Implementación de Alta Disponibilidad y Balanceo de Carga en Azure

Proyecto: MarketOnline S.A.C. (Proyecto 15)
Responsable: Silva Pino, Jesus Francisco
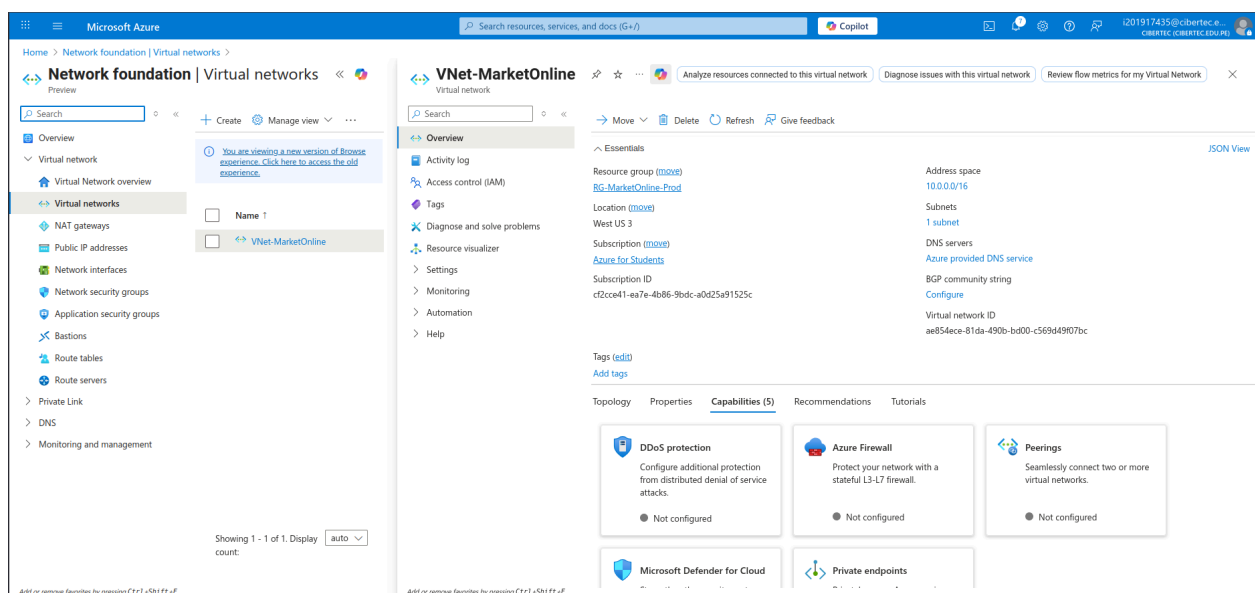Fecha: 13/12/2025

1. Objetivo

Implementar una arquitectura de alta disponibilidad utilizando Microsoft Azure, desplegando dos servidores web en diferentes zonas de disponibilidad y un Azure Load Balancer (Standard) para distribuir el tráfico y garantizar la continuidad del servicio ante fallos.

2. Fase 1: Despliegue de Infraestructura Base

Se crearon los recursos de red y cómputo en la región **West US 3** para asegurar compatibilidad de costos y recursos.

2.1. Configuración de Red (VNet)

- **Nombre:** VNet-MarketOnline
- **Región:** West US 3
- **Espacio de direcciones:** 10.0.0.0/16
- **Subnet:** WebSubnet (10.0.1.0/24)

```
Silva@VM-Web01: ~                          Silva@VM-Web02: ~

Scanning processes...                      Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
Scanning linux images...                   Scanning processes...
                                           Scanning linux images...
Running kernel seems to be up-to-date.
                                           Running kernel seems to be up-to-date.
No services need to be restarted.
                                           No services need to be restarted.
No containers need to be restarted.
                                           No containers need to be restarted.
No user sessions are running outdated binaries.
                                           No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
<h1>Hola desde el Servidor 1 (Zona 1)</h1> No VM guests are running outdated hypervisor (qemu) binaries on this host.
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-in   <h1>Hola desde el Servidor 1 (Zona 1)</h1>
stall.                                     Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-in
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2   stall.
Silva@VM-Web01:~$ ping 20.163.11.29"       Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
ping: 20.163.11.29": Name or service not known   Silva@VM-Web02:~$ vi /var/www/html/index.html
Silva@VM-Web01:~$ ping 10.0.1.5            Silva@VM-Web02:~$ sudo vi /var/www/html/index.html
PING 10.0.1.5 (10.0.1.5) 56(84) bytes of data.   Silva@VM-Web02:~$ ping 10.0.1.4
64 bytes from 10.0.1.5: icmp_seq=1 ttl=64 time=1.94 ms   PING 10.0.1.4 (10.0.1.4) 56(84) bytes of data.
64 bytes from 10.0.1.5: icmp_seq=2 ttl=64 time=0.854 ms   64 bytes from 10.0.1.4: icmp_seq=1 ttl=64 time=0.749 ms
64 bytes from 10.0.1.5: icmp_seq=3 ttl=64 time=0.840 ms   64 bytes from 10.0.1.4: icmp_seq=2 ttl=64 time=0.847 ms
64 bytes from 10.0.1.5: icmp_seq=4 ttl=64 time=0.719 ms   64 bytes from 10.0.1.4: icmp_seq=3 ttl=64 time=0.774 ms
64 bytes from 10.0.1.5: icmp_seq=5 ttl=64 time=0.801 ms   64 bytes from 10.0.1.4: icmp_seq=4 ttl=64 time=0.949 ms
64 bytes from 10.0.1.5: icmp_seq=6 ttl=64 time=0.773 ms   64 bytes from 10.0.1.4: icmp_seq=5 ttl=64 time=0.945 ms
64 bytes from 10.0.1.5: icmp_seq=7 ttl=64 time=0.761 ms   64 bytes from 10.0.1.4: icmp_seq=6 ttl=64 time=0.871 ms
^C                                         ^C
--- 10.0.1.5 ping statistics ---           --- 10.0.1.4 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6144ms   6 packets transmitted, 6 received, 0% packet loss, time 5104ms
rtt min/avg/max/mdev = 0.719/0.954/1.936/0.402 ms   rtt min/avg/max/mdev = 0.749/0.855/0.949/0.076 ms
Silva@VM-Web01:~$                          Silva@VM-Web02:~$
```

2.2. Aprovisionamiento de Servidores (Máquinas Virtuales)

Se desplegaron dos máquinas virtuales con redundancia de zona para cumplir el requisito de tolerancia a fallos.

Servidor 1 (VM-Web01):
- **Imagen:** Ubuntu Server 20.04/24.04 LTS
- **Tamaño:** Standard_B2ats_v2 (2 vCPUs, 1 GiB RAM)
- **Zona de Disponibilidad:** Zona 1
- **Red:** VNet-MarketOnline
- **IP Privada:** 10.0.1.4
- **IP Pública: 1**34.33.72.66

Servidor 2 (VM-Web02):
- **Configuración:** Idéntica a VM-Web01.
- **Zona de Disponibilidad:** Zona 2 (Para redundancia física).
- **Red:** VNet-MarketOnline
- **IP Privada:** 10.0.1.5
- **IP Pública:** 20.163.11.29

Compute infrastructure | Virtual machines
Microsoft

List VMs by security posture   Show me protected VMs in XXX region   Recommend VM sizes for my workloads

Virtual machines   Get started

+ Create ∨   Reservations ∨   Manage view ∨   Refresh   Export to CSV   Open query   Assign tags   Start   Restart   Stop   Delete   Services ∨   Maintenance ∨   ⋯   Group by none ∨

You are viewing a new version of Browse experience. Click here to access the old experience.

Filter for any field...   Subscription equals all   Type equals all   Resource Group equals all   Location equals all   + Add filter

	Name ↑	Subscription	Resource Group	Location	Status	Operating system	Size	Public IP address	Disks	Update status
☐	VM-Web01 ⋯	Azure for Students	RG-MarketOnline...	West US 3	Running	Linux	Standard_B2ats_v2	134.33.72.66	1	Enable periodic as...
☐	VM-Web02 ⋯	Azure for Students	RG-MARKETONLI...	West US 3	Running	Linux	Standard_B2ats_v2	20.163.11.29	1	Enable periodic as...

Showing 1 - 2 of 2. Display count: auto ∨

Give feedback

Add or remove favorites by pressing Ctrl+Shift+F

Compute infrastructure | Virtual machines
Microsoft

Silva@VM-Web01: ~

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

System information as of Sat Dec 13 20:15:12 UTC 2025

  System load:  0.15              Processes:           130
  Usage of /:   5.6% of 28.02GB   Users logged in:     0
  Memory usage: 34%               IPv4 address for eth0: 10.0.1.4
  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 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.

Silva@VM-Web01:~$
```

Silva@VM-Web02: ~

```
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1014-azure x86_64)

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

System information as of Sat Dec 13 20:15:38 UTC 2025

  System load:  0.39              Processes:           131
  Usage of /:   5.6% of 28.02GB   Users logged in:     0
  Memory usage: 35%               IPv4 address for eth0: 10.0.1.5
  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 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.
```

Update status

Enable periodic as...

Enable periodic as...

Showing 1 - 2 of 2. Display count: auto ∨

Give feedback

Add or remove favorites by pressing Ctrl+Shift+F

3. Fase 2: Configuración del Servidor Web (Apache)

Se instaló el servidor web Apache en ambas instancias para servir el contenido de la aplicación. Se personalizaron los archivos index.html para diferenciar la respuesta de cada nodo durante las pruebas.
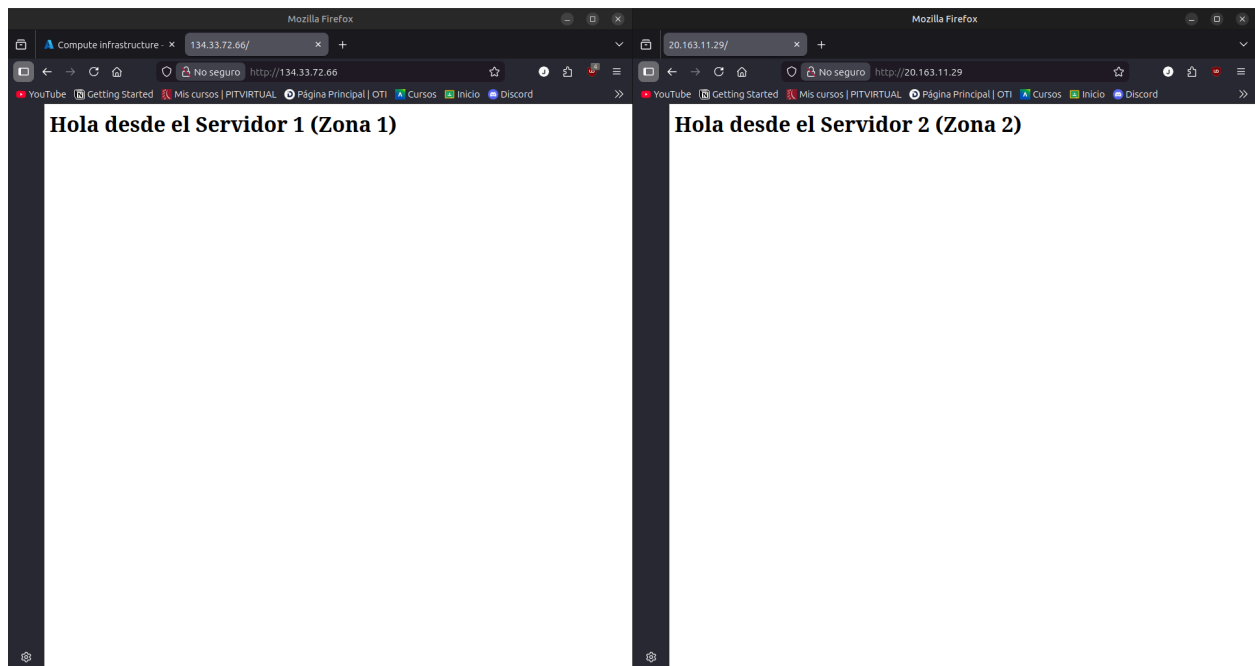**Comandos ejecutados:**

Bash

```
sudo apt-get update
sudo apt-get install apache2 -y
echo "<h1>Hola desde el Servidor X...</h1>" | sudo tee /var/www/html/index.html
```


4. Fase 3: Implementación del Azure Load Balancer

Se configuró un balanceador de carga de capa 4 para gestionar el tráfico entrante.

4.1. Creación del Recurso

- **Nombre:** LB-MarketOnline
- **SKU:** Standard (Requerido para soporte de Zonas de Disponibilidad).
- **Región:** West US 3.
- **Frontend IP:** 4.249.44.56 (IP-Publica-LB).

Home > Load balancing and content delivery | Load balancers >

Create load balancer   ...

                                                                                                                                        ✕

Basics   Frontend IP configuration   Backend pools   Inbound rules   Outbound rules   Tags   Review + create

Azure load balancer is a layer 4 load balancer that distributes incoming traffic among healthy virtual machine instances. Load balancers uses a hash-based distribution algorithm. By default, it uses a 5-tuple (source IP, source port, destination IP, destination port, protocol type) hash to map traffic to available servers. Load balancers can either be internet-facing where it is accessible via public IP addresses, or internal where it is only accessible from a virtual network. Azure load balancers also support Network Address Translation (NAT) to route traffic between public and private IP addresses. Learn more. ⧉

Project details

Subscription *   Azure for Students   ⌄

    Resource group *   RG-MarketOnline-Prod   ⌄
                                  Create new

Instance details

Name *   LB-MarketOnline

Region *   West US 3   ⌄

SKU * ⓘ   ⦿ Standard (Distribute traffic to backend resources)
                                  ○ Gateway (Direct traffic to network virtual appliances)

Type * ⓘ   ⦿ Public
                                  ○ Internal

Tier *   ⦿ Regional
                                  ○ Global

Review + create   < Previous   Next : Frontend IP configuration >   Download a template for automation   Give feedback

---

Home > Load balancing and content delivery | Load balancers >

Create load balancer   ...

Basics   Frontend IP configuration   Backend pools   Inbound rules   Outbound rules   Tags   Review + create

A frontend IP configuration is an IP address used for inbound and/or outbound communication as defined within load balancing, inbound NAT, and outbound rules.

+ Add a frontend IP configuration

Name ↑↓	IP address ↑↓
Add a frontend IP to get started	

Add frontend IP configuration

LB-MarketOnline

Name *   IP-Frontal-LB

IP version   ⦿ IPv4
                          ○ IPv6

IP type   ⦿ IP address
                          ○ IP prefix

Public IP address *   (new) IP-Publica-LB   ⌄
                          Create new

Gateway Load balancer ⓘ   None   ⌄

                                                  Give feedback

Save   Cancel

Review + create   < Previous   Next : Backend pools >   Download a template for automation   Give feedback

4.2. Configuración del Backend Pool

Se creó el grupo Pool-Servidores integrando a **VM-Web01** y **VM-Web02** mediante su interfaz de red (NIC).

4.3. Configuración de Health Probe (Sonda de Salud)

Se definió un monitor para verificar la disponibilidad de los servidores.
- **Protocolo:** HTTP
- **Puerto:** 80
- **Intervalo:** 5 segundos

Add health probe ...
LB-MarketOnline

ⓘ Health probes are used to check the status of a backend pool instance. If the health probe fails to get a response from a backend instance then no new connections will be sent to that backend instance until the health probe succeeds again.

Name *	Sonda-Web
Protocol *	HTTP
Port * ⓘ	80
Path * ⓘ	/
Interval (seconds) * ⓘ	5
Used by * ⓘ	Not used

Save   Cancel

Give feedback

LB-MarketOnline | Health probes ☆ ...
Load balancer

🔍 he

+ Add   🔄 Refresh   Give feedback

🔍 Type to start filtering ...

- Overview
- Diagnose and solve problems
- Settings
  - Frontend IP configuration
  - **Health probes**
  - Properties
- Monitoring
  - Diagnostic settings
- Automation
  - Export template
- Help
  - Resource health
  - Support + Troubleshooting

ⓘ To check the health status of your instances, navigate to the Load Balancing Rules page

Name	Protocol	Port	Path	Used By	
Sonda-Web	Http	80	/	Regla-Puerto80	🗑

4.4. Regla de Balanceo (Load Balancing Rule)

Se estableció la lógica de distribución de tráfico.
- **Puerto Entrada:** 80 (TCP)
- **Puerto Backend:** 80
- **Persistencia de sesión:** Ninguna (None) - *Para permitir la distribución Round-Robin.*

Home > CreateLoadBalancerBladeV2-20251213153613 | Overview > LB-MarketOnline | Load balancing rules >

Add load balancing rule ...
LB-MarketOnline

A load balancing rule distributes incoming traffic that is sent to a selected IP address and port combination across a group of backend pool instances. Only backend instances that the health probe considers healthy receive new traffic. Learn more.

Name *
Regla-Puerto80

IP version *
◉ IPv4
○ IPv6

Frontend IP address * ⓘ
IP-Frontal-LB (4.249.44.56)

Backend pool * ⓘ
Pool-Servidores

Protocol
◉ TCP
○ UDP

Port *
80

Backend port * ⓘ
80

Health probe * ⓘ
Sonda-Web (HTTP:80)
Create new

Session persistence
None

ⓘ Session persistence specifies that traffic from a client should be handled by the same virtual machine in the backend pool for the duration of a session. Learn more.

Save   Cancel

Give feedback

Home > CreateLoadBalancerBladeV2-20251213153613 | Overview > LB-MarketOnline

LB-MarketOnline | Load balancing rules ☆ ...
Load balancer

Search

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Resource visualizer
- Settings
  - Frontend IP configuration
  - Backend pools
  - Health probes
  - **Load balancing rules**
  - Inbound NAT rules
  - Outbound rules
  - Properties
  - Locks
- Monitoring
- Automation
- Help

+ Add ↻ Refresh ↓ Export to CSV 🗑 Delete

A load balancer rule is used to define how incoming traffic is distributed to all the instances within the backend pool. A load-balancing rule maps a given frontend IP configuration and port to multiple backend IP addresses and ports. An example would be a rule created on port 80 to load balance web traffic. Learn more.

Filter by name...

| Name ↑↓	Protocol ↑↓	Backend pool ↑↓	Health probe ↑↓	Health st...	
Regla-Puerto80	TCP/80	Pool-Servidores	Sonda-Web	View details	🗑

Give feedback

Add or remove favorites by pressing Ctrl+Shift+F

---

5. Fase 4: Validación Funcional y Pruebas

Se realizaron pruebas de conectividad utilizando la IP Pública del Balanceador (4.249.44.56).
Prueba 1: Distribución de Tráfico
Al acceder desde un navegador en modo incógnito, se observó que las peticiones fueron atendidas alternadamente por el Servidor 1 y el Servidor 2, confirmando el funcionamiento del algoritmo de balanceo.

Hola desde el Servidor 1 (Zona 1)

Hola desde el Servidor 2 (Zona 2)
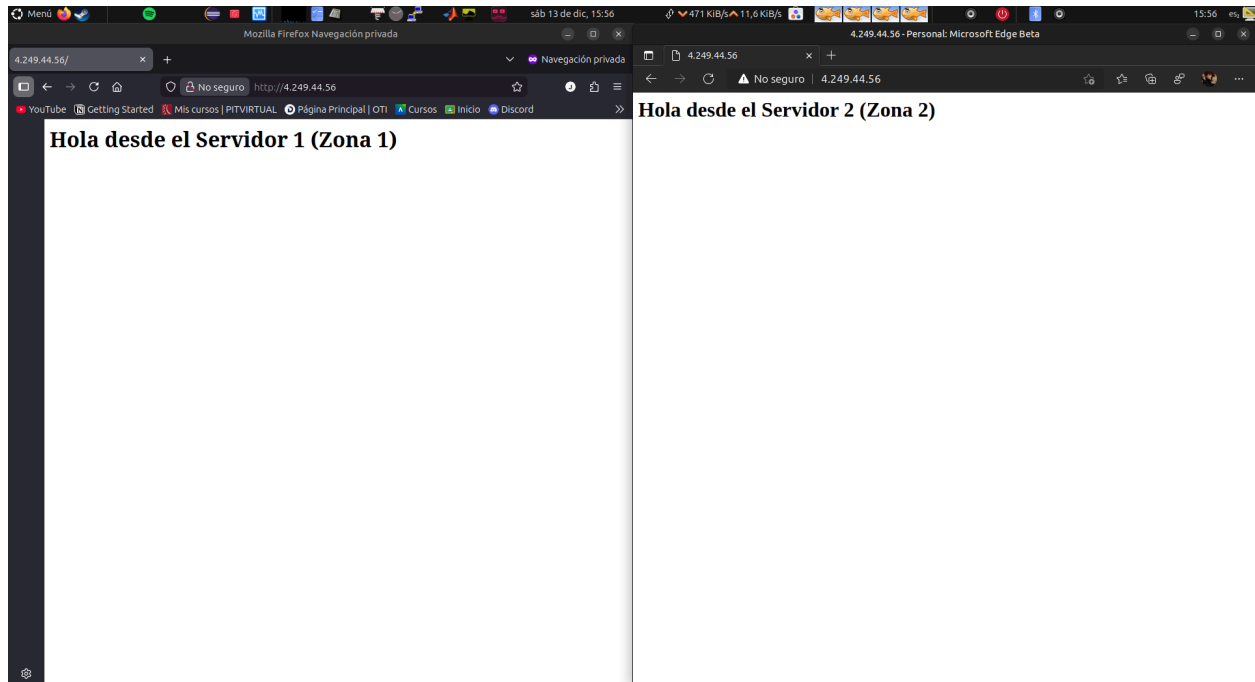
6. Conclusiones

La implementación se ha completado exitosamente cumpliendo con los requisitos del Sprint 3. La infraestructura actual soporta fallos de zona y distribuye la carga equitativamente, mejorando la disponibilidad del servicio para MarketOnline S.A.C.