

BALANCEO DE CARGAS EN AZURE – LABORATORIO # 10

AUTORES:

YORKS GOMEZ – CESAR VÁSQUEZ

PROFESOR:

JAVIER IVAN TOQUICA

ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

ARQUITECTURAS DE SOTFWARE - GRUPO # 1

INGENIERÍA DE SISTEMAS

BOGOTÁ D.C.

2023

Parte 0 - Entendiendo el escenario de calidad

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%.

Parte 1 - Escalabilidad vertical

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

Crear una máquina virtual ...

Cree una máquina virtual que ejecuta Linux o Windows. Seleccione una imagen de Azure Marketplace o use una imagen personalizada propia. Complete la pestaña Conceptos básicos y, después, use Revisar y crear para aprovisionar una máquina virtual con parámetros predeterminados o bien revise cada una de las pestañas para personalizar la configuración. [Más información](#)

Es posible que esta suscripción no sea apta para implementar máquinas virtuales de ciertos tamaños en determinadas regiones.

Detalles del proyecto

Seleccione la suscripción para administrar recursos implementados y los costes. Use los grupos de recursos como carpetas para organizar y administrar todos los recursos.

Suscripción * ⓘ

Azure for Students

Grupo de recursos * ⓘ

(Nuevo) SCALABILITY_LAB

Crear nuevo

Detalles de instancia

Nombre de máquina virtual * ⓘ

VERTICAL-SCALABILITY

Región * ⓘ

(US) East US 2

Opciones de disponibilidad ⓘ

No se requiere redundancia de la infraestructura

Tipo de seguridad ⓘ

Estándar

Imagen * ⓘ

Ubuntu Minimal 18.04 LTS - x64 Gen1

Tamaño * ⓘ

Standard_B1ls - 1 vcpu, 0.5 GiB de memoria (USD 3.80/mes)

Ver todos los tamaños

Cuenta de administrador

Tipo de autenticación ⓘ

Clave pública SSH

Contraseña

Ahora, Azure genera automáticamente un par de claves SSH y le permite almacenarlo para usarlo en el futuro. Es una forma rápida, sencilla y segura de conectarse a la máquina virtual.

Nombre de usuario * ⓘ

scalability_lab

Origen de clave pública SSH

Usar la clave existente almacenada en Azure

Claves almacenadas

MV-TwoCode_key

Reglas de puerto de entrada

Seleccione los puertos de red de máquina virtual que son accesibles desde la red Internet pública. Puede especificar acceso de red más limitado o granular en la pestaña Red.

Puertos de entrada públicos * ⓘ

Ninguno

Permitir los puertos seleccionados

Seleccionar puertos de entrada *

HTTP (80), SSH (22)

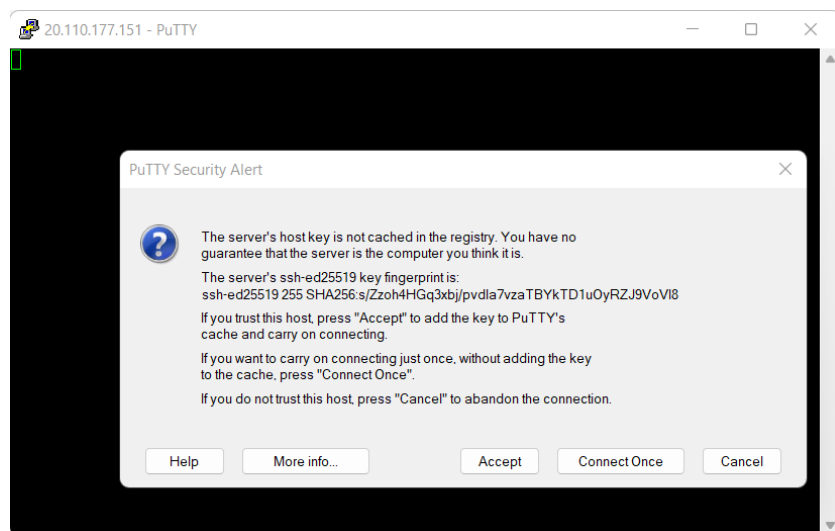
Datos básicos

Suscripción	Azure for Students
Grupo de recursos	(nuevo) SCALABILITY_LAB
Nombre de máquina virtual	VERTICAL-SCALABILITY
Región	East US 2
Opciones de disponibilidad	No se requiere redundancia de la infraestructura
Tipo de seguridad	Estándar
Imagen	Ubuntu Minimal 18.04 LTS - Gen1
Arquitectura de VM	x64
Tamaño	Standard B1ls (1 vcpu, 0.5 GiB de memoria)
Tipo de autenticación	Clave pública SSH
Nombre de usuario	scalability_lab
Nombre de par de claves	MV-TwoCode_key
Puertos de entrada públicos	SSH, HTTP
Azure de acceso puntual	No

Redes

Red virtual	(nuevo) VERTICAL-SCALABILITY-vnet
Subred	(nuevo) default (10.1.0.0/24)
IP pública	(nuevo) VERTICAL-SCALABILITY-ip
Redes aceleradas	Desactivado
¿Quiere colocar esta máquina virtual como subyacente respecto a una solución de equilibrio de carga existente?	No
Eliminar IP pública y NIC cuando se elimine la VM	Deshabilitado

2. Procedemos a conectarnos a la Máquina virtual con Putty.



```
scalability_lab@VERTICAL-SCALABILITY: ~
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

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

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 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:~$
```

3. Instalamos Node.js y npm usando NVM con el siguiente comando:

curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.35.3/install.sh | bash

```
scalability_lab@VERTICAL-SCALABILITY: ~
=> 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" ] && \. "$NVM_DIR/bash_completion" # This loads
nvm bash completion
scalability_lab@VERTICAL-SCALABILITY:~$
scalability_lab@VERTICAL-SCALABILITY:~$ curl -o- https://raw.githubusercontent.com/nvm-sh/n
vm/v0.35.3/install.sh | bash
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 13527  100 13527    0     0  338k      0 --:--:-- --:--:-- --:--:-- 338k
=> nvm is already installed in /home/scalability_lab/.nvm, trying to update the script

=> nvm source string already in /home/scalability_lab/.bashrc
=> bash_completion source string already in /home/scalability_lab/.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" ] && \. "$NVM_DIR/bash_completion" # This loads nvm bash_c
ompletion
scalability_lab@VERTICAL-SCALABILITY:~$
```

En esta sección es importante cerrar la terminal y abrirla de nuevo para que se agreguen las rutas al NVM script de la sesión del Shell actual, o agregarlas manualmente para posteriormente verificar con el comando `nvm --version` la versión y de esta forma comprobar que todo está en orden.

```
scalability_lab@VERTICAL-SCALABILITY:~$ nvm --version
0.35.3
scalability_lab@VERTICAL-SCALABILITY:~$
```

Posterior a esto, ya contando con nvm se procede a instalar node.js con el siguiente comando:

nvm install node

```
scalability_lab@VERTICAL-SCALABILITY: ~
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat May  6 19:18:24 2023 from 200.118.62.147
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

scalability_lab@VERTICAL-SCALABILITY:~$ nvm --version
0.35.3
scalability_lab@VERTICAL-SCALABILITY:~$ nvm install node
Downloading and installing node v20.1.0...
Downloading https://nodejs.org/dist/v20.1.0/node-v20.1.0-linux-x64.tar.xz...
##### 100.0%
Computing checksum with sha256sum
Checksums matched!
node: /lib/x86_64-linux-gnu/libc.so.6: version `GLIBC_2.28' not found (required
by node)
nvm is not compatible with the npm config "prefix" option: currently set to ""
Run `nvm use --delete-prefix v20.1.0` to unset it.
scalability_lab@VERTICAL-SCALABILITY:~$
```

Posteriormente se llevará a cabo la instalación de la última version correspondientes a LTS y la versión del mismo 8.10.0.

```
scalability_lab@VERTICAL-SCALABILITY: ~
nvm is not compatible with the npm config "prefix" option: currently set to ""
Run `nvm use --delete-prefix v20.1.0` to unset it.
Creating default alias: default -> node (-> v20.1.0)
scalability_lab@VERTICAL-SCALABILITY:~$ node --version
-bash: node: command not found
scalability_lab@VERTICAL-SCALABILITY:~$ nvm install --lts
Installing latest LTS version.
Downloading and installing node v18.16.0...
Downloading https://nodejs.org/dist/v18.16.0/node-v18.16.0-linux-x64.tar.xz...
##### 100.0%
Computing checksum with sha256sum
Checksums matched!
node: /lib/x86_64-linux-gnu/libc.so.6: version `GLIBC_2.28' not found (required
by node)
nvm is not compatible with the npm config "prefix" option: currently set to ""
Run `nvm use --delete-prefix v18.16.0` to unset it.
scalability_lab@VERTICAL-SCALABILITY:~$ nvm install 8.10.0
Downloading and installing node v8.10.0...
Downloading https://nodejs.org/dist/v8.10.0/node-v8.10.0-linux-x64.tar.xz...
##### 100.0%
Computing checksum with sha256sum
Checksums matched!
Now using node v8.10.0 (npm v5.6.0)
scalability_lab@VERTICAL-SCALABILITY:~$
```

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:

```
scalability_lab@VERTICAL-SCALABILITY: ~  
scalability_lab@VERTICAL-SCALABILITY:~$ git clone https://github.com/rayo100/lab10-arsw.git  
Cloning into 'lab10-arsw'...  
remote: Enumerating objects: 37, done.  
remote: Counting objects: 100% (37/37), done.  
remote: Compressing objects: 100% (31/31), done.  
remote: Total 37 (delta 4), reused 37 (delta 4), pack-reused 0  
Unpacking objects: 100% (37/37), done.  
scalability_lab@VERTICAL-SCALABILITY:~$
```

Accedemos o nos paramos en la carpeta FibonnaciApp

```
scalability_lab@VERTICAL-SCALABILITY: ~/lab10-arsw/FibonacciApp  
scalability_lab@VERTICAL-SCALABILITY:~/lab10-arsw/FibonacciApp$ npm install  
npm WARN notice [SECURITY] qs has the following vulnerability: 1 high. Go here f  
or more details: https://github.com/advisories?query=qs - Run `npm i npm@latest  
-g` to upgrade your npm version, and then `npm audit` to get more info.  
npm WARN fibonacciapp@1.0.0 No description  
npm WARN fibonacciapp@1.0.0 No repository field.  
  
added 51 packages in 1.451s  
scalability_lab@VERTICAL-SCALABILITY:~/lab10-arsw/FibonacciApp$
```

5. Para ejecutar la aplicación puede usar el comando NPM FibiinacciApp.js, sin embargo una vez pierda la conexión SSH la aplicación dejará de funcionar. Para evitar ese comportamiento usaremos forever. Ejecutamos los siguientes comandos dentro de la VM.

npm install forever -g

```
scalability_lab@VERTICAL-SCALABILITY: ~/lab10-arsw/FibonacciApp
added 51 packages in 1.451s
scalability_lab@VERTICAL-SCALABILITY:~/lab10-arsw/FibonacciApp$ npm install forever -g
npm WARN notice [SECURITY] nconf has the following vulnerability: 1 high. Go here for more details: https://github.com/advisories?query=nconf - Run `npm i npm@latest -g` to upgrade your npm version, and then `npm audit` to get more info.
npm WARN notice [SECURITY] minimist has the following vulnerabilities: 1 critical, 1 moderate. Go here for more details: https://github.com/advisories?query=minimist - Run `npm i npm@latest -g` to upgrade your npm version, and then `npm audit` to get more info.
npm WARN deprecated chokidar@2.1.8: Chokidar 2 does not receive security updates since 2019. Upgrade to chokidar 3 with 15x fewer dependencies
npm WARN notice [SECURITY] glob-parent has the following vulnerability: 1 high. Go here for more details: https://github.com/advisories?query=glob-parent - Run `npm i npm@latest -g` to upgrade your npm version, and then `npm audit` to get more info.
npm WARN deprecated fsevents@1.2.13: The v1 package contains DANGEROUS / INSECURE binaries. Upgrade to safe fsevents v2
npm WARN deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-resolve#deprecated
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-url#deprecated
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
/home/scalability_lab/.nvm/versions/node/v8.10.0/bin/forever -> /home/scalability_lab/.nvm/versions/node/v8.10.0/lib/node_modules/forever/bin/forever
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@^1.2.7 (node_modules/forever/node_modules/chokidar/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})

+ forever@4.0.3
added 284 packages in 12.198s
scalability_lab@VERTICAL-SCALABILITY:~/lab10-arsw/FibonacciApp$
```

forever start FibonacciApp.js

```
scalability_lab@VERTICAL-SCALABILITY: ~/lab10-arsw/FibonacciApp
scalability_lab@VERTICAL-SCALABILITY:~/lab10-arsw/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
scalability_lab@VERTICAL-SCALABILITY:~/lab10-arsw/FibonacciApp$
```

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*.

Inicio > VERTICAL-SCALABILITY

VERTICAL-SCALABILITY | Redes ☆ ...

Máquina virtual

Buscar

Comentarios Adjuntar interfaz de red Desasociar interfaz de red

vertical-scalability883

Configuración de IP ⓘ

ipconfig1 (Principal)

Interfaz de red: vertical-scalability883 Reglas de seguridad vigentes Solucionar problemas

Red virtual/subred: VERTICAL-SCALABILITY-vnet/default IP pública de NIC: 20.110.177.151 IP privada de NIC: 10.0.0.4

Reglas de puerto de entrada Reglas de puerto de salida Grupos de seguridad de aplicación

Grupo de seguridad de red VERTICAL-SCALABILITY-nsg (se conectó a la interfaz de red: vertical-scalability883)

Impactos 0 subredes, 1 interfaces de red

Prioridad	Nombre	Puerto	Protocolo
300	SSH	22	TCP
320	HTTP	80	TCP
330	Port_3000	3000	Cualquiera
65000	AllowVnetInBound	Cualquiera	Cualquiera
65001	AllowAzureLoadBalancerInBound	Cualquiera	Cualquiera
65500	DenyAllInBound	Cualquiera	Cualquiera

¿Necesita ayuda?

Port_3000
VERTICAL-SCALABILITY-nsg

Origen ⓘ

Any

Intervalos de puertos de origen * ⓘ

*

Destino ⓘ

Any

Servicio ⓘ

Custom

Intervalos de puertos de destino * ⓘ

3000

Protocolo

☒ Any

☐ TCP

☐ UDP

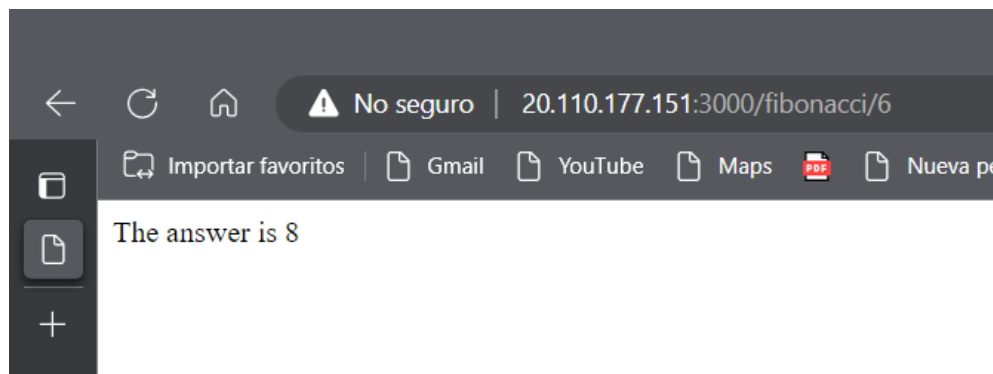
☐ ICMP

Acción

☒ Permitir

☐ Denegar

Guardar Cancelar



7. La función que calcula en enésimo número de la secuencia de Fibonacci está muy mal construido 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

The screenshot shows a web browser at the URL `20.110.177.151:3000/fibonacci/1000000`. The page displays the 1000000th Fibonacci number: `195328212870775773163201494759625633244354299659187339695340519457162525788701569476664`. The network tab shows a single request named `1000000` with a status of `200`, a type of `document`, and a size of `209 kB`. The response is a large string of digits.

1010000

The screenshot shows a web browser at the URL `20.110.177.151:3000/fibonacci/1010000`. The page displays the 1010000th Fibonacci number: `146949284679595242601574258743848413919764336681708419968365087254334695712876205283607`. The network tab shows a single request named `1010000` with a status of `200`, a type of `document`, and a size of `211 kB`. The response is a large string of digits.

1020000

The screenshot shows a web browser at the URL `20.110.177.151:3000/fibonacci/1020000`. The page displays the 1020000th Fibonacci number: `1105528584451383515152314253322190506257985489709022909934322977309431901100440337721750`. The network tab shows a single request named `1020000` with a status of `200`, a type of `document`, and a size of `213 kB`. The response is a large string of digits.

1030000

The screenshot shows a web browser at the URL `20.110.177.151:3000/fibonacci/1030000`. The page displays the 1030000th Fibonacci number: `8317110584811090520655370445566899120440918849704908502152696824551390247111142721586887`. The network tab shows a single request named `1030000` with a status of `200`, a type of `document`, and a size of `215 kB`. The response is a large string of digits.

1040000

The answer is 6257127084082073809575095119126550964699799149599567693502949730030396059384386844878758...

Nombre	Estado	Tipo	Iniciador	Tamaño	Hora	Completado por	Cascada
1040000	200	docum...	Otros	218 KB	3.1 m		

1050000

The answer is 47073606809861481028868612274344811789406303477807063735473769722185444888474957158338...

Nombre	Estado	Tipo	Iniciador	Tamaño	Hora	Completado por	Cascada
1050000	200	docum...	Otros	220 KB	3.1 m		

1060000

The answer is 35414407096296260040802563456349397077153948759670009648634918145420639971908963030687...

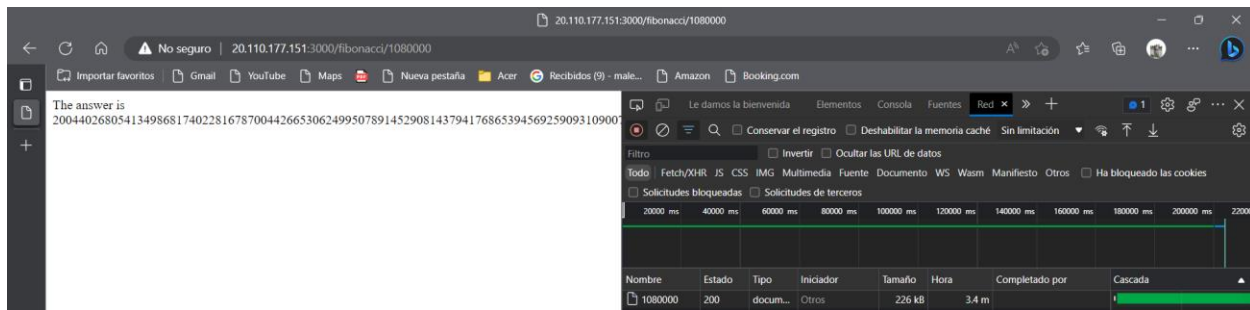
Nombre	Estado	Tipo	Iniciador	Tamaño	Hora	Completado por	Cascada
1060000	200	docum...	Otros	222 KB	3.2 m		

1070000

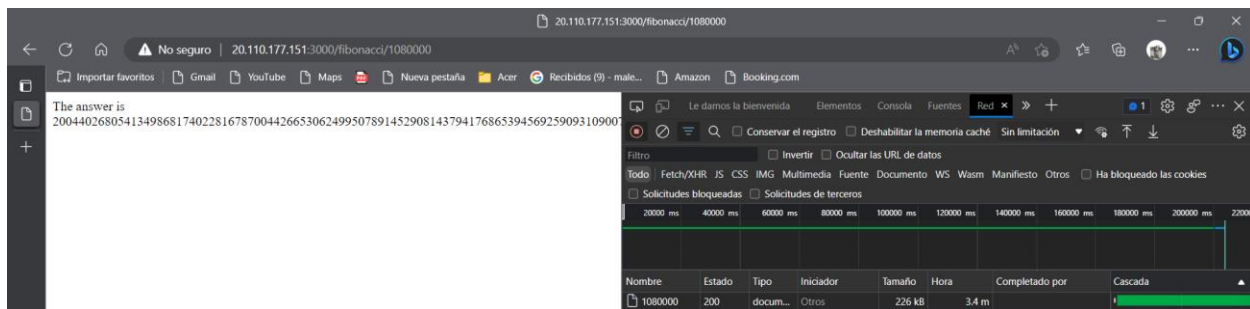
The answer is 266429601421461474799065122164609079803602086118409471992402269706455836538761622383884...

Nombre	Estado	Tipo	Iniciador	Tamaño	Hora	Completado por	Cascada
1070000	200	docum...	Otros	224 KB	3.3 m		

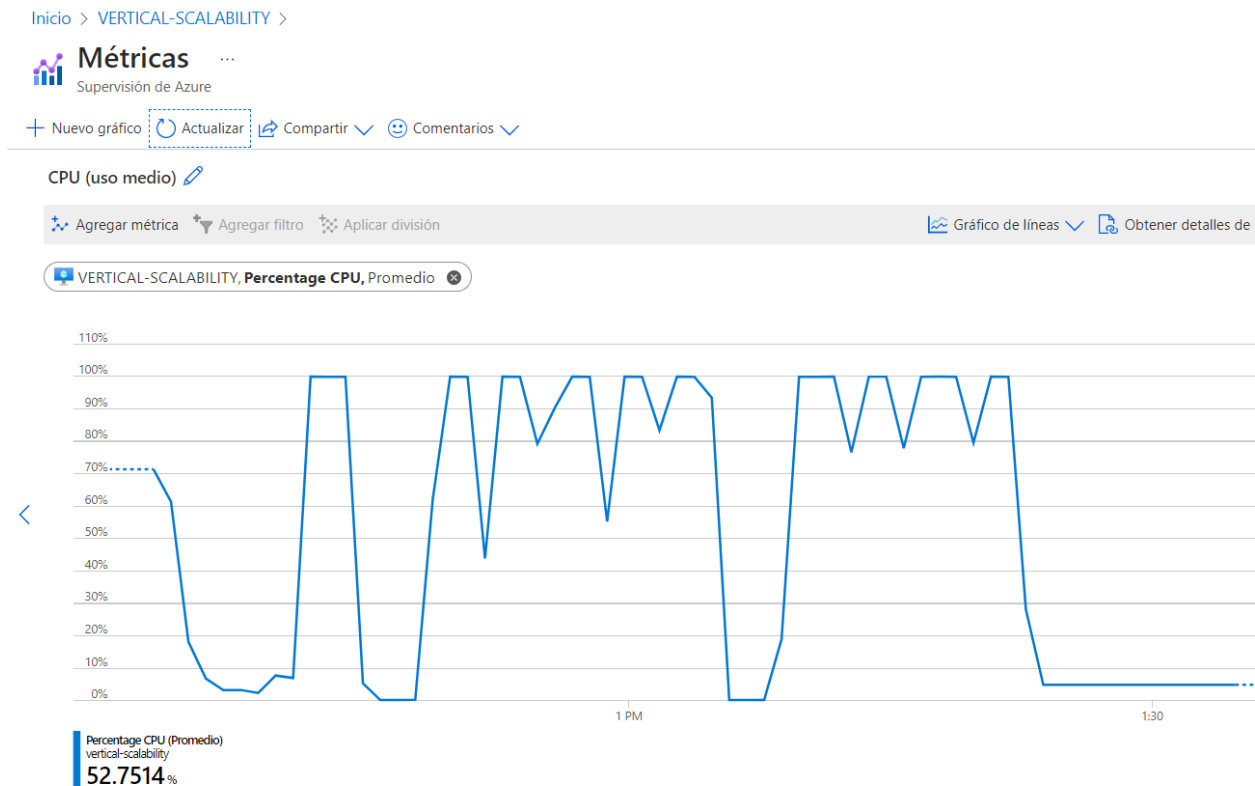
1080000



1090000

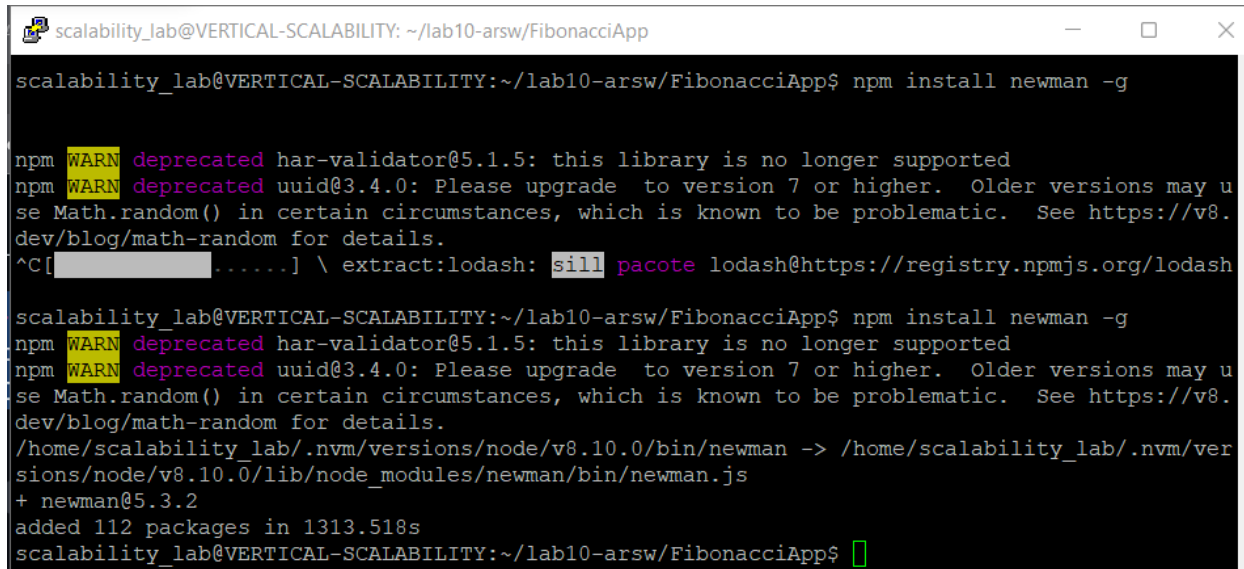


8. En Azure verificamos el consumo de CPU para la VM. (Los resultados pueden tardar 5 minutos en aparecer).



9. Ahora usaremos Postman para simular una carga concurrente a nuestro sistema. Seguiremos estos pasos.

Instalamos newman con el comando *npm install newman -g*.



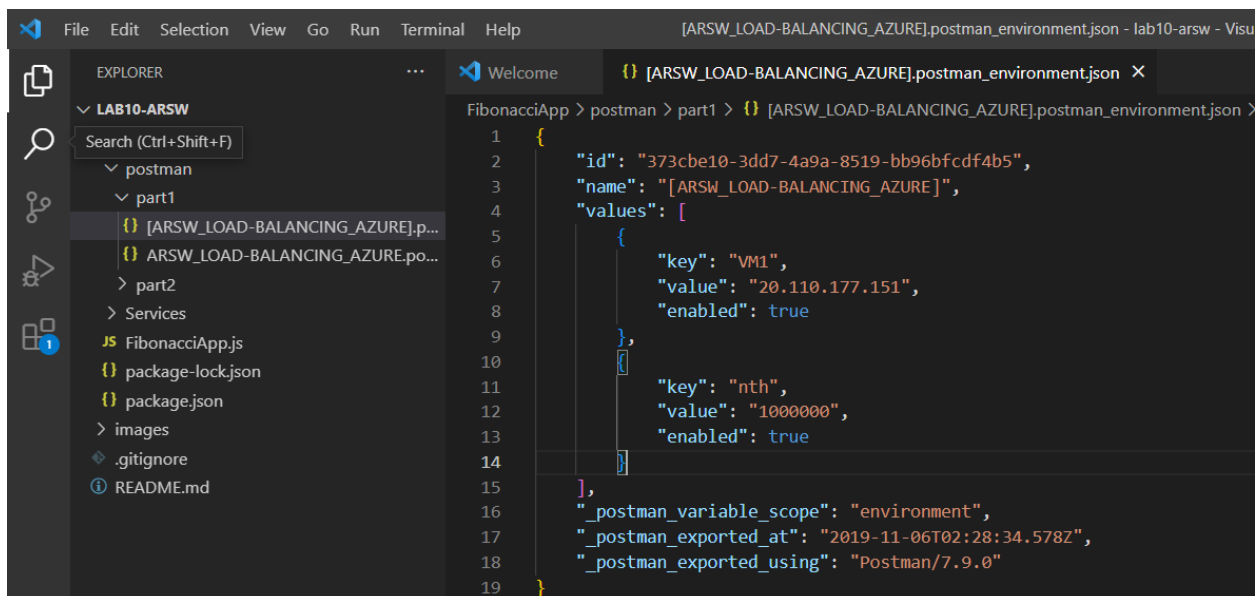
```
scalability_lab@VERTICAL-SCALABILITY: ~/lab10-arsw/FibonacciApp
scalability_lab@VERTICAL-SCALABILITY:~/lab10-arsw/FibonacciApp$ npm install newman -g

npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated uuid@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.
^C[REDACTED] \ extract:lodash: sill pacote lodash@https://registry.npmjs.org/lodash

scalability_lab@VERTICAL-SCALABILITY:~/lab10-arsw/FibonacciApp$ npm install newman -g
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated uuid@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.
/home/scalability_lab/.nvm/versions/node/v8.10.0/bin/newman -> /home/scalability_lab/.nvm/versions/node/v8.10.0/lib/node_modules/newman/bin/newman.js
+ newman@5.3.2
added 112 packages in 1313.518s
scalability_lab@VERTICAL-SCALABILITY:~/lab10-arsw/FibonacciApp$
```

Vamos hasta la ruta *FibonacciApp/postman* en una maquina diferente a la VM.

Para el archivo [ARSW_LOAD-BALANCING_AZURE].postman_environment.json cambiamos el valor del parámetro VM1 para que coincida con la IP de nuestra VM.



```
File Edit Selection View Go Run Terminal Help [ARSW_LOAD-BALANCING_AZURE].postman_environment.json - lab10-arsw - Visu
EXPLORER
LAB10-ARSW
  Search (Ctrl+Shift+F)
  postman
    part1
      {} [ARSW_LOAD-BALANCING_AZURE].p...
      {} ARSW_LOAD-BALANCING_AZURE.po...
    part2
    Services
  JS FibonacciApp.js
  {} package-lock.json
  {} package.json
  > images
  .gitignore
  README.md

FibonacciApp > postman > part1 > {} [ARSW_LOAD-BALANCING_AZURE].postman_environment.json >
1 {
2   "id": "373cbe10-3dd7-4a9a-8519-bb96bfcd4b5",
3   "name": "[ARSW_LOAD-BALANCING_AZURE]",
4   "values": [
5     {
6       "key": "VM1",
7       "value": "20.110.177.151",
8       "enabled": true
9     },
10    {
11      "key": "nth",
12      "value": "1000000",
13      "enabled": true
14    }
15  ],
16  "_postman_variable_scope": "environment",
17  "_postman_exported_at": "2019-11-06T02:28:34.578Z",
18  "_postman_exported_using": "Postman/7.9.0"
19 }
```

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

Preguntas

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

- Red virtual
- Máquina Virtual
- Dirección IP pública
- Grupo de seguridad de red
- Interfaz de red
- Disco
- Clave SSH

¿Brevemente describa para qué sirve cada recurso?

- Red virtual: Es la red privada de Azure quien permite la comunicación entre los recursos, el internet y las redes on-premise de forma segura
- Máquina Virtual: Realiza una emulación de un computador con sus respectivos componentes y recursos
- Dirección IP pública: El identificador unico para el ordenador que esta siendo simulado que permite las diferentes conexiones
- Grupo de seguridad de red: Tiene la tarea de filtrar el tráfico de red bajo premisas de seguridad que permiten o restringen dicho tráfico de entrada en la red, así como también el tráfico de salida
- Interfaz de red: Realiza la función de conectar la máquina virtual con el internet y recursos del sistema

- Disco: Tiene el papel de almacenar los distintos datos del sistema
- Clave SSH: es un protocolo de conexión cifrada que permite que las conexiones se realicen inicios de sesión seguros en un contexto de conexiones que no son consideradas como seguras componiéndose de una llave pública y una llave privada, de manera que al conectarse el cliente SSH se asegura de que la llave privada sea la correcta para así otorgar el acceso a la VM

¿Al cerrar la conexión ssh con la VM, por qué se cae la aplicación que ejecutamos con el comando `npm FibonacciApp.js`?

Al utilizar el comando `npm FibonacciApp.js`, y cerrar la conexión SSH la aplicación se detiene ya que el comando corre en dicha consola de la máquina virtual, por ende al terminar la conexión se termina por completo la conexión ya que el SSH hace un llamado a todos los procesos para que se cierren, puede suceder por inactividad, si existe algún error o simplemente por terminar la conexión, es por esto que se utiliza el comando `forever start FibonacciApp.js`, para que sea posible que siga en ejecución el script sin importar si se cierra la conexión siempre y cuando la máquina virtual este encendida

¿Por qué debemos crear un Inbound port rule antes de acceder al servicio?

Se debía agregar este *Inbound port rule* para permitir la conexión al servicio por el puerto 3000 como se realizó en este caso, además de poder elegir porque protocolo se recibirá o denegara según la elección que se realice es una forma de personalizar la conexión (el tráfico de red) según la necesidad.

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

B1ls			B2ms	
1000000	3.1 m		1000000	
1010000	2.9 m		1010000	
1020000	3.0 m		1020000	

1030000	3.0 m		1030000	
1040000	3.1 m		1040000	
1050000	3.1 m		1050000	
1060000	3.2 m		1060000	
1070000	3.3 m		1070000	
1080000	3.4 m		1080000	
1090000	3.5 m		1090000	

La manera en la cual se está encontrando el número de Fibonacci es de Complejidad $O(n)$ razón por la cual al enviar solicitudes de números grandes tardara bastante al no contar con memorización y tener que calcular siempre desde cero las veces que lo necesita en las iteraciones para encontrar el resultado.

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

B1ls



B2ms

El alto consumo de la CPU se debe a que en la operación de encontrar el número de Fibonacci, la VM se encuentra realizando por un tiempo prolongado operaciones matemáticas de números grandes esto en la capacidad de la máquina que contaba con bajas capacidades llegando a valores bastantes altos, de la misma forma, aunque mejora con la configuración de la VM en Bm2, baja el consumo de CPU pero sigue siendo alto para el momento en el cual se encuentra realizando dichas operaciones matemáticas.

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

Tiempos de ejecución de cada petición.

Si hubo fallos documentelos y explique.

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

Tamaño	Vcpu	RAM(GiB)	Discos de datos max	Gib de Almacenamiento temporal (SSD)	Base CPU Performance VM	Rendimiento máximo de almacenamiento temporal y en caché: IOPS / MBps	Rendimiento máximo del disco sin almacenamiento en la caché: IOPS / MBps	Costo
B1ls	1	0.5	2	4	5 %	200/10	160/10	\$ 3.80 UD
B2ms	2	8	3	16	60 %	2400/22.5 \$	1920/22.5	\$ 60.74 US

¿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?

Es una buena solución para reducir la carga del servidor y obtener tiempos de respuesta más rápidos porque el nuevo tamaño permite completar más operaciones en menos tiempo. Sin embargo, el sistema tampoco admite que muchos usuarios envíen solicitudes al mismo tiempo sin afectar el tiempo o la cantidad de respuesta.

La aplicación de Fibonacci se detiene a pesar de haber puesto forever ya que la VM cambio de tamaño y se debe reiniciar, razón por la cual el servicio se detiene

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

La Máquina Virtual debe reiniciarse para poder cambiar su tamaño, lo que representa negativamente que el servicio no esté disponible durante el escalamiento vertical lo que podría representar sobrecostos y falta de disponibilidad

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

Se presento una mejora tanto en el consumo de CPU como también en los tiempos de respuesta para las solicitudes, sin embargo, a pesar de mejorar con el escalamiento vertical, no representa una mejora realmente significativa ya que aun tarda bastante en las solicitudes y el consumo de CPU continua siendo alto para la operación de la Aplicación de Fibonacci

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