

## **Laboratorio 8 Escalamiento y alta disponibilidad**

**Sebastián Cardona, Laura Gil, Zayra Gutiérrez**

**Ingeniero**  
**Javier Toquica Barrera**

**Universidad Escuela Colombiana de ingeniería Julio Garavito**

## Contenido

|  |    |
|--|----|
| Introducción.....  | 3  |
| Desarrollo del Laboratorio .....                           | 4  |
| Parte 0 .....  | 4  |
| Parte 1 – Escalabilidad Vertical.....                      | 5  |
| Preguntas parte-1: .....                                   | 23 |
| Parte 2 – Escalabilidad Horizontal.....                    | 27 |
| Crear las máquinas virtuales (Nodos) .....                 | 31 |
| Probar el resultado final de nuestra infraestructura ..... | 36 |
| Preguntas - Parte 2: .....                                 | 40 |
| Conclusiones .....   | 46 |

## Introducción

Este laboratorio tiene como objetivo explorar y aplicar los conceptos de escalabilidad y alta disponibilidad en la plataforma de Microsoft Azure, utilizando una aplicación que calcula el enésimo número de la secuencia de Fibonacci como caso de estudio. A través de dos enfoques principales (escalabilidad vertical y horizontal), se busca garantizar que el sistema pueda manejar cargas concurrentes de usuarios solicitando números superiores a 1,000,000, manteniendo el consumo de CPU por debajo del 70% y respondiendo a todas las peticiones bajo condiciones normales de operación. En la primera parte, se implementa una solución de escalabilidad vertical ajustando el tamaño de una máquina virtual (VM), mientras que en la segunda parte se emplea un balanceador de carga para distribuir el tráfico entre múltiples VMs, logrando escalabilidad horizontal. Este proceso permite analizar el rendimiento, los costos y los desafíos asociados a cada estrategia, ofreciendo una comprensión práctica de cómo Azure facilita la gestión de recursos para cumplir con requerimientos no funcionales de disponibilidad y escalabilidad.

Enlace Repositorio: <https://github.com/SebastianCardona-P/ARSW-Lab8>

Enlace Video Demostrativo: <https://youtu.be/eLGz29rNTcl>

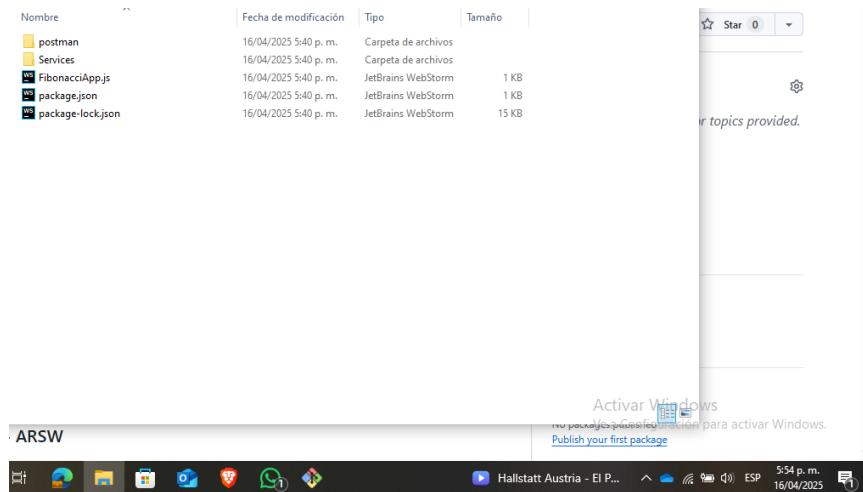
# Desarrollo del Laboratorio

## Parte 0

Adjunto a este laboratorio usted podrá encontrar una aplicación totalmente desarrollada que tiene como objetivo calcular el enésimo valor de la secuencia de Fibonacci.

**Escalabilidad** Cuando un conjunto de usuarios consulta un enésimo número (superior a 1000000) de la secuencia de Fibonacci de forma concurrente y el sistema se encuentra bajo condiciones normales de operación, todas las peticiones deben ser respondidas y el consumo de CPU del sistema no puede superar el 70%.

Efectivamente se encuentra la totalidad de un proyecto simple



## Parte 1 – Escalabilidad Vertical

1. Diríjase a el [Portal de Azure](#) y a continuación cree una maquina virtual con las características básicas descritas en la imagen 1 y que corresponden a las siguientes:

- Resource Group = SCALABILITY\_LAB
- Virtual machine name = VERTICAL-SCALABILITY
- Image = Ubuntu Server
- Size = Standard B1ls
- Username = scalability\_lab
- SSH publi key = Su llave ssh publica

A continuación, procedemos a crear nuestra maquina virtual

Configuración básica:

your resources.

Subscription \*  Resource group \*  Create new

Instance details

Virtual machine name \*

Region \*

Availability options

Security type   
Configure security features  
Trusted launch virtual machine is required when using 1P Gallery images.

Image \*  See all images | Confidential VMs source info

Activar Windows  
Ve a Configuración para activar | Give feedback

< Previous | Next : Disks > | Review + create

VM architecture  ARM64

Run with Azure Spot discount

Size \*  See all sizes

Enable Hibernation   
Hibernate does not currently support Trusted launch and Confidential virtual machines for Linux images. [Learn more](#)

Administrator account

Authentication type  Password

Azure now automatically generates an SSH key pair for you and allows you to store it for future use. It is a fast, simple, and secure way to connect to your virtual machine.

Activar Windows  
Ve a Configuración para activar | Give feedback

< Previous | Next : Disks > | Review + create

Username \*

SSH public key source

SSH Key Type    
Ed25519 provides a fixed security level of no more than 128 bits for 256-bit key, while RSA could offer better security with keys longer than 3072 bits.

Key pair name \*

Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

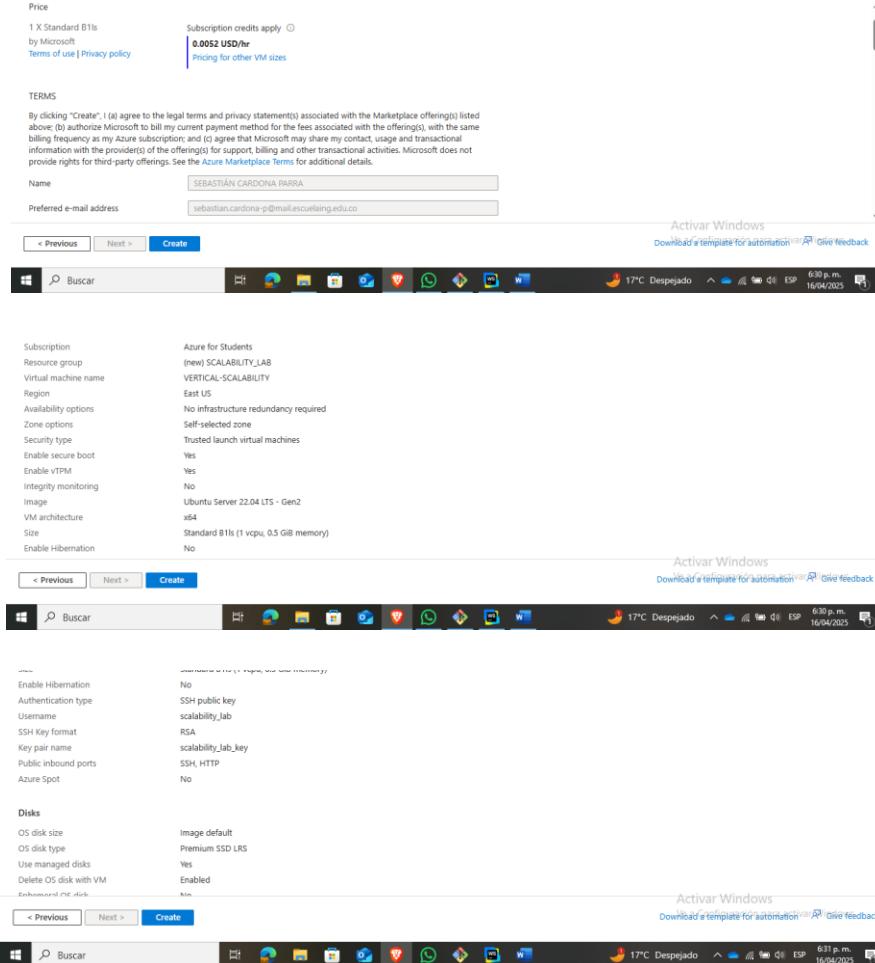
Public inbound ports \*  None

Select inbound ports

Activar Windows  
Ve a Configuración para activar | Give feedback

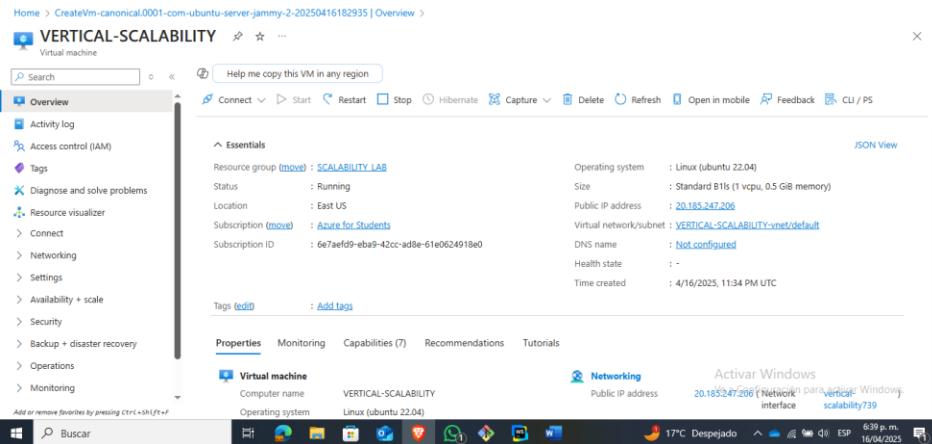
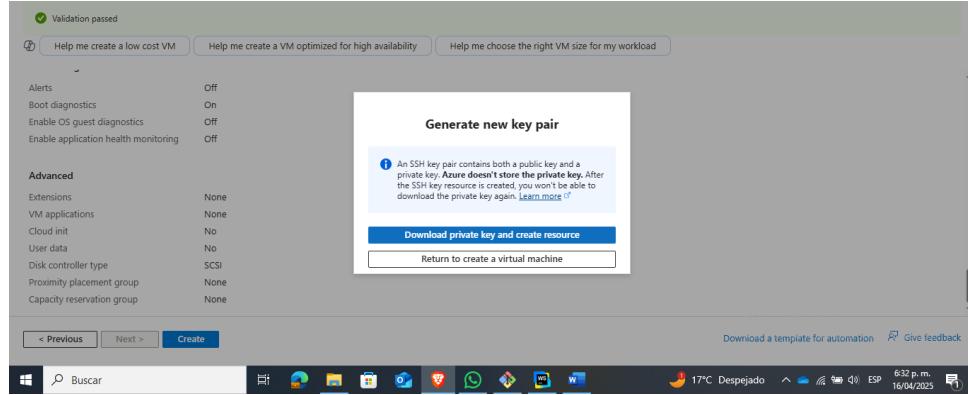
< Previous | Next : Disks > | Review + create

Como siguiente paso, seleccionaremos review + créate



Posteriormente en créate

Como dejamos que azure generara el par de claves para ssh, debemos guardar la clave privada:



- Para conectarse a la VM use el siguiente comando, donde las x las debe remplazar por la IP de su propia VM (Revise la sección "Connect" de la virtual machine creada para tener una guía más detallada).

```
ssh scalability_lab@xxx.xxx.xxx.xxx
```

Nos conectamos desde nuestro CMD y nos situamos donde guardamos la llave privada, para conectarnos usamos el siguiente comando:

```
ssh -i ./scalability_lab_key.pem scalability_lab@20.185.247.206
```

```
C:\Users\Sebastian\Desktop\Universidad\Octavo Semestre\ARSW\Laboratorios\Lab08\FibonacciApp>ssh -i ./scalability_lab_key .pm scalability_lab@20.185.247.206
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1026-azure x86_64)

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

System information as of Wed Apr 16 23:45:48 UTC 2025

System load: 0.08 Processes: 106
Usage of /: 6.8% of 28.89GB Users logged in: 0
Memory usage: 70% IPv4 address for eth0: 10.0.0.4
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

10 updates can be applied immediately.
5 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 programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

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

scalability_lab@VERTICAL-SCALABILITY:~$
```

Activar Windows  
Ve a Configuración para activar Windows.

Español (Colombia) Predicciones de texto activado Accesibilidad: es necesario investigar Concentración

647 p.m. 16/04/2025

```
scalability_lab@VERTICAL-SCALABILITY:~$ Swap usage: 0%
Expanded Security Maintenance for Applications is not enabled.

10 updates can be applied immediately.
5 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 programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

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

scalability_lab@VERTICAL-SCALABILITY:~$
```

Activar Windows  
Ve a Configuración para activar Windows.

BRKB 6:49 p.m. 16/04/2025

3. Instale node, para ello siga la sección *Installing Node.js and npm using NVM* que encontrará en este [enlace](#).

Simplemente debemos correr estos comandos

Sudo apt update

sudo apt install nodejs npm

```
Unpacking libcurlpp1-amd64 (1:1.2.0-1) ...
Unpacking previously unselected package libidn11-amd64 ...
Unpacking libimage0-amd64 (1:1.5-2+deb10u1_amd64.deb) ...
Unpacking liblxdege1-amd64 (1:1.1.5-2+deb10u2_amd64.deb) ...
Selecting previously unselected package liblxinerama1-amd64.
Preparing to unpack .../lxinerama1_1.4-3_amd64.deb ...
Selecting previously unselected package liblxcommon0-amd64.
Preparing to unpack .../lxcommon_1.4.0-1_amd64.deb ...
Selecting previously unselected package liblxrandr2-amd64.
Preparing to unpack .../lxrandr2_2.3@1.1.2-2+deb10u1_amd64.deb ...
Unpacking liblxrandr2-amd64 (2.1.5.2-2+deb10u1) ...
Selecting previously unselected package liblxcommon3-common ...
Preparing to unpack .../lxcommon3-common_3.24.33~ubuntu2.2_all.deb ...
Unpacking liblxcommon3-common (3.24.33~ubuntu2.2) ...
Selecting previously unselected package liblxcommon3-amd64 ...
Preparing to unpack .../lxcommon3-amd64_3.24.33~ubuntu2.2_amd64.deb ...
Unpacking liblxcommon3-amd64 (3.24.33~ubuntu2.2) ...
Selecting previously unselected package liblgbt3-bin ...
Selecting previously unselected package liblgbt3-3.24.33~ubuntu2.2_amd64.deb ...
Unpacking liblgbt3-bin (3.24.33~ubuntu2.2) ...
Selecting previously unselected package librsvg2-2-amd64.
Preparing to unpack .../rsvg2_2.52.5-0+deb10u2_amd64.deb ...
Unpacking librsvg2-2-amd64 (2.51.5-0+deb10u2_amd64) ...
Selecting previously unselected package libllywm1-amd64.
Preparing to unpack .../llywm1_1.0-6_amd64.deb ...
Unpacking libllywm1-amd64 (1:1.0-6) ...

[unprocess] [ 45% #####]
```

Cuando haya acabado de instalar, cerificamos con los siguientes comandos:

nodejs –version

npm –version

```
killall  
scalability_lab@VERTICAL-SCALABILITY:~$ scalability_lab@VERTICAL-SCALABILITY:~$ ##### scalability_lab@VERTICAL-SCALABILITY:~$ scalability_lab@VERTICAL-SCALABILITY:~$ npm --version  
8.5.1  
scalability_lab@VERTICAL-SCALABILITY:~$ nodejs --version  
v12.22.9  
scalability_lab@VERTICAL-SCALABILITY:~$ scalability_lab@VERTICAL-SCALABILITY:~$ sudo apt install nodejs npm
```

4. Para instalar la aplicación adjunta al Laboratorio, suba la carpeta FibonacciApp a un repositorio al cual tenga acceso y ejecute estos comandos dentro de la VM:

```
git clone <your_repo>
```

```
cd <your_repo>/FibonacciApp
```

npm install

usamos git clone <https://github.com/SebastianCardona-P/ARSW-Lab8.git>

```
nodejs is already the newest version (12.22.9~dfsg-1ubuntu3.6).
0 upgraded, 0 newly installed, 0 to remove and 10 not upgraded.
scalability_lab@VERTICAL-SCALABILITY:~$ git clone https://github.com/SebastianCardona-P/ARSW-Lab8.git
Cloning into 'ARSW-Lab8'...
remote: Enumerating objects: 34, done.
remote: Counting objects: 100% (34/34), done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 34 (delta 3), reused 34 (delta 3), pack-reused 0 (from 0)
Receiving objects: 100% (34/34), 3.55 MiB | 26.51 MiB/s, done.
Resolving deltas: 100% (3/3), done.
scalability_lab@VERTICAL-SCALABILITY:~$
```



Luego de ello corremos npm install:

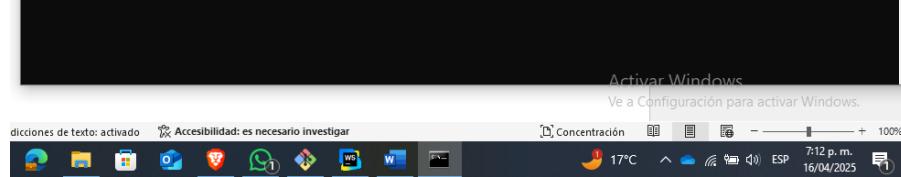
```
FibonacciApp.js Services package-lock.json package.json postman
scalability_lab@VERTICAL-SCALABILITY:~/ARSW-Lab8/FibonacciApp$ npm install
npm WARN old lockfile
npm WARN old lockfile The package-lock.json file was created with an old version of npm,
npm WARN old lockfile so supplemental metadata must be fetched from the registry.
npm WARN old lockfile
npm WARN old lockfile This is a one-time fix-up, please be patient...
npm WARN old lockfile

added 51 packages, and audited 52 packages in 5s

7 vulnerabilities (3 low, 4 high)

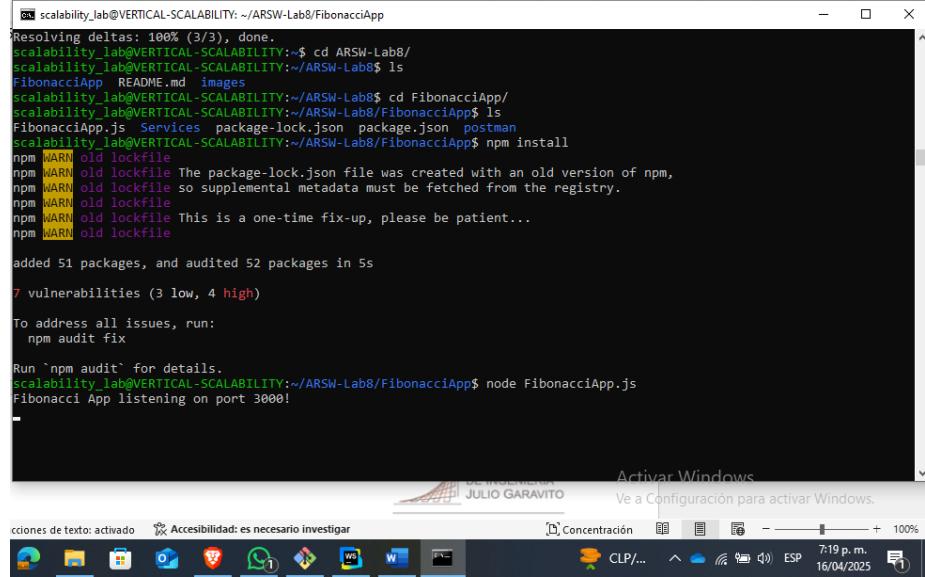
To address all issues, run:
  npm audit fix

Run `npm audit` for details.
scalability_lab@VERTICAL-SCALABILITY:~/ARSW-Lab8/FibonacciApp$
```



- Para ejecutar la aplicación puede usar el comando `npm FibinacciApp.js`, sin embargo una vez pierda la conexión ssh la aplicación dejará de funcionar. Para evitar ese comportamiento usaremos `forever`. Ejecute los siguientes comando dentro de la VM.

`node FibonaciApp.js`



```

scalability_lab@VERTICAL-SCALABILITY:~/ARSW-Lab8/FibonacciApp
Resolving deltas: 100% (3/3), done.
scalability_lab@VERTICAL-SCALABILITY:~$ cd ARSW-Lab8/
scalability_lab@VERTICAL-SCALABILITY:~/ARSW-Lab8$ ls
FibonacciApp README.md images
scalability_lab@VERTICAL-SCALABILITY:~/ARSW-Lab8$ cd FibonacciApp/
scalability_lab@VERTICAL-SCALABILITY:~/ARSW-Lab8/FibonacciApp$ ls
FibonacciApp.js Services package-lock.json package.json postman
scalability_lab@VERTICAL-SCALABILITY:~/ARSW-Lab8/FibonacciApp$ npm install
npm WARN old lockfile
npm WARN old lockfile The package-lock.json file was created with an old version of npm,
npm WARN old lockfile so supplemental metadata must be fetched from the registry.
npm WARN old lockfile
npm WARN old lockfile This is a one-time fix-up, please be patient...
npm WARN old lockfile

added 51 packages, and audited 52 packages in 5s

7 vulnerabilities (3 low, 4 high)

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
scalability_lab@VERTICAL-SCALABILITY:~/ARSW-Lab8/FibonacciApp$ node FibonacciApp.js
Fibonacci App listening on port 3000!

```

Activar Windows  
Ve a Configuración para activar Windows.

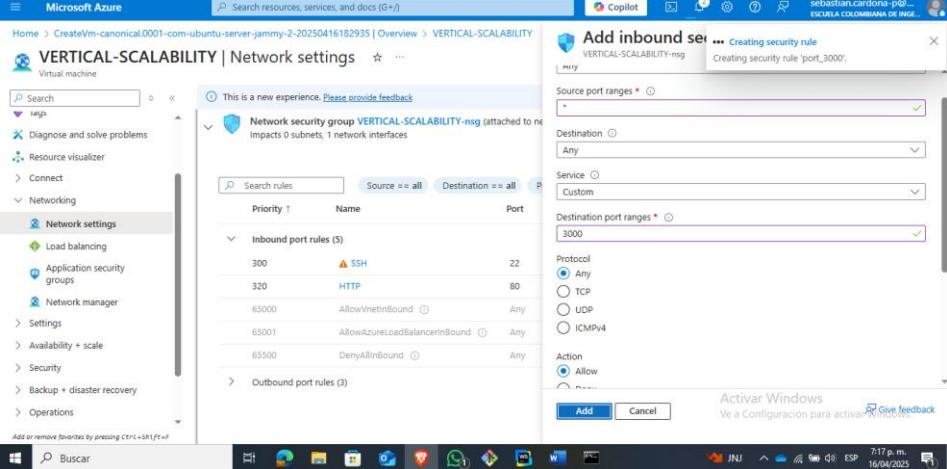
Accesibilidad: es necesario investigar

Concentración

CLP... 7:19 p.m. ESP 16/04/2025

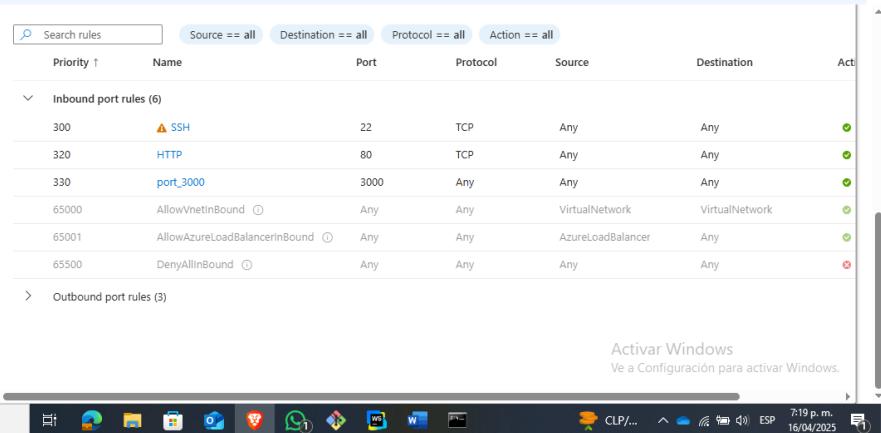
6. Antes de verificar si el endpoint funciona, en Azure vaya a la sección de *Networking* y cree una *Inbound port rule* tal como se muestra en la imagen. Para verificar que la aplicación funciona, use un browser y user el endpoint <http://xxx.xxx.xxx.xxx:3000/fibonacci/6>. La respuesta debe ser The answer is 8.

Configuramos la nueva regla



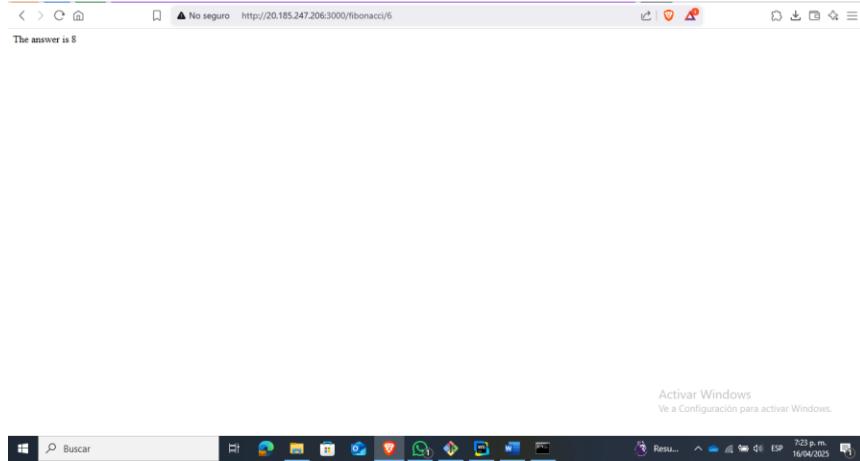
| Priority ↑                | Name                          | Port | Protocol | Source            | Destination    | Action |
|---------------------------|-------------------------------|------|----------|-------------------|----------------|--------|
| Inbound port rules (6)    |                               |      |          |                   |                |        |
| 300                       | SSH                           | 22   | TCP      | Any               | Any            |        |
| 320                       | HTTP                          | 80   | TCP      | Any               | Any            |        |
| 330                       | port_3000                     | 3000 | Any      | Any               | Any            |        |
| 65000                     | AllowVnetInBound              | Any  | Any      | VirtualNetwork    | VirtualNetwork |        |
| 65001                     | AllowAzureLoadBalancerInBound | Any  | Any      | AzureLoadBalancer | Any            |        |
| 65500                     | DenyAllInBound                | Any  | Any      | Any               | Any            |        |
| > Outbound port rules (3) |                               |      |          |                   |                |        |

Activar Windows  
Ve a Configuración para activar Windows.



Ahora bien, en un Browser intentamos el siguiente enlace:

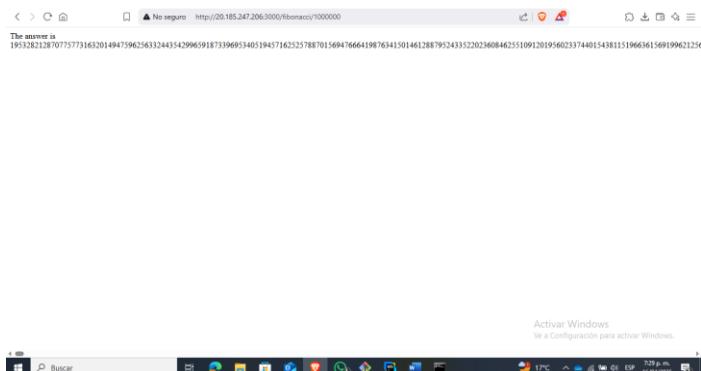
<http://20.185.247.206:3000/fibonacci/6>



Efectivamente ha funcionado

7. La función que calcula en enésimo número de la secuencia de Fibonacci está muy mal construida y consume bastante CPU para obtener la respuesta. Usando la consola del Browser documente los tiempos de respuesta para dicho endpoint usando los siguientes valores:

- 1000000



22 segundos

- 1010000



The answer is  
14694024676932426021742187403844239197547366817041299626508724134697128762028967514967399881719684145021776121767687122795799271271412676422093821872869

28 segundos

- 1020000



The answer is  
116552838445138351515231425332194050623798548970902290993432297730941190110044033772175034125543151372035937203475522868753418076483096885780757751245895240225747783795



Activar Windows  
Ve a Configuración para activar Windows.

24 segundos

- 1030000



The answer is  
8317110584811090520655370445566899120409188497040050215269682455139024711142721586887853818651342269785700081723495938698056940856800818350051290746321879115470238356



Activar Windows  
Ve a Configuración para activar Windows.

23 segundos

- 1040000

The answer is  
6257127084082073809570951191265509646997991495956769352949730030960593841868448787835526496906397667784999856674671943730973410005937686641232491564426987798155



25 segundos

- 1050000

The answer is  
47073699086148102898861227434811789406304777807065754727897221854468847895715838913159850178206828108016903463297211263042989462522186644372694429432898452



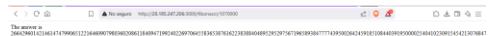
24 segundos

- 1060000



25 segundos

- 1070000



26 segundos

- 1080000



26 segundos

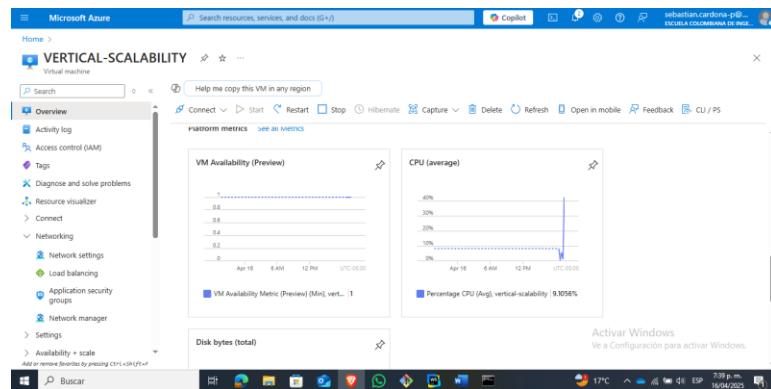
- 1090000

The source is  
130791815083774761030108486796794886291424661087612464743149883902311429614322108466335714810482100048198515422897467291279467040952334721127081978



27 segundos

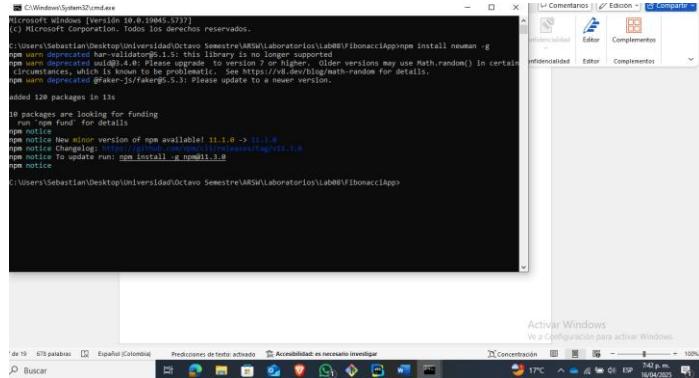
- Dírijase ahora a Azure y verifique el consumo de CPU para la VM. (Los resultados pueden tardar 5 minutos en aparecer).



En total se consumió alrededor de 40% de CPU

- Ahora usaremos Postman para simular una carga concurrente a nuestro sistema. Siga estos pasos.

- Instale newman con el comando `npm install newman -g`. Para conocer más de Newman consulte el siguiente [enlace](#).



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5777]
(c) Microsoft Corporation. Todos los derechos reservados.

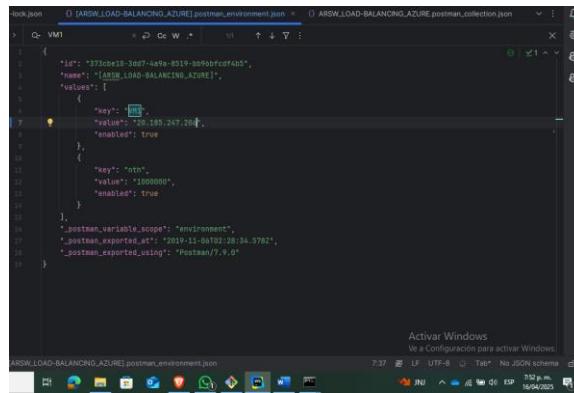
C:\Users\Sebastian\Desktop\Universidad\Octavo Semestre\Lab08\Laboratorios\Lab08\FibonacciApp\npm install Newman -g
npm WARN deprecated unilog@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain
circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated graceful-fs@4.2.1: graceful-fs v4.2.1 is no longer maintained. Upgrade to a newer version.
npm WARN deprecated har-validator@5.1.5: This package is no longer supported. Upgrade to 6.0.0 or higher.

added 128 packages in 13s

10 packages are looking for funding
  run `npm audit fix` for details
npm notice
  npm notice New version of npm available! 11.1.0 -> 11.1.0
  npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.1.0
  npm notice To update run: npm install -g npm@11.1.0
npm notice

C:\Users\Sebastian\Desktop\Universidad\Octavo Semestre\Lab08\Laboratorios\Lab08\FibonacciApp>
```

- Diríjase hasta la ruta FibonacciApp/postman en una maquina diferente a la VM.
- Para el archivo [ARSW\_LOAD-BALANCING\_AZURE].postman\_environment.json cambie el valor del parámetro VM1 para que coincida con la IP de su VM.

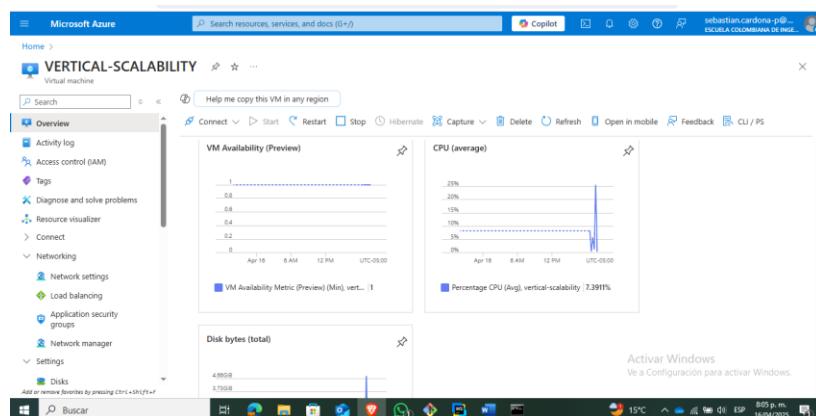


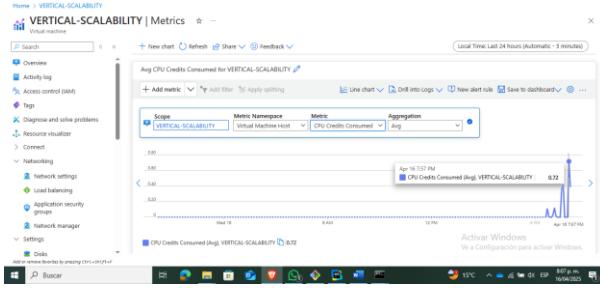
- Ejecute el siguiente comando.

```
newman run ARSW_LOAD-BALANCING_AZURE.postman_collection.json -e
[ARSW_LOAD-BALANCING_AZURE].postman_environment.json -n 10 &

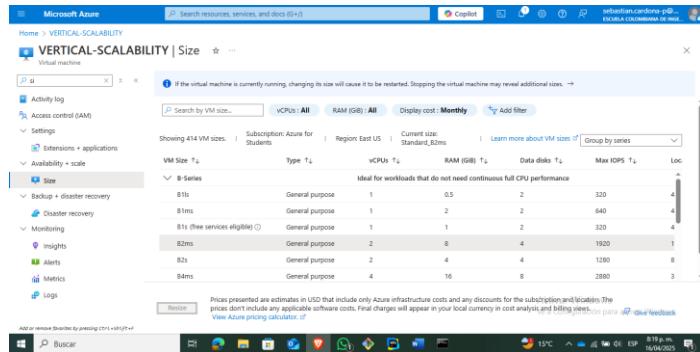
newman run ARSW_LOAD-BALANCING_AZURE.postman_collection.json -e
[ARSW_LOAD-BALANCING_AZURE].postman_environment.json -n 10
```

Realmente disminuyo la capacidad de respuesta del servidor en gran medida





10. La cantidad de CPU consumida es bastante grande y un conjunto considerable de peticiones concurrentes pueden hacer fallar nuestro servicio. Para solucionarlo usaremos una estrategia de Escalamiento Vertical. En Azure diríjase a la sección size y a continuación seleccione el tamaño B2ms.

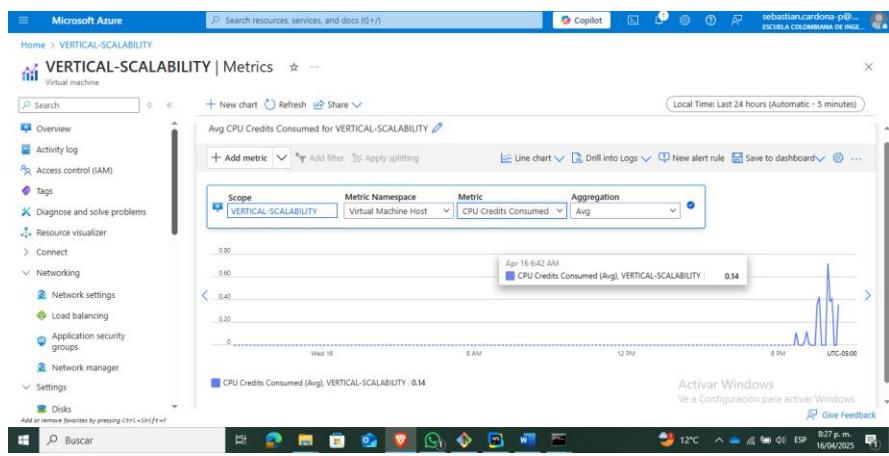
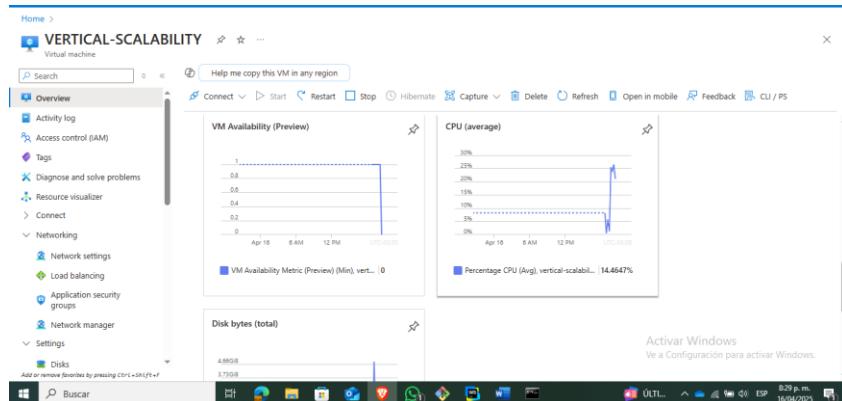


| VM Size | Type            | vCPUs | RAM (GB) | Data disks | Max IOPS | Loc |
|---------|-----------------|-------|----------|------------|----------|-----|
| B1ms    | General purpose | 1     | 0.5      | 2          | 320      | 4   |
| B2ms    | General purpose | 2     | 8        | 4          | 1920     | 1   |
| B3ms    | General purpose | 3     | 12       | 6          | 2880     | 1   |
| B4ms    | General purpose | 4     | 16       | 8          | 2880     | 3   |

Cambiamos el tamaño total de la máquina, haciendo un escalado vertical

11. Una vez el cambio se vea reflejado, repita el paso 7, 8 y 9.

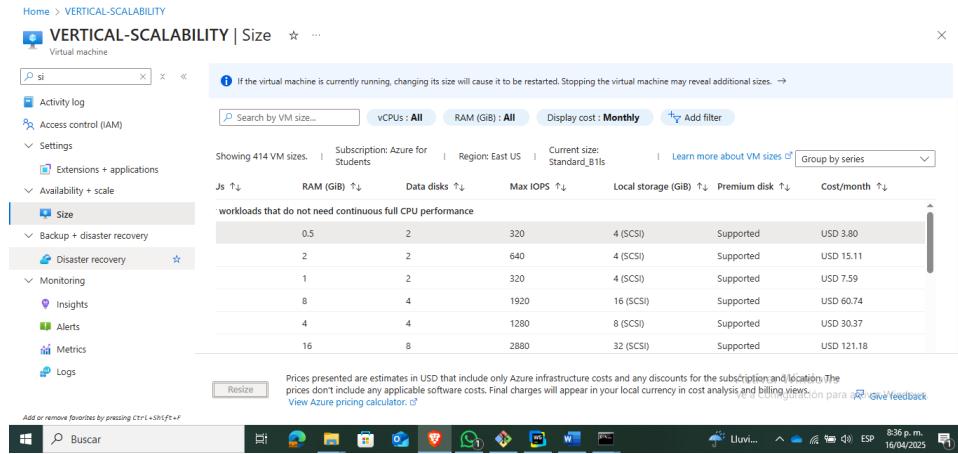
Realizando los mismos pasos, podemos evidenciar como el consumo de la CPU disminuye, ya que se le dio mejores características a esta misma máquina



12. evalúe el escenario de calidad asociado al requerimiento no funcional de escalabilidad y concluya si usando este modelo de escalabilidad logramos cumplirlo.

Así es, pues pese a que se sube el precio de la máquina virtual en Azure, el consumo de la CPU disminuye considerablemente

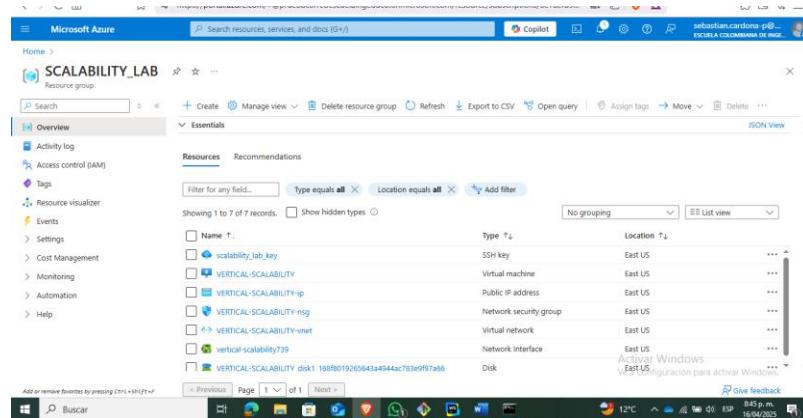
### 13. Vuelva a dejar la VM en el tamaño inicial para evitar cobros adicionales.



| vCPUs | RAM (GiB) | Data disks | Max IOPS  | Local storage (GiB) | Premium disk | Cost/month |
|-------|-----------|------------|-----------|---------------------|--------------|------------|
| All   | All       | All        | All       | All                 | All          | All        |
| 0.5   | 2         | 320        | 4 (SCSI)  | Supported           | USD 3.80     |            |
| 2     | 2         | 640        | 4 (SCSI)  | Supported           | USD 15.11    |            |
| 1     | 2         | 320        | 4 (SCSI)  | Supported           | USD 7.59     |            |
| 8     | 4         | 1920       | 16 (SCSI) | Supported           | USD 60.74    |            |
| 4     | 4         | 1280       | 8 (SCSI)  | Supported           | USD 30.37    |            |
| 16    | 8         | 2880       | 32 (SCSI) | Supported           | USD 121.18   |            |

### Preguntas parte-1:

#### 1. ¿Cuántos y cuáles recursos crea Azure junto con la VM?



- Una máquina virtual VERTICAL-SCALABILITY
- Una dirección IP pública VERTICAL-SCALABILITY-ip
- Un grupo de seguridad VERTICAL-SCALABILITY-nsg

- Una red virtual VERTICAL-SCALABILITY-vnet
- Una interfaz de red VERTICAL-SCALABILITY739
- Un disco VERTICAL-SCALABILITY-disk
- Claves SSH CCALABILITY\_LAB\_key

Fueron 7 servicios que se abrieron

2. ¿Brevemente describa para qué sirve cada recurso?

**Máquina virtual:** Las Azure Virtual Machines son instancias escalables bajo demanda que actúan como servidores, proporcionando hardware virtual (CPU, memoria, almacenamiento) de forma aislada del sistema subyacente.

**Dirección IP pública:** Facilitan la comunicación entrante de Internet a recursos de Azure y la conexión de estos con servicios públicos y la red.

**Grupo de seguridad:** Permiten configurar reglas de seguridad en la red agrupando máquinas virtuales y sus políticas.

**Red virtual:** Azure Virtual Network crea redes privadas seguras que conectan recursos de Azure entre sí, con Internet y con redes locales, ofreciendo escalabilidad y aislamiento.

**Interfaz de red:** Una NIC permite que las máquinas virtuales de Azure se comuniquen con redes internas, externas y locales.

**Disco:** Azure Managed Disks ofrece almacenamiento de bloques duradero para máquinas virtuales con opciones como Ultra Disk, SSD Premium, SSD estándar y HDD estándar, facturados según tamaño y uso.

3. ¿Al cerrar la conexión ssh con la VM, por qué se cae la aplicación que ejecutamos con el comando npm FibonacciApp.js? ¿Por qué debemos crear un *Inbound port rule* antes de acceder al servicio?

Al ejecutar npm FibonacciApp.js en SSH, el proceso se asocia al terminal activo y se termina al cerrar la sesión. Para mantenerlo activo, se usa forever para ejecutarlo en segundo plano.

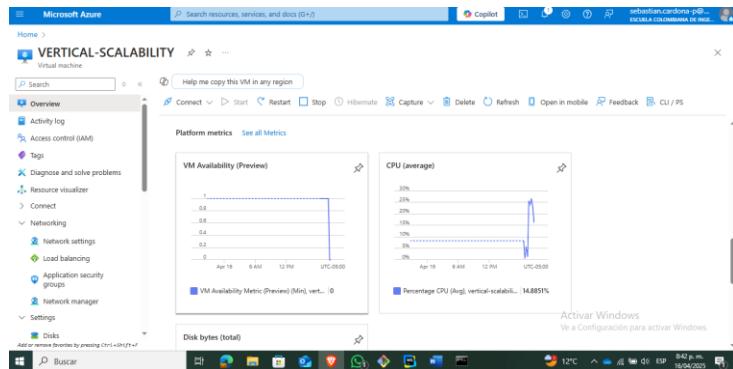
También se debe crear una Inbound port rule para que se permita el tráfico en el puerto 3000 que es el del programa, en caso contrario de no realizarlo, al momento de consumir el endpoint, no funcionará

4. Adjunte tabla de tiempos e interprete por qué la función tarda tanto tiempo.

La función presenta un tiempo de ejecución creciente con cada iteración debido a la acumulación de operaciones. Esto hace que el desempeño dependa directamente del número de iteraciones, pudiendo aumentar de forma proporcional o exponencial según la complejidad interna. Optimizar su diseño es esencial para manejar volúmenes altos de iteraciones eficientemente.

Una idea de optimización es usar la programación con memoria dinámica para guardar los resultados grandes y así demorar menos en futuras consultas

5. Adjunte imagen del consumo de CPU de la VM e interprete por qué la función consume esa cantidad de CPU.



Se consume tal cantidad de CPU, pues el procedimiento de encontrar un número grande es bastante laburioso, por lo que se destinan grandes cantidades de recursos a dicho procedimiento

6. Adjunte la imagen del resumen de la ejecución de Postman. Interprete:

- Tiempos de ejecución de cada petición.
- Si hubo fallos documentelos y explique.

Sin escalado vertical:

Con escalado vertical:

Los tiempos en promedio fueron altos con 23 segundos aproximadamente, pero en la versión sin escalar se presentó un error en una petición:

El error se pudo deber a que la conexión entre cliente y servidor se interrumpe inesperadamente, generalmente por cierre abrupto, tiempo de espera agotado, sobrecarga del servidor o problemas de red.

7. ¿Cuál es la diferencia entre los tamaños B2ms y B1ls (no solo busque especificaciones de infraestructura)?

El tamaño B1ls es ideal para tareas ligeras y económicas, como servidores de desarrollo o pruebas básicas, ya que tiene recursos limitados (1 vCPU, 0.5 GiB de memoria) y un costo bajo (\$3.80/mes). Por otro lado, el tamaño B2ms ofrece mayor capacidad (2 vCPU, 8 GiB de memoria) a un costo mayor (\$58.40/mes), siendo más

adecuado para aplicaciones web pequeñas, bases de datos ligeras o entornos que requieren más flexibilidad y rendimiento.

8. ¿Aumentar el tamaño de la VM es una buena solución en este escenario?, ¿Qué pasa con la FibonacciApp cuando cambiamos el tamaño de la VM?

El análisis previo muestra que el escalamiento vertical es una solución efectiva: mientras que el tamaño **B1ls** alcanzaba casi su capacidad máxima bajo múltiples solicitudes, al escalar al tamaño **B2ms**, el uso de CPU se mantuvo por debajo del 50%. Sin embargo, también es crucial optimizar el algoritmo utilizado para mejorar la eficiencia general de la aplicación puesto que los tiempos de respuesta antes solicitudes altas sigue siendo igual.

9. ¿Qué pasa con la infraestructura cuando cambia el tamaño de la VM? ¿Qué efectos negativos implica?

Aumenta el precio, pero también los recursos virtuales de la máquina

10. ¿Hubo mejora en el consumo de CPU o en los tiempos de respuesta? Si/No ¿Por qué?

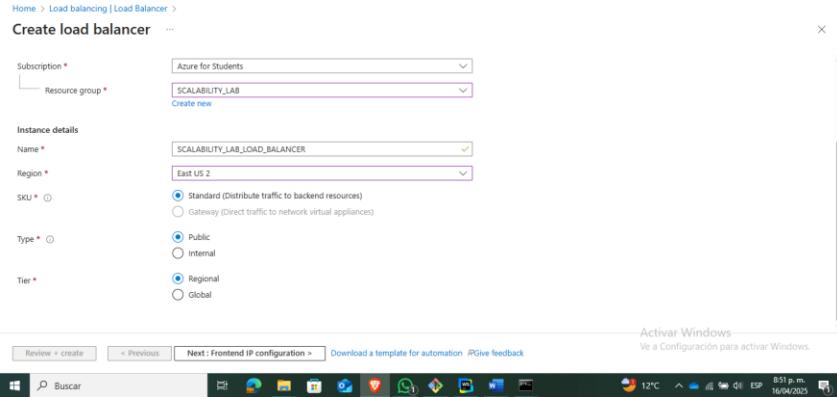
Si, hubo mejoras en el consumo pues los recursos eran mejores, pero en tiempos sigue igual

11. Aumente la cantidad de ejecuciones paralelas del comando de postman a 4. ¿El comportamiento del sistema es porcentualmente mejor?

No mejoran los tiempos

## Parte 2 – Escalabilidad Horizontal

1. El Balanceador de Carga es un recurso fundamental para habilitar la escalabilidad horizontal de nuestro sistema, por eso en este paso cree un balanceador de carga dentro de Azure tal cual como se muestra en la imagen adjunta.

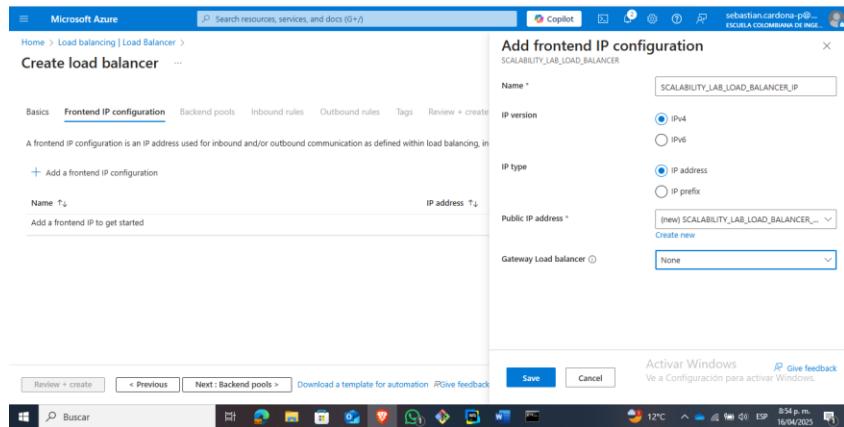


The screenshot shows the 'Create load balancer' wizard in the Azure portal. The 'Instance details' section is filled out as follows:

- Subscription:** Azure for Students
- Resource group:** SCALABILITY\_LAB
- Name:** SCALABILITY\_LAB\_LOAD\_BALANCER
- Region:** East US 2
- SKU:** Standard (Distribute traffic to backend resources)
- Type:** Public
- Tier:** Regional

At the bottom, there are buttons for 'Review + create', '< Previous', 'Next : Frontend IP configuration >', 'Download a template for automation', and 'Give feedback'.

Seleccionamos next:

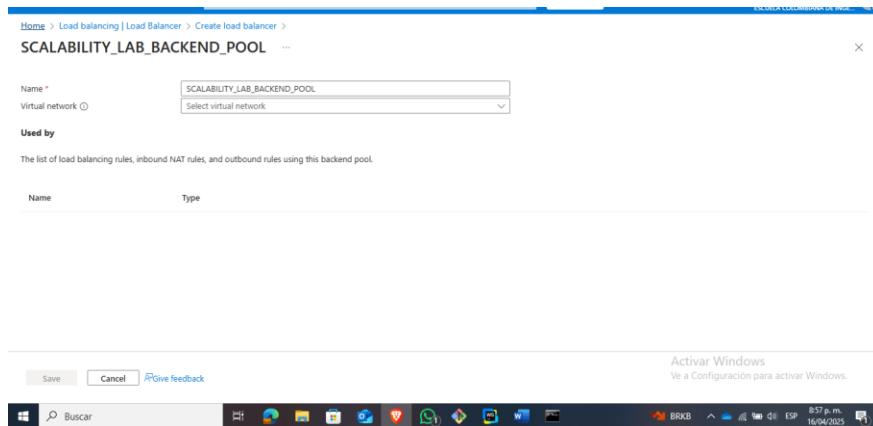


The screenshot shows the 'Add frontend IP configuration' wizard in the Azure portal. The 'Frontend IP configuration' tab is selected. The configuration is as follows:

- Name:** SCALABILITY\_LAB\_LOAD\_BALANCER\_IP
- IP version:** IPv4
- IP type:** IP address
- Public IP address:** (new) SCALABILITY\_LAB\_LOAD\_BALANCER\_IP
- Gateway Load balancer:** None

At the bottom, there are buttons for 'Review + create', '< Previous', 'Next : Backend pools >', 'Download a template for automation', and 'Give feedback'.

En Backend Pool

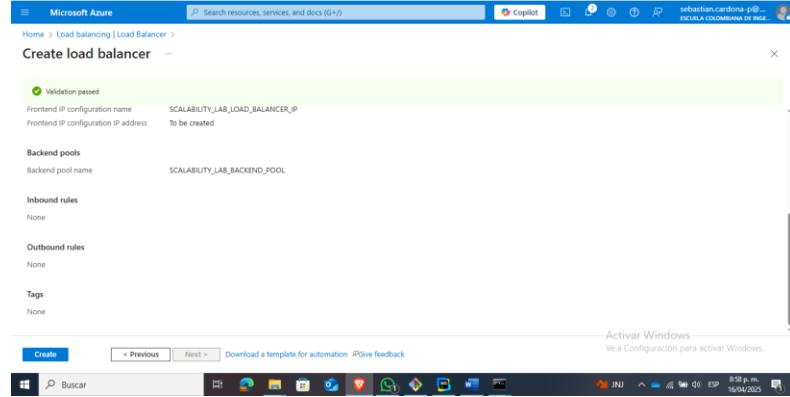


The screenshot shows the 'SCALABILITY\_LAB\_BACKEND\_POOL' configuration page in the Azure portal. The 'Used by' section indicates it's used by the load balancer. The configuration is as follows:

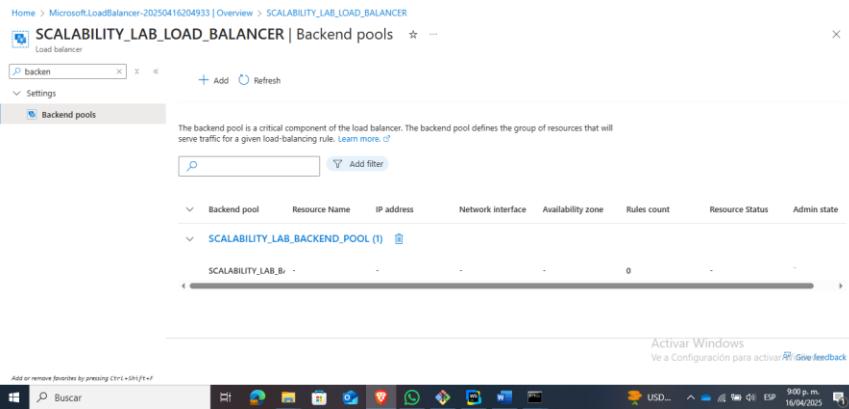
- Name:** SCALABILITY\_LAB\_BACKEND\_POOL
- Virtual network:** Select virtual network

At the bottom, there are buttons for 'Save', 'Cancel', and 'Give feedback'.

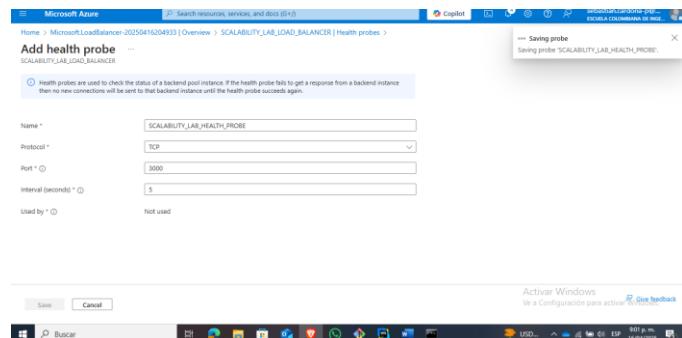
Damos en review and create y creamos el Load Balancer:



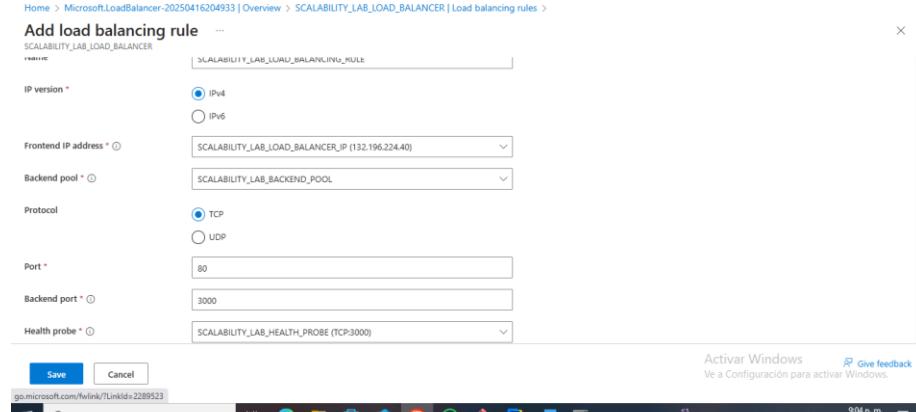
2. A continuación cree un *Backend Pool*, guíese con la siguiente imagen.



3. A continuación cree un *Health Probe*, guíese con la siguiente imagen.



4. A continuación cree un *Load Balancing Rule*, guíese con la siguiente imagen.



Home > Microsoft.LoadBalancer-20250416204933 | Overview > SCALABILITY\_LAB\_LOAD\_BALANCER | Load balancing rules > Add load balancing rule

SCALABILITY\_LAB\_LOAD\_BALANCER

IP version \*

Frontend IP address \*

Backend pool \*

Protocol

Port \*

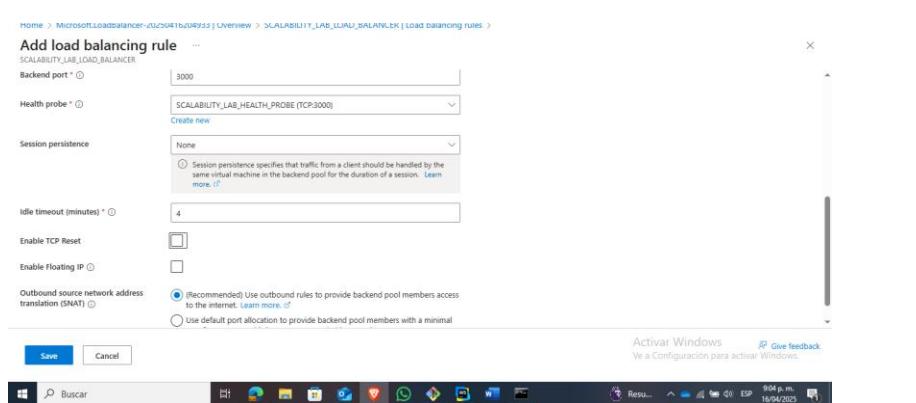
Backend port \*

Health probe \*

Save Cancel

Activar Windows [Give feedback](#)  
Ve a Configuración para activar Windows.

go.microsoft.com/fwlink/?LinkId=2289523



Home > Microsoft.LoadBalancer-20250416204933 | Overview > SCALABILITY\_LAB\_LOAD\_BALANCER | Load balancing rules > Add load balancing rule

SCALABILITY\_LAB\_LOAD\_BALANCER

Backend port \*

Health probe \*

Session persistence

Idle timeout (minutes) \*

Enable TCP Reset

Enable Floating IP

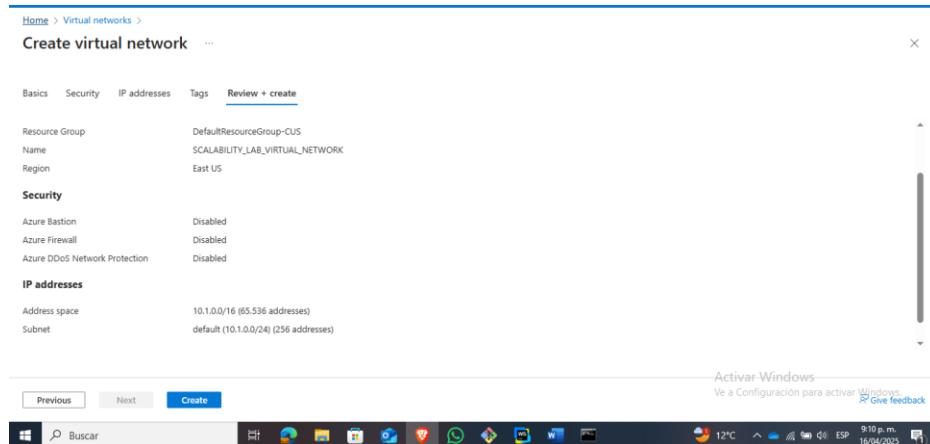
Outbound source network address translation (SNAT) \*

Save Cancel

Activar Windows [Give feedback](#)  
Ve a Configuración para activar Windows.

9:04 p.m.  
16/04/2025

## 5. Cree una *Virtual Network* dentro del grupo de recursos, guíese con la siguiente imagen.



Home > Virtual networks >

Create virtual network ...

Basics Security IP addresses Tags Review + create

Resource Group DefaultResourceGroup-CUS

Name SCALABILITY\_LAB\_VIRTUAL\_NETWORK

Region East US

Security

Azure Bastion Disabled

Azure Firewall Disabled

Azure DDoS Network Protection Disabled

IP addresses

Address space 10.1.0.0/16 (65,536 addresses)

Subnet default (10.1.0.0/24) (256 addresses)

Previous Next Create

Activar Windows [Give feedback](#)  
Ve a Configuración para activar Windows.

12°C 9:10 p.m.  
16/04/2025

## Crear las máquinas virtuales (Nodos)

1. En la configuración básica de la VM guíese por la siguiente imagen. Es importante que se fije en la "Availability Zone", donde la VM1 será 1, la VM2 será 2 y la VM3 será 3.

**VM, VM2 y VM3 son similares**

Home > Compute infrastructure | Virtual machines >

Create a virtual machine ...

Help me create a low cost VM Help me create a VM optimized for high availability Help me choose the right VM size for my workload

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* Azure for Students

Resource group \* SCALABILITY\_LAB Create new

Instance details

Virtual machine name \* VM1

Region \* (US) West US 2

Availability options Availability zone

Zone options Self-selected zone Choose up to 3 availability zones, one VM per zone

Azure-selected zone (Preview)

Activar Windows Ve a Configuración para activar [Give feedback](#)

< Previous Next : Disks > Review + create

Buscar Help me create a low cost VM Help me create a VM optimized for high availability Help me choose the right VM size for my workload

Availability zone \* Zone 1 You can now select multiple zones. Selecting multiple zones will create one VM per zone [Learn more](#)

Security type Trusted launch virtual machines Configure security features Trusted launch virtual machine is required when using 1P Gallery images.

Image \* Ubuntu Server 22.04 LTS - x64 Gen2 See all images | Configure VM generation

VM architecture Arm64 x64

Run with Azure Spot discount

Size \* Standard\_B1s - 1 vcpu, 0.5 GiB memory (USD 3.80/month) See all sizes

Activar Windows Ve a Configuración para activar [Give feedback](#)

< Previous Next : Disks > Review + create https://portal.azure.com/#

virtual machine.

Username \* vm1

SSH public key source Use existing key stored in Azure Ed25519 and RSA-3072 formats are supported for the selected VM image. Ed25519 provides a fixed security level of no more than 128 bits for 256-bit key, while RSA could offer better security with keys longer than 3072 bits.

Stored Keys scalability\_lab\_key

Inbound port rules Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

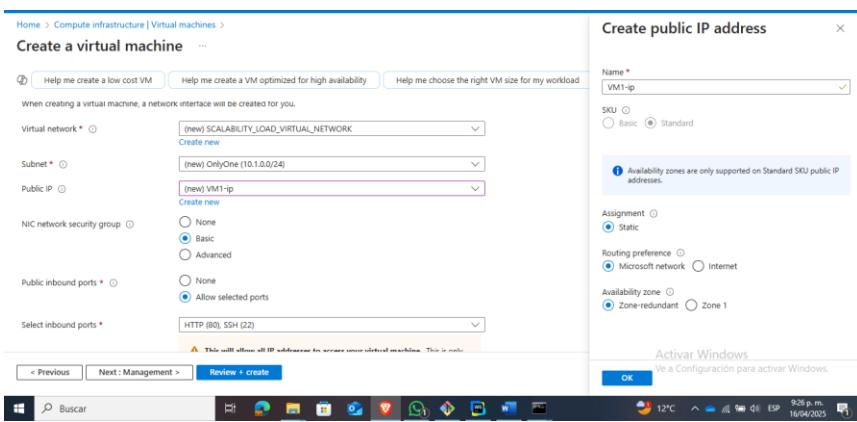
Public inbound ports \* None Allow selected ports

Select inbound ports \* HTTP (80), SSH (22)

Activar Windows Ve a Configuración para activar [Give feedback](#)

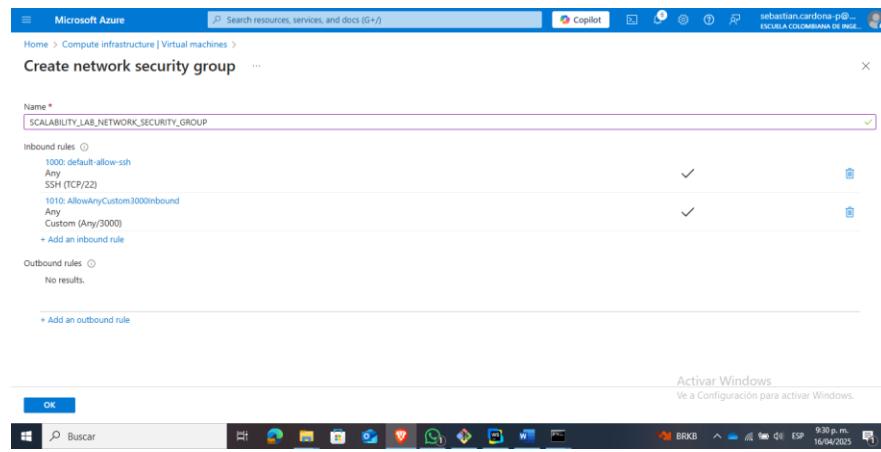
< Previous Next : Disks > Review + create https://portal.azure.com/#

2. En la configuración de networking, verifique que se ha seleccionado la *Virtual Network* y la *Subnet* creadas anteriormente. Adicionalmente asigne una IP pública y no olvide habilitar la redundancia de zona.
- VM, VM2 y VM3 son similares



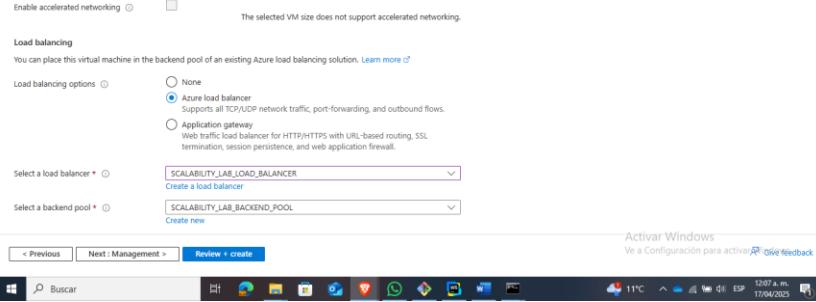
The screenshot shows the Microsoft Azure portal interface. On the left, the 'Create a virtual machine' wizard is open, showing configuration steps for a new VM. On the right, a separate 'Create public IP address' dialog box is overlaid. The IP address being created is named 'VM1-ip'. The dialog includes fields for SKU (set to Standard), Assignment (set to Static), Routing preference (set to Microsoft network), and Availability zone (set to Zone-redundant). A note indicates that availability zones are only supported on Standard SKU public IP addresses. At the bottom of the dialog, there is an 'Activar Windows' section with a link to configuration.

3. Para el Network Security Group seleccione "avanzado" y realice la siguiente configuración. No olvide crear un *Inbound Rule*, en el cual habilite el tráfico por el puerto 3000. Cuando cree la VM2 y la VM3, no necesita volver a crear el *Network Security Group*, sino que puede seleccionar el anteriormente creado.



The screenshot shows the Microsoft Azure portal interface. The 'Create network security group' wizard is open. The 'Name' field is filled with 'SCALABILITY\_LAB\_NETWORK\_SECURITY\_GROUP'. Under 'Inbound rules', there is a list with two entries: 'Any SSH (TCP/22)' and '1010 AllowAnyCustom3000 inbound Any Custom (Any/3000)'. Both rules have a checkmark next to them. Below this, under 'Outbound rules', it says 'No results.' and has a '+ Add an outbound rule' button. At the bottom of the wizard, there is an 'Activar Windows' section with a link to configuration.

4. Ahora asignaremos esta VM a nuestro balanceador de carga, para ello siga la configuración de la siguiente imagen.



- Finalmente debemos instalar la aplicación de Fibonacci en la VM. para ello puede ejecutar el conjunto de los siguientes comandos, cambiando el nombre de la VM por el correcto  
VM1, VM2 y VM3

Nos conectamos a la maquina con el siguiente enlace, teniendo en cuenta que usamos una llave privada de Azure para el ssh, el proceso con las demás maquinas es similar:

`ssh -i ./VM1_key.pem vm1@4.155.154.118`

Después clonamos el repositorio:

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

[ubuntu]:~$ git clone https://github.com/SebastianCardona-P/ARSN-Lab.git
Cloning into 'ARSN-Lab'...
remote: Enumerating objects: 34, done.
remote: Counting objects: 100% (34/34), done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 34 (delta 3), reused 34 (delta 3), pack-reused 0 (from 0)
Receiving objects: 100% (34/34), 3.55 MiB | 24.54 MiB/s, done.
Resolving deltas: 100% (3/3), done.
[ubuntu]:~$
```

Realice este proceso para las 3 VMs; por ahora lo haremos a mano una por una, sin embargo es importante que usted sepa que existen herramientas para automatizar este proceso, entre ellas encontramos Azure Resource Manager, ODisk Images, Terraform con Vagrant y Paket, Puppet, Ansible entre otras.

Activar Windows  
Ve a Configuración para activar Windows.

Instalamos node y npm

```
ca vm1@VM: ~
Cloning into 'ARSW-Lab8'...
remote: Enumerating objects: 34, done.
remote: Counting objects: 100% (34/34), done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 34 (delta 3), reused 34 (delta 3), pack-reused 0 (from 0)
Receiving objects: 100% (34/34), 3.55 MiB | 24.54 MiB/s, done.
Resolving deltas: 100% (3/3), done.
vm1@VM1:~$ curl -o https://raw.githubusercontent.com/creationix/nvm/v0.34.0/install.sh | bash
  % Total    % Received % Xferd  Average Speed   Time     Time  Current
          Dload  Upload Total Spent   Left  Speed
100 13226  100 13226    0     0  178k      0 --:--:--:--:--:-- 179k
->  Downloading nvm from git to '/home/vm1/.nvm'
->  Cloning into '/home/vm1/.nvm'...
remote: Enumerating objects: 278, done.
remote: Counting objects: 100% (278/278), done.
remote: Compressing objects: 100% (245/245), done.
remote: Total 278 (delta 32), reused 103 (delta 20), pack-reused 0 (from 0)
Receiving objects: 100% (278/278), 142.50 KiB | 4.91 MiB/s, done.
Resolving deltas: 100% (32/32), done.
->  Compressing and cleaning up git repository

-> Appending nvm source string to /home/vm1/.bashrc
-> Appending bash_completion source string to /home/vm1/.bashrc
-> Close and reopen your terminal to start using nvm or run the following to use it now:

export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion" # This loads nvm bash_completion
vm1@VM1:~$ source /home/vm1/.bashrc
vm1@VM1:~$ nvm install node
Active Windows
Ve a Configuración para activar Windows.
```

man que se realizaron en la parte 1 y haga un informe comparativo donde

11°C 11:32 p. m. 16/04/2025

Iniciamos el Proyecto como lo hicimos anteriormente

```
ca vm1@VM: ~/ARSW-Lab8/FibonacciApp
v23.11.0
vm1@VM1:~$ npm --version
10.9.2
vm1@VM1:~$ cd ARSW-Lab8/
vm1@VM1:~/ARSW-Lab8$ ls
FibonacciApp README.md images
vm1@VM1:~/ARSW-Lab8$ cd FibonacciApp/
vm1@VM1:~/ARSW-Lab8/FibonacciApp$ npm install
npm warn old lockfile
npm warn old lockfile The package-lock.json file was created with an old version of npm,
npm warn old lockfile so supplemental metadata must be fetched from the registry.
npm warn old lockfile
npm warn old lockfile This is a one-time fix-up, please be patient...
npm warn old lockfile

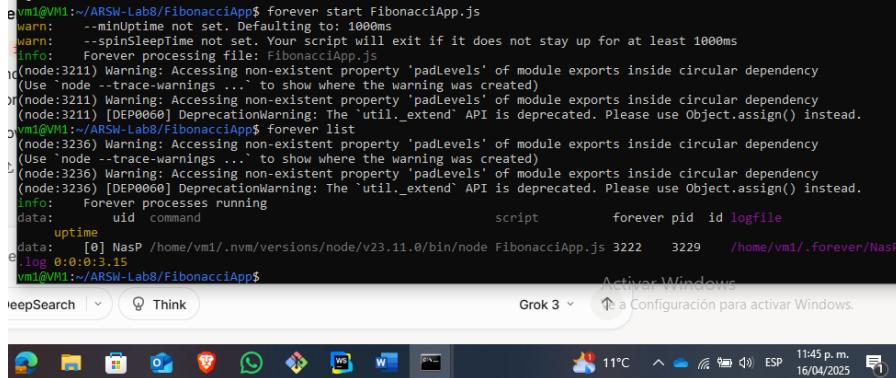
added 51 packages, and audited 52 packages in 5s
7 vulnerabilities (3 low, 4 high)

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
npm notice
npm notice New major version of npm available! 10.9.2 → 11.3.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.3.0
npm notice To update run: npm install -g npm@11.3.0
npm notice
vm1@VM1:~/ARSW-Lab8/FibonacciApp$ npm install forever -g
Active Windows
Ve a Configuración para activar Windows.
```

man que se realizaron en la parte 1 y haga un informe comparativo donde

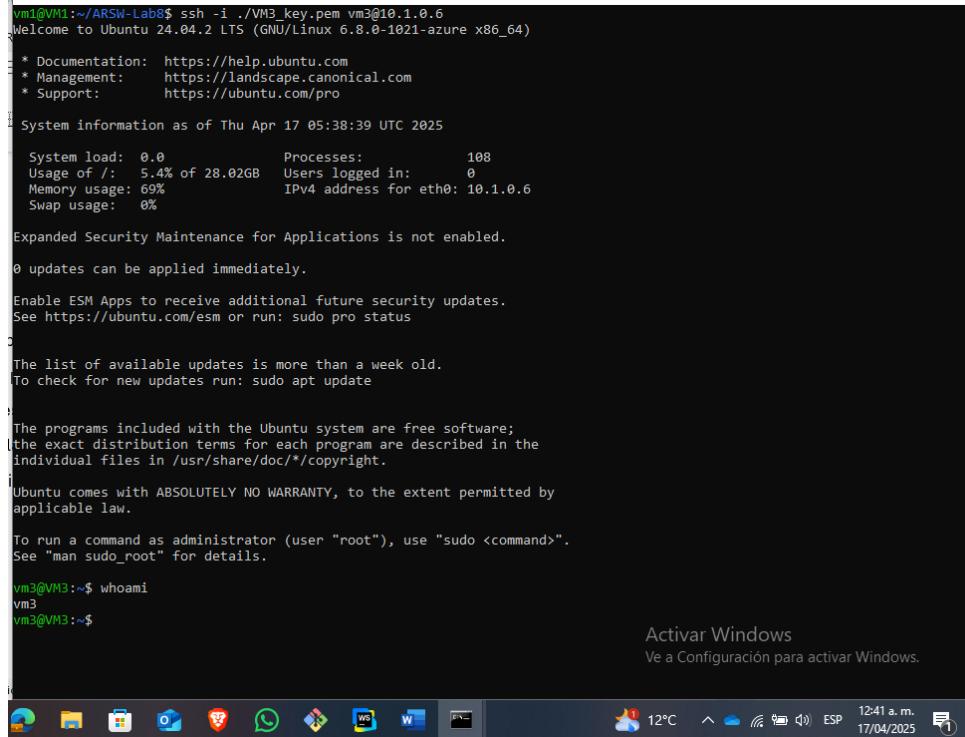
11°C 11:35 p. m. 16/04/2025



```

vm1@VM1:~/ARSW-Lab8/FibonacciApp$ forever start FibonacciApp.js
warn: --minUptime not set. Defaulting to: 1000ms
warn: --spinSleepTime not set. Your script will exit if it does not stay up for at least 1000ms
info: Forever processing file: FibonacciApp.js
(node:3211) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
(node:3211) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
(node:3211) [DEP0066] DeprecationWarning: The 'util._extend' API is deprecated. Please use Object.assign() instead.
vm1@VM1:~/ARSW-Lab8/FibonacciApp$ forever list
(node:3211) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
(node:3236) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
(node:3236) [DEP0066] DeprecationWarning: The 'util._extend' API is deprecated. Please use Object.assign() instead.
info: Forever processes running
data:   uid  command                                script          forever pid  id logfile
data:   [0] NasP /home/vm1/.nvm/versions/node/v23.11.0/bin/node FibonacciApp.js 3222    3229  /home/vm1/.forever/NasP
.log 0:0:0:3.15
vm1@VM1:~/ARSW-Lab8/FibonacciApp$
```

La aplicación ya está corriendo. Debido a que la condición de cuenta estudiantil solo permite tener tres ips públicas (Load balancer , VM1, VM2), no podíamos tener acceso a la VM3 por medio de ssh sin una ip pública, por lo tanto aprovechamos que las maquinas VM1, VM2 Y VM3 tienen conexión entre sí, para entrar a VM1 por medio de la IP pública y luego desde VM1, entrar a VM3



```

vm1@VM1:~/ARSW-Lab8$ ssh -i ./VM3_key.pem vm3@10.1.0.6
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1021-azure x86_64)

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

System information as of Thu Apr 17 05:38:39 UTC 2025

 System load:  0.0           Processes:      108
 Usage of /:  5.4% of 28.02GB  Users logged in:     0
 Memory usage: 69%            IPv4 address for eth0: 10.1.0.6
 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

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

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

Vm3@VM3:~$ whoami
Vm3
Vm3@VM3:~$
```

Activar Windows  
Ve a Configuración para activar Windows.

Ahora si están las tres maquinas en el Load Balancer



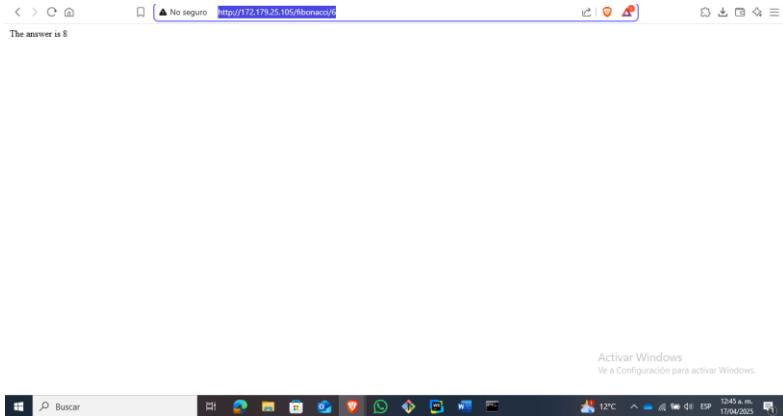
| Nombre de la VM                  | Nombre de la máquina | IP pública | Reservado | Prioridad | Tiempo límite | Estado  | Reservado para | Última ejecución |
|----------------------------------|----------------------|------------|-----------|-----------|---------------|---------|----------------|------------------|
| SCALABILITY_LAB_BACKEND_POOL (3) |                      |            |           |           |               |         |                |                  |
| SCALABILITY_LAB_BACKER1          | VM1                  | 10.1.0.4   | vm1478_z1 | 1         | 1             | Running | None           |                  |
| SCALABILITY_LAB_BACKER1          | VM2                  | 10.1.0.5   | vm2414_z1 | 1         | 1             | Running | None           |                  |
| SCALABILITY_LAB_BACKER1          | VM3                  | 10.1.0.6   | vm3841_z1 | 1         | 1             | Running | None           |                  |

Activar Windows  
Ve a Configuración para activar Windows [Give Feedback](#)

## Probar el resultado final de nuestra infraestructura

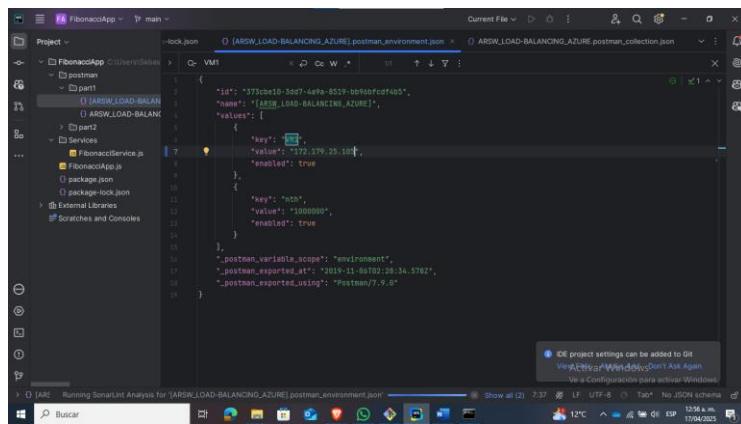
1. Porsupuesto el endpoint de acceso a nuestro sistema será la IP pública del balanceador de carga, primero verifiquemos que los servicios básicos están funcionando, consuma los siguientes recursos:

<http://172.179.25.105/fibonacci/6>



2. Realice las pruebas de carga con newman que se realizaron en la parte 1 y haga un informe comparativo donde contrasta: tiempos de respuesta, cantidad de peticiones respondidas con éxito, costos de las 2 infraestrucruras, es decir, la que desarrollamos con balanceo de carga horizontal y la que se hizo con una maquina virtual escalada.

Cambiamos nuevamente la ip para determinar el comportamiento

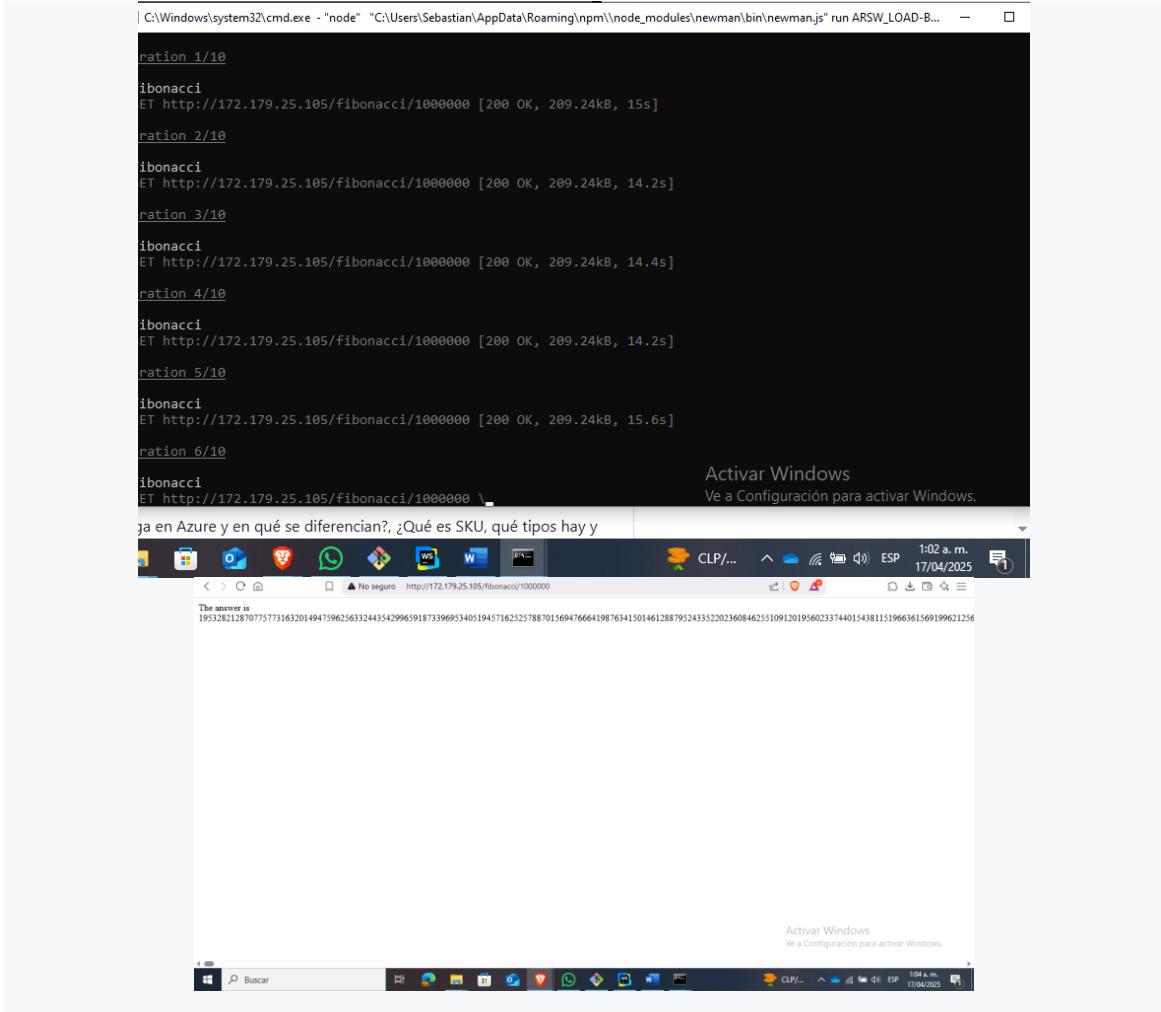


Ejecutamos el comando:

```
newman run ARSW_LOAD-BALANCING_AZURE.postman_collection.json -e
[ARSW_LOAD-BALANCING_AZURE].postman_environment.json -n 10
```

3. Agregue una 4 máquina virtual y realice las pruebas de newman, pero esta vez no lance 2 peticiones en paralelo, sino que incrementelo a 4. Haga un informe donde presente el comportamiento de la CPU de las 4 VM y explique porque la tasa de éxito de las peticiones aumento con este estilo de escalabilidad.

```
newman run ARSW_LOAD-BALANCING_AZURE.postman_collection.json -e [ARSW_LOAD-BALANCING_AZURE].postman_environment.json -n 10 &
newman run ARSW_LOAD-BALANCING_AZURE.postman_collection.json -e [ARSW_LOAD-BALANCING_AZURE].postman_environment.json -n 10 &
newman run ARSW_LOAD-BALANCING_AZURE.postman_collection.json -e [ARSW_LOAD-BALANCING_AZURE].postman_environment.json -n 10 &
newman run ARSW_LOAD-BALANCING_AZURE.postman_collection.json -e [ARSW_LOAD-BALANCING_AZURE].postman_environment.json -n 10
```



The image shows a Windows desktop environment. At the top, there is a taskbar with various pinned icons. A command prompt window is open, showing a series of API requests to a local host (172.179.25.105) for a Fibonacci sequence endpoint. The requests are labeled 'ration 1/10' through 'ration 6/10'. Each request shows a successful response (HTTP 200 OK) with a 209.24kB file size and a duration between 14.2s and 15.6s. Below the command prompt, a browser window is open to the same URL, displaying a long string of digits as the response content. A tooltip from the operating system suggests activating Windows.

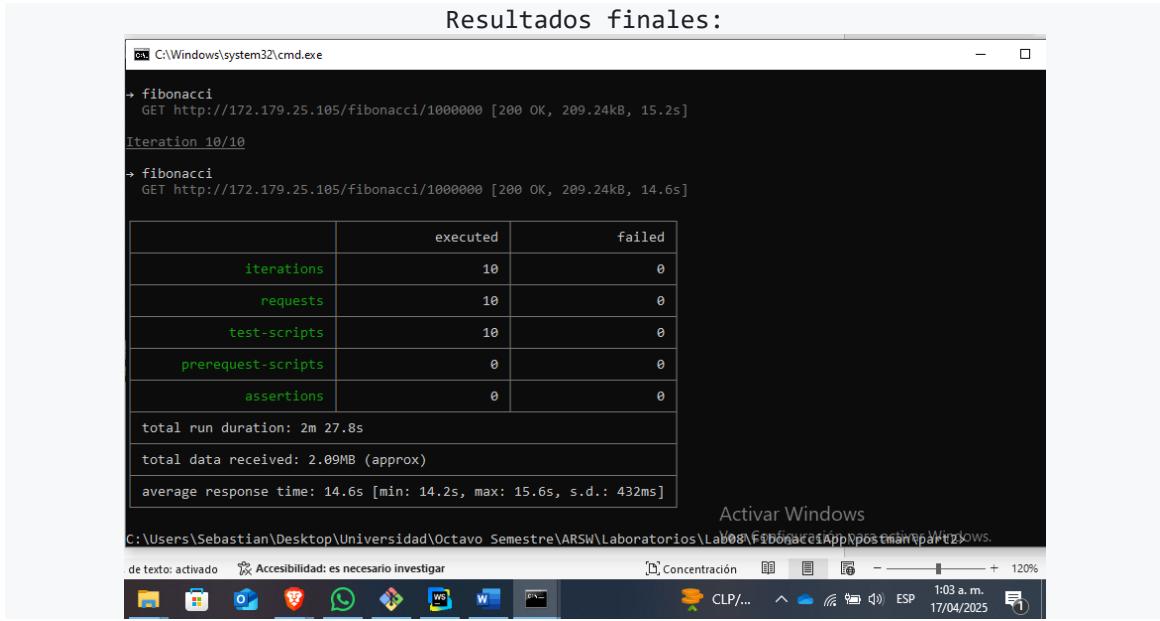
```
| C:\Windows\system32\cmd.exe - "node" "C:\Users\Sebastian\AppData\Roaming\npm\node_modules\newman\bin\newman.js" run ARSW_LOAD-B...
ration 1/10
ibonacci
ET http://172.179.25.105/fibonacci/1000000 [200 OK, 209.24kB, 15s]
ration 2/10
ibonacci
ET http://172.179.25.105/fibonacci/1000000 [200 OK, 209.24kB, 14.2s]
ration 3/10
ibonacci
ET http://172.179.25.105/fibonacci/1000000 [200 OK, 209.24kB, 14.4s]
ration 4/10
ibonacci
ET http://172.179.25.105/fibonacci/1000000 [200 OK, 209.24kB, 14.2s]
ration 5/10
ibonacci
ET http://172.179.25.105/fibonacci/1000000 [200 OK, 209.24kB, 15.6s]
ration 6/10
ibonacci
ET http://172.179.25.105/fibonacci/1000000 \_
```

Activar Windows  
Ve a Configuración para activar Windows.

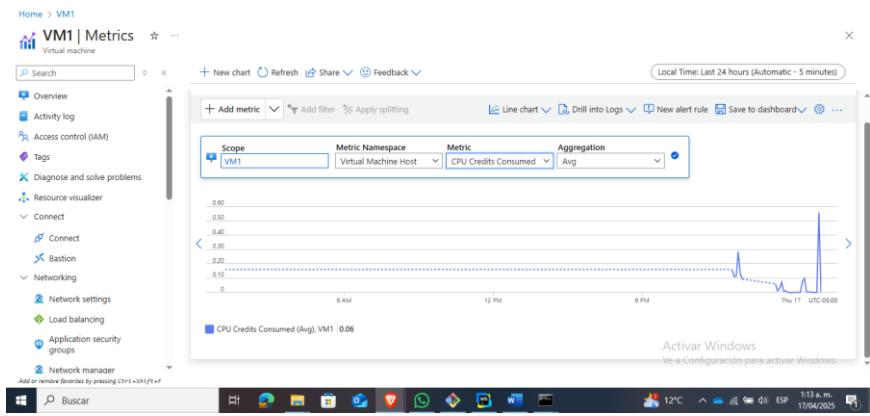
ja en Azure y en qué se diferencian?, ¿Qué es SKU, qué tipos hay

The answer is  
19532821287077577316320149475962563324435429965918733969534051945716252578870156947666419876341501461288795243352202360846255109120195602337440154381151966361569199621256

Activar Windows  
Ve a Configuración para activar Windows.



## Consumo de CPU óptimo



## Preguntas - Parte 2:

- ¿Cuáles son los tipos de balanceadores de carga en Azure y en qué se diferencian?, ¿Qué es SKU, qué tipos hay y en qué se diferencian?, ¿Por qué el balanceador de carga necesita una IP pública?

### 1. Azure Load Balancer:

- **Descripción:** Opera en la capa 4 (modelo OSI), distribuyendo tráfico TCP/UDP entre máquinas virtuales (VMs), conjuntos de escalado de VMs o direcciones IP. Es ideal para aplicaciones que requieren alto rendimiento y baja latencia.
- **Uso:** Equilibra tráfico interno y externo a VMs, soporta escenarios híbridos (conexión desde redes locales), y permite alta disponibilidad mediante distribución en zonas o regiones.
- **Ejemplo:** Distribuir tráfico entre VMs que ejecutan la app de Fibonacci
- **Características:** Usa un algoritmo de hash basado en tuplas (hasta 5 tuplas) para distribuir tráfico, soporta sondeos de estado, y no interactúa con la carga de la aplicación (es transparente para las cargas útiles).

### 2. Azure Application Gateway:

- **Descripción:** Opera en la capa 7, enfocándose en tráfico HTTP/HTTPS. Ofrece enrutamiento basado en rutas, descarga SSL, y capacidades de Web Application Firewall (WAF).
- **Uso:** Ideal para aplicaciones web que necesitan enrutamiento avanzado (por ejemplo, redirigir /images/\* a un grupo de VMs específico).
- **Diferencia con Azure Load Balancer:** Mientras que Azure Load Balancer trabaja a nivel de red (capa 4), Application Gateway gestiona tráfico a nivel de aplicación (capa 7), lo que permite reglas más sofisticadas basadas en contenido.
- **Ejemplo:** Se puede usar para enrutar tráfico a tu app de Fibonacci según patrones de URL, aunque no es necesario para este caso actual.

### 3. Azure Traffic Manager:

- **Descripción:** Es un balanceador basado en DNS que opera a nivel global, distribuyendo tráfico entre regiones de Azure según latencia, disponibilidad o peso.
- **Uso:** Proporciona redundancia geográfica y menor latencia al dirigir a los usuarios al punto de conexión más cercano.
- **Diferencia:** No maneja tráfico directamente (como Azure Load Balancer), sino que resuelve DNS para redirigir clientes. Su supervisión de estados es más básica comparada con Azure Load Balancer.
- **Algoritmos:** Incluye rendimiento (latencia), conmutación por error, round-robin y distribución geográfica.

### 4. Azure Front Door:

- **Descripción:** Es un servicio global de capa 7 que combina balanceo de carga con aceleración de aplicaciones (como una CDN). Soporta enrutamiento basado en rutas, descarga SSL, y alta disponibilidad entre regiones.
- **Uso:** Ideal para aplicaciones web globales que necesitan baja latencia y protección contra fallos regionales.
- **Diferencia:** Combina características de Traffic Manager (balanceo global) y Application Gateway (capa 7), añadiendo optimización de rendimiento mediante redes de entrega de contenido.

### 5. Equilibrador de Carga de Puerta de Enlace (Gateway Load Balancer):

- **Descripción:** Diseñado para escenarios específicos como inspección de tráfico (por ejemplo, con appliances de terceros). Permite encadenar con otros servicios como Azure Load Balancer.
- **Uso:** Útil para integraciones con firewalls o sistemas de seguridad.
- **Diferencia:** Es más especializado y no aplica directamente a tu caso de Fibonacci, que se centra en balanceo básico de tráfico.

**SKU (Stock Keeping Unit)** en Azure Load Balancer se refiere a las diferentes versiones o niveles del servicio, que varían en características, rendimiento y precio. Azure Load Balancer tiene tres tipos de SKU:

**1. Básico (Basic):**

- Soporta hasta 100 instancias en el backend.
- Solo permite sondeos de estado TCP y HTTP (no HTTPS).
- Limitado a VMs en un conjunto de disponibilidad o escalado.
- No soporta zonas de disponibilidad ni emparejamiento de red virtual global.
- No permite conexiones fuera de la zona donde se encuentra.

**2. Estándar (Standard):**

- Soporta hasta 1,000 instancias en el backend.
- Incluye sondeos TCP, HTTP y HTTPS.
- Soporta zonas de disponibilidad para alta resistencia.
- Permite emparejamiento de red virtual global y conexiones entre zonas.
- Ofrece métricas multidimensionales en Azure Monitor y paneles preconfigurados.
- Requiere definir explícitamente conectividad saliente (por ejemplo, con reglas de salida o IP pública estándar).
- Basado en un modelo de seguridad de Confianza Cero (cerrado a conexiones entrantes por defecto, requiere NSG para abrir).

**3. Puerta de Enlace (Gateway):**

- Diseñado para encadenar con otros servicios (por ejemplo, para inspección de tráfico).

- Soporta escenarios avanzados como integración con appliances de terceros.

Un balanceador necesita una IP pública para poder:

Tener Acceso externo a internet y que los usuarios tengan acceso a la app en este caso de Fibonacci

La IP pública actúa como el punto de contacto para los clientes. Azure Load Balancer usa esta IP para recibir solicitudes y luego las redirige a las VMs del backend (como VM1, VM2, VM3) según las reglas de balanceo y sondeos de estado. Esto simplifica el acceso, ya que los clientes no necesitan conocer las IPs privadas de las VMs.

- ¿Cuál es el propósito del *Backend Pool*?

El BackEnd Pool distribuye el tráfico entrante hacia recursos como máquinas virtuales o instancias de contenedor en grupos de escalado. Utiliza algoritmos como **Round Robin** o **Hash de IP** para garantizar una carga balanceada y eficiente entre los recursos disponibles.

- ¿Cuál es el propósito del *Health Probe*?

El **Health Probe** monitorea los recursos en el Backend Pool para garantizar que solo los que están en buen estado reciban tráfico. Envía solicitudes periódicas (HTTP o TCP) para verificar su disponibilidad:

**Recurso saludable:** Responde correctamente y sigue recibiendo tráfico.

**Recurso no saludable:** No responde o muestra errores, y se excluye del tráfico.

- ¿Cuál es el propósito de la *Load Balancing Rule*? ¿Qué tipos de sesión persistente existen, por qué esto es importante y cómo puede afectar la escalabilidad del sistema?.

La **Load Balancing Rule** dirige el tráfico entrante hacia las instancias de backend de manera eficiente y equitativa. Permite configurar el enrutamiento según criterios como rutas URL o puertos, asegurando una distribución uniforme de solicitudes.

- ¿Qué es una *Virtual Network*? ¿Qué es una *Subnet*? ¿Para qué sirven los *address space* y *address range*?

#### **Virtual Network (VNet):**

- Servicio para crear una red virtual aislada en la nube.
- Permite definir un rango de direcciones IP, subredes, reglas de seguridad y puertas de enlace para conectividad con Internet o redes locales.

#### **Subnet:**

- Subdivisión de una VNet para segmentar la red en partes más pequeñas.
- Cada Subnet tiene su propio rango de direcciones IP dentro del espacio de la VNet y reglas de seguridad específicas.

#### **Address Space:**

- Rango de direcciones IP privadas asignado a una VNet.
- Define las direcciones disponibles para la VNet y sus subredes.

#### **Address Range:**

- Rango de direcciones IP asignado a una Subnet dentro de la VNet.
- Los recursos de la Subnet reciben direcciones IP de este rango.

- ¿Qué son las *Availability Zone* y por qué seleccionamos 3 diferentes zonas?. ¿Qué significa que una IP sea *zone-redundant*?

Una Availability Zone es un grupo de centros de datos interconectados dentro de una misma región, ubicados en sitios físicos independientes y aislados de fallas

en otras zonas. Esto garantiza mayor disponibilidad y resiliencia para las aplicaciones alojadas.

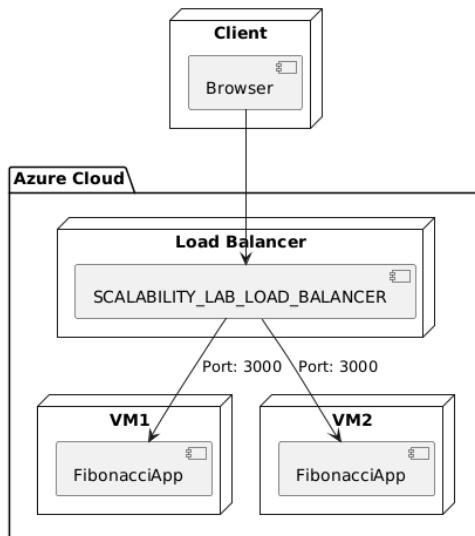
Distribuir los recursos de una aplicación en tres zonas de disponibilidad mejora su disponibilidad y rendimiento, ya que permite operar incluso si una o dos zonas fallan. Además, el tráfico se distribuye de manera más eficiente entre las zonas.

La IP zone-redundant es una dirección IP pública assignable a recursos como máquinas virtuales o balanceadores de carga. Está disponible en todas las zonas de una región, asegurando acceso continuo incluso en caso de fallos en una de las zonas.

- ¿Cuál es el propósito del *Network Security Group*?

El propósito del Network Security Group (NSG) en Azure es proporcionar un nivel adicional de seguridad al permitir o denegar el tráfico de red hacia y desde los recursos de Azure, como máquinas virtuales, subredes y redes virtuales. El NSG actúa como un firewall virtual que filtra el tráfico basándose en reglas que específicas.

- Presente el Diagrama de Despliegue de la solución.



## Conclusiones

A lo largo de este laboratorio, se adquirió un entendimiento profundo sobre la implementación de escalabilidad y alta disponibilidad en Microsoft Azure, utilizando una aplicación de Fibonacci como base para las pruebas. La escalabilidad vertical demostró ser una solución efectiva para reducir el consumo de CPU, pasando de un 70% a 80% en una VM de tamaño B1ls a menos del 50% al escalar a B2ms, lo que permitió cumplir el requerimiento de mantener el uso de CPU por debajo del 70%. Sin embargo, esta estrategia no mejoró los tiempos de respuesta (alrededor de 23-28 segundos para valores grandes), evidenciando la necesidad de optimizar el algoritmo de Fibonacci, posiblemente mediante programación dinámica para almacenar resultados intermedios. Además, el escalamiento vertical incrementa los costos (de \$3.80/mes a \$58.40/mes), lo que puede no ser sostenible a largo plazo.

Por otro lado, la escalabilidad horizontal, implementada mediante un balanceador de carga con tres VMs (posteriormente cuatro), distribuyó eficientemente el tráfico, mejorando la tasa de éxito de las peticiones bajo cargas concurrentes. Con cuatro VMs y cuatro ejecuciones paralelas de Newman, el consumo de CPU se mantuvo óptimo (por debajo del 50% en cada VM), y la tasa de éxito aumentó porque el tráfico se dividió entre más nodos, reduciendo la sobrecarga por máquina. Sin embargo, los tiempos de respuesta mejoraron, pero no significativamente, reafirmando que el cuello de botella está en el diseño del algoritmo más que en la infraestructura, aunque si bajo de 25 segundos en promedio a 16 segundos en promedio.

El uso de Azure Load Balancer (SKU Estándar) facilitó la alta disponibilidad al soportar zonas de disponibilidad, asegurando redundancia frente a fallos regionales. Recursos como el Backend Pool, Health Probe y Load Balancing Rule fueron fundamentales para gestionar la distribución del tráfico y garantizar que solo las VMs saludables recibieran solicitudes. Además, se aprendió a optimizar la conectividad en escenarios con restricciones de IPs públicas, utilizando un jump host para acceder a VMs sin IP pública (como VM3) y transfiriendo archivos mediante SCP. Finalmente, el laboratorio destacó la importancia de balancear costos y rendimiento: mientras la escalabilidad horizontal es más flexible y tolerante a fallos, requiere una configuración más compleja y un manejo cuidadoso de la red (VNet, Subnets, NSG), lo que refuerza la necesidad de planificar cuidadosamente la arquitectura para cumplir con los requerimientos no funcionales de manera eficiente.