://www.prodware.es/soluciones/cloud-business/microsoft-azure>

<https://www.xataka.com/analisis/como-es-azure-la-nueva-estrella-dentro-de-microsoft>

## **Microsoft Azure, ¿que no ofrece?**

<https://azure.microsoft.com/es-es/overview/what-is-azure/>

## **Configuración del balanceador de carga usando Azure**

<https://docs.microsoft.com/es-es/azure/load-balancer/quickstart-create-basic-load-balancer-portal>

<https://docs.microsoft.com/en-us/azure/virtual-machines/linux/tutorial-load-balancer>