

# **COMPARATIVA DE INFRAESTRUCTURA**



**Esteban Garcés Pérez  
Christian Ballesteros Pérez  
Gonzalo Bravo Maroto**

## ÍNDICE

<b>1.</b>	<i>Introducción</i>	<b>2</b>
<b>2.</b>	<i>Requisitos</i>	<b>2</b>
<b>3.</b>	<i>Descripción de los procesos en ambas plataformas</i>	<b>3</b>
3.1.	AWS	3
3.1.1.	Creación de recursos en AWS	3
3.2.	AZURE	26
3.2.1.	Creación de recursos en Azure	26
<b>4.</b>	<i>Comparación de características</i>	<b>49</b>
4.1.	AWS	49
4.2.	AZURE	52
<b>5.</b>	<i>Precios en Azure y AWS</i>	<b>55</b>
<b>6.</b>	<i>Conclusiones y recomendaciones</i>	<b>60</b>

## 1. Introducción

En el ámbito de la computación en la nube, Azure y AWS se han consolidado como dos de las principales plataformas para la creación y gestión de infraestructura. Ambas ofrecen un amplio conjunto de servicios que permiten a las organizaciones desplegar soluciones escalables, seguras y altamente disponibles.

El propósito de este informe es analizar y comparar el proceso de creación de infraestructura en Azure y AWS, centrándose en la implementación de máquinas virtuales, redes virtuales y balanceadores de carga. Para ello, se examinarán las diferencias en configuración, facilidad de uso, costos aproximados y opciones de implementación, considerando tanto las interfaces gráficas como el uso de herramientas de línea de comandos y tecnologías de infraestructura como código (IaC).

A través de esta investigación, los estudiantes obtendrán una visión clara de las similitudes y diferencias entre ambas plataformas, lo que les permitirá tomar decisiones informadas al trabajar con servicios en la nube. Finalmente, se presentarán conclusiones y recomendaciones basadas en los hallazgos obtenidos.

## 2. Requisitos

- 4 máquinas virtual (VM) Linux
- Red virtual y subredes (si aplica)
- Balanceador de carga
- Se deben considerar aspectos como:
  - Consola grafica vs. CLI vs. Infrastructure as Code (Terraform o ARM Templates/CloudFormation).
  - Facilidad de uso
  - Costos aproximados (sin entrar en alto nivel de detalle).
  - Diferencias en configuraciones y opciones disponibles.
  - Implementación práctica (Opcional según disponibilidad de recursos).

### 3. Descripción de los procesos en ambas plataformas.

#### 3.1. AWS

##### 3.1.1. Creación de recursos en AWS

###### ■ Máquinas virtuales

Una instancia de computación en la nube es un recurso de servidor que brindan servicios en la nube de terceros. Nos referimos al término de “instancia” pues una máquina virtual en AWS es eso, una instancia.

Para la creación de estas instancias en AWS será necesario utilizar el servicio “Amazon Elastic Compute Cloud” o también conocido como EC2.

Este servicio proporciona capacidad de computación escalable en la nube de AWS. Elimina la necesidad de invertir inicialmente en hardware, de manera que se puede desarrollar e implementar aplicaciones en menos tiempo.

Para crear estas instancias podremos hacerlo desde tres métodos distintos:

- **Consola Gráfica (AWS Management Console):**

En primer lugar, en el apartado de EC2, accederemos al panel de instancias y seleccionaremos la opción “Instancias” que nos redirigirá a nuestra página de instancias la cual estará vacía ya que aún no hemos creado ninguna.

Como se puede apreciar, no existe ninguna instancia todavía y para proceder a crear nuestra primera instancia, simplemente seleccionaremos la opción “Lanzar Instancias”.

## COMPARATIVA AWS Y AZURE

Aquí, aparecerán los pasos que tendremos que ir siguiendo para configurar nuestra instancia. En primer lugar, deberemos elegir una imagen o AMI para la propia instancia, se nos especifica que sea Linux y, por tanto, escogemos la opción de “Amazon Linux 2 AMI”, por ejemplo, ya que hay múltiples opciones dentro de las imágenes que se nos ofrecen.

**Paso 1: Elegir una imagen de Amazon Machine (AMI)**

Una AMI es una plantilla que contiene la configuración de software (sistema operativo, servicios de aplicaciones y aplicaciones) necesaria para lanzar la instancia. Puede seleccionar una AMI proporcionada por AWS, nuestra comunidad de usuarios o AWS Marketplace, o puede seleccionar una de sus propias AMI.

Cancelar y salir

Para buscar una AMI, escriba un término de búsqueda, por ejemplo, "Windows"

Buscar por parámetros de Systems Manager

Inicio rápido

Mis AMI

AWS Marketplace

AMI de la comunidad

Solo copia gratuita

**Amazon Linux 2 AMI (HVM, SSD Volume Type) - ami-0dc2d3e4c0f9ebdf18 (64 bits x86) / ami-008a8487adc2b32ec (64 bits Arm)**

Aplicar para la copia

Aplicar para la copia complementarios

Tipo de dispositivo raíz: ebs Tipo de virtualización: hvm Habilitado para ENA: Sí

64 bits (x86)

64 bits (Arm)

**macOS Big Sur 11.4 - ami-059ff8b2c04bed21**

The macOS Big Sur AMI is an EBS-backed, AWS-supported image. This AMI includes the AWS Command Line Interface, Command Line Tools for Xcode, Amazon SSM Agent, and Homebrew. The AWS Homebrew Tap includes the latest versions of multiple AWS packages included in the AMI.

Tipo de dispositivo raíz: ebs Tipo de virtualización: hvm Habilitado para ENA: Sí

**macOS Catalina 10.15.7 - ami-093900cc07f14a8f7**

The macOS Catalina AMI is an EBS-backed, AWS-supported image. This AMI includes the AWS Command Line Interface, Command Line Tools for Xcode, Amazon SSM Agent, and Homebrew. The AWS Homebrew Tap includes the latest versions of multiple AWS packages included in the AMI.

Tipo de dispositivo raíz: ebs Tipo de virtualización: hvm Habilitado para ENA: Sí

**macOS Mojave 10.14.6 - ami-052c0a4e1da715c4e**

The macOS Mojave AMI is an EBS-backed, AWS-supported image. This AMI includes the AWS Command Line Interface, Command Line Tools for Xcode, Amazon SSM Agent, and Homebrew. The AWS Homebrew Tap includes the latest versions of multiple AWS packages included in the AMI.

Tipo de dispositivo raíz: ebs Tipo de virtualización: hvm Habilitado para ENA: Sí

64 bits (Mac)

64 bits (Mac)

**Selecciónar**

**Selecciónar**

**Selecciónar**

**Selecciónar**

**Selecciónar**

**Selecciónar**

1 de 448 de 448 instancias

El siguiente paso será escoger un tipo de instancia. Como se puede observar, los tipos de instancia están organizados por una serie de características como familia, tipo, cantidad de CPUs, memoria, etc. Las características de nuestra instancia van a depender del uso que le vayamos a dar o de las necesidades que tengamos. Hay que tener en cuenta que AWS nos ofrece una opción gratuita con unas características algo pobres pero que sin duda es muy atractiva si tenemos un presupuesto muy ajustado.

**Paso 2: Página Choose an Instance Type**

Amazon EC2 proporciona una amplia selección de tipos de instancias optimizados para adaptarse a diferentes casos de uso. Las instancias son servidores virtuales que pueden ejecutar aplicaciones. Tienen distintas combinaciones de CPU, memoria, almacenamiento y capacidad de red, lo que proporciona una gran flexibilidad para elegir la combinación de recursos adecuada para las aplicaciones. Más información acerca de los tipos de instancias y cómo pueden satisfacer sus necesidades de computación.

Filtrar por: Todas las familias de instancias Generación actual Mostrar/ocultar columnas

Seleccionada actualmente: t2.micro (t2, 1 vCPU, 2.5 GHz, 1 GiB memoria, EBS solo)

Familia	Tipo	vCPU	Memoria (GiB)	Almacenamiento de la instancia (GB)	Optimizado para EBS disponible	Desempeño de la red	Compatibilidad con IPv6
12	t2.nano	1	0.5	EBS solo	-	De bajo a moderado	<input checked="" type="checkbox"/> Sí
<b>12</b>	<b>t2.micro</b> Aplicar para la copia gratuita	<b>1</b>	<b>1</b>	EBS solo	-	De bajo a moderado	<input checked="" type="checkbox"/> Sí
12	t2.small	1	2	EBS solo	-	De bajo a moderado	<input checked="" type="checkbox"/> Sí
12	t2.medium	2	4	EBS solo	-	De bajo a moderado	<input checked="" type="checkbox"/> Sí
12	t2.large	2	8	EBS solo	-	De bajo a moderado	<input checked="" type="checkbox"/> Sí
12	t2.xlarge	4	16	EBS solo	-	Moderada	<input checked="" type="checkbox"/> Sí
12	t2.2xlarge	8	32	EBS solo	-	Moderada	<input checked="" type="checkbox"/> Sí
13	t3.nano	2	0.5	EBS solo	Sí	Hasta 5 gigabits	<input checked="" type="checkbox"/> Sí
13	t3.micro	2	1	EBS solo	Sí	Hasta 5 gigabits	<input checked="" type="checkbox"/> Sí

## COMPARATIVA AWS Y AZURE

En este tercer paso, podremos configurar nuestra instancia, hay algunas opciones que son muy específicas y que, probablemente, no necesitaremos, pero en cambio aquí podremos configurar el número de instancias que queremos lanzar, en nuestro caso necesitaremos cuatro instancias, así que simplemente en el apartado de número de instancias seleccionaríamos “4” y ya se nos lanzaría esa cantidad de instancias al finalizar la configuración.

Aquí, también podemos escoger el tipo de Red o subred, si la necesitáramos.

**Paso 3: Página Configuración de los detalles de la instancia**

Configure la instancia adecuada a sus requisitos. Puede lanzar varias instancias desde la misma AMI, solicitar instancias de spot para aprovecharse de los precios reducidos y asignar un rol de administración de acceso a la instancia, entre otras operaciones.

Número de instancias	1	Lanzar en grupo de Auto Scaling
Opción de compra	<input type="checkbox"/> Solicitar instancias de spot	
Red	vpc-28cca855 (predeterminado)	<input type="button"/> Crear nueva VPC
Subred	Sin preferencia (subred predeterminada de cualquier)	<input type="button"/> Crear nueva subred
Asignar automáticamente IP pública	<input type="checkbox"/> Usar configuración de subred (habilitar)	
Grupo de ubicación	<input type="checkbox"/> Agregue la instancia a un grupo de ubicación.	
Reserva de capacidad	<input type="button"/> Abrir	
Directorio de unión al dominio	Ningún directorio	<input type="button"/> Crear nuevo directorio
Role de IAM	Ninguno	<input type="button"/> Crear un nuevo rol de IAM
Comportamiento de cierre	<input type="button"/> Detener	
Detener: comportamiento de hibernación	<input type="checkbox"/> Habilitar la hibernación como un comportamiento de cierre adicional	
Habilitar la protección de terminación	<input type="checkbox"/> Protegerse contra la terminación accidental	
Monitoreo	<input type="checkbox"/> Habilitar monitoreo detallado de CloudWatch	

Se aplican cargos adicionales.

En este apartado podremos escoger el tipo de almacenamiento, así como añadir almacenamiento a nuestra instancia. Entre las opciones que nos ofrece está el tamaño del almacenamiento o el tipo de volumen.

**Paso 4: Adición de almacenamiento**

Su instancia se lanzará con la siguiente configuración de dispositivo de almacenamiento. Puede asociar volúmenes de EBS y volúmenes del almacenamiento adicional a la instancia o editar la configuración del volumen raíz. También puede asociar volúmenes de EBS adicionales después de lanzar una instancia, pero no volúmenes del almacenamiento adicional. Obtenga más información acerca de las opciones de almacenamiento de Amazon EC2.

Tipo de volumen	Dispositivo	Snapshot	Tamaño (GiB)	Tipo de volumen	IOPS	Velocidad (MB/s)	Eliminar al terminar	Cifrado
Raíz	/dev/xvda	snap-053c42bdb1128764a	8	SSD de uso general (gp2)	100/3000	N/D	<input checked="" type="checkbox"/>	No cifrado
<input type="button"/> Añadir nuevo volumen								

Podremos también agregar etiquetas a nuestra instancia, si fuera necesario.

**Paso 5: Agregar etiquetas**

Una etiqueta consta de un par de clave-valor en el que se distingue entre mayúsculas y minúsculas. Por ejemplo, puede definir una etiqueta con la clave = Nombre y el valor = Servidor web. Se puede aplicar una copia de una etiqueta a los volúmenes, las instancias o ambos. Las etiquetas se aplicarán a todas las instancias y los volúmenes. Más información sobre cómo etiquetar los recursos de Amazon EC2.

Clave (128 caracteres como máximo)	Valor (256 caracteres como máximo)	Instancias	Volúmenes	Interfaces de red
Actualmente, este recurso no tiene etiquetas				
Elija el botón "Añadir una etiqueta" o <a href="#">haga clic para añadir una etiqueta</a> Nombre. Asegúrese de que su <a href="#">Política de IAM</a> incluye permisos para crear etiquetas.				
<input type="button"/> Añadir etiqueta (Hasta 50 etiquetas como máximo)				

Por último, podremos configurar los grupos de seguridad que el propio AWS nos ofrece ya un grupo existente, pero con una seguridad mínima o nula. Por defecto, al escoger una imagen Linux, nos propone un grupo de tipo SSH, con protocolo TCP/IP y que el origen sea cualquiera. Podremos añadir cuantas reglas sean necesarias.

Paso 6: Página Configure Security Group

Un grupo de seguridad es un conjunto de reglas del firewall que controlan el tráfico de la instancia. En esta página, puede agregar reglas para permitir que determinado tráfico llegue a la instancia. Por ejemplo, si desea configurar un servidor web y permitir que el tráfico de Internet llegue a la instancia, agregue reglas que permitan el acceso sin restricción a los puertos HTTP y HTTPS. Puede crear un nuevo grupo de seguridad o seleccionar uno existente a continuación. Más información sobre los grupos de seguridad de Amazon EC2.

Asignar un grupo de seguridad:  Crear un nuevo grupo de seguridad  Seleccionar un grupo de seguridad existente

Nombre del grupo de seguridad: launch-wizard-1

Descripción: launch-wizard-1 created 2021-07-11T17:39:19.660+02:00

Tipo	Protocolo	Rango de puertos	Origen	Descripción
SSH	TCP	22	Personalizado 0.0.0.0/0	por ejemplo SSH for Admin Desktop

[Añadir regla](#)

Finalmente, se nos muestra un análisis de la configuración final de la instancia, donde se nos muestra la AMI que utilizaremos, el tipo de instancia escogido y los grupos de seguridad. Se nos advierte, como he comentado anteriormente, que nuestro grupo de seguridad está abierto a todo el mundo, podremos modificarlo o no.

Para lanzar nuestra instancia simplemente seleccionaremos “lanzar” y ya tendríamos lista nuestra instancia o instancias para arrancarlas cuando fuera necesario.

Paso 7: Página Review Instance Launch

Revise los detalles de lanzamiento de su instancia. Haga clic en **Lanzar** para asignar un par de claves a la instancia y completar el proceso de lanzamiento.

Mejore la seguridad de su instancia. Su grupo de seguridad, G, SSH Y WEB, está abierto a todo el mundo.

Su instancia puede estar accesible desde cualquier dirección IP. Le recomendamos que actualice las reglas de su grupo de seguridad para permitir el acceso únicamente desde direcciones IP conocidas. También puede abrir puertos adicionales en su grupo de seguridad para facilitar el acceso a la aplicación o el servicio que esté ejecutando, por ejemplo, HTTP (80) para los servidores web. [Editar grupos de seguridad](#)

Detalles de la AMI

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0dc2d3e4c0f9ebd18

Aplicar a: Amazon Linux 2 incluye cinco años de soporte. Proporciona el kernel de Linux 4.14 adaptado para un rendimiento óptimo en Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Brutus 2.29.1 y en los últimos paquetes de software a través de complementos.

Detalles de la instancia

Tipo de instancia	ECU	vCPU	Memoria (GiB)	Almacenamiento de la instancia (GB)	Optimizado para EBS disponible	Desempeño de la red
t2.micro	-	1	1	EBS solo	-	Low to Moderate

Grupos de seguridad

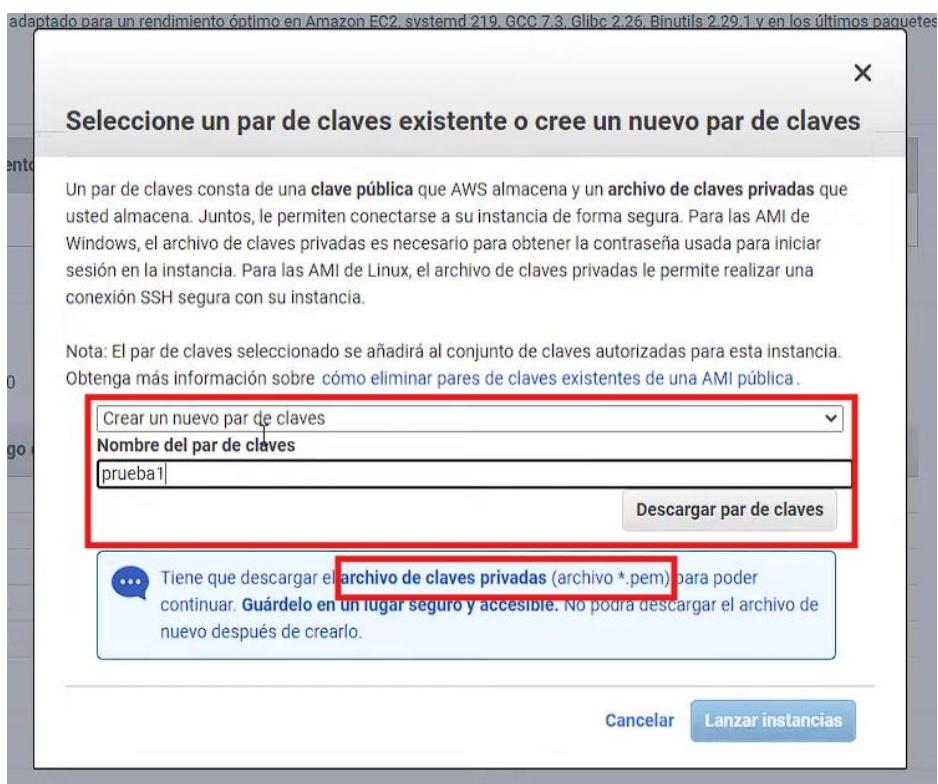
Nombre del grupo de seguridad: G, SSH Y WEB

Descripción: launch-wizard-1 created 2021-07-11T17:39:19.660+02:00

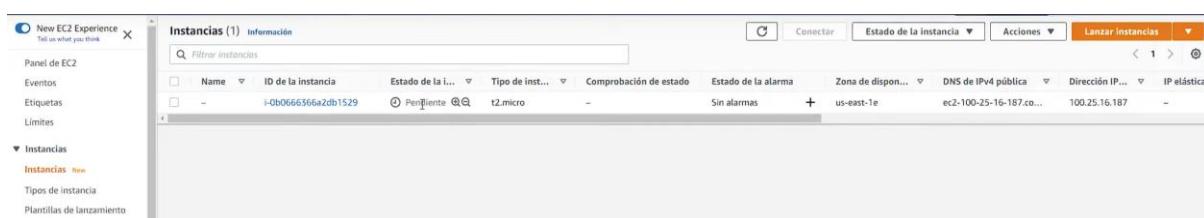
Tipo	Protocolo	Rango de puertos	Origen	Descripción
SSH	TCP	22	0.0.0.0/0	
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	>0	
HTTPS	TCP	443	0.0.0.0/0	
HTTPS	TCP	443	>0	

[Cancelar](#) [Anterior](#) [Lanzar](#)

Para confirmar el proceso se nos requerirá que seleccionemos un par de claves existentes o que las creamos. Simplemente seleccionamos la opción “Crear un nuevo par de claves” y escogeremos un nombre para éstas. Se nos generará un archivo. pem cuando seleccionemos la opción “Descargar par de claves” y simplemente seleccionando “lanzar instancias” ya estaría finalizado el proceso.



Una vez finalizado el proceso, nos llevará de vuelta al panel de instancias en el que aparecerá la instancia que hayamos creado.



- **CLI (AWS CLI):**

No siempre es necesario utilizar un entorno gráfico para crear instancias en EC2, a veces, puede ser tedioso y repetitivo, sobre todo si tenemos que crear múltiples instancias y de diferentes tipos.

Para facilitar este proceso podemos utilizar el CLI que nos proporciona AWS. Mediante el siguiente comando, podremos crear nuestra instancia con sus respectivas características:

```
aws ec2 run-instances --image-id ami-0abcdef1234567890 --count 4 --  
instance-type t2.micro --key-name MyKeyPair --security-groups  
MySecurityGroup
```

Desgranando este código, por ejemplo, vemos que:

- ***aws ec2 run-instances***: Es el comando principal que se utiliza para lanzar nuevas instancias EC2.
- ***--image-id ami-0abcdef1234567890***: Especifica la ID de la AMI que se utilizará en las instancias.
- ***--count 4***: Definirá cuántas instancias desearemos lanzar.
- ***--instance-type t2.micro***: Esto determinará el tipo de instancia que deseamos lanzar.
- ***--key-name MyKeyPair***: Especifica el nombre de la clave SSH que se usará para acceder de forma remota a las instancias.
- ***--security-groups MySecurityGroup***: Por último, asocia las instancias a un grupo de seguridad específico.

- **IaC (CloudFormation o Terraform):**

Para definir infraestructuras utilizando Infraestructura como Código (IaC) en la nube, se pueden usar dos de las herramientas más populares: AWS CloudFormation y Terraform. Ambas permiten crear y administrar recursos de manera automatizada, pero con diferencias clave en su enfoque, sintaxis y alcance.

- AWS CloudFormation es un servicio de AWS que permite describir y provisionar recursos de infraestructura en la nube mediante archivos de plantillas. Estas plantillas pueden estar en formato JSON o YAML.

```
AWSTemplateFormatVersion: '2010-09-09'
Description: Una plantilla básica de CloudFormation para lanzar una instancia EC2

Parameters:
  InstanceType:
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.micro
      - t2.small
      - t2.medium
    Description: Tipo de instancia EC2

Resources:
  MyEC2Instance:
    Type: 'AWS::EC2::Instance'
    Properties:
      InstanceType: !Ref InstanceType
      ImageId: ami-0abcdef1234567890
      KeyName: MyKeyPair
      SecurityGroups:
        - MySecurityGroup

Outputs:
  InstanceID:
    Description: ID de la instancia EC2 creada
    Value: !Ref MyEC2Instance
```

- Terraform es una herramienta de **HashiCorp** para la gestión de infraestructuras que permite definir la infraestructura en archivos de configuración usando HCL (HashiCorp Configuration Language). Terraform es agnóstico a proveedores, lo que significa que puedes usarlo para administrar infraestructuras de AWS, Azure, Google Cloud, entre otros. Un archivo de configuración de Terraform generalmente contiene “providers” (proveedores), “resources” (recursos), “variables” (parámetros) y “outputs” (salidas de recursos).

```
provider "aws" {  
    region = "us-east-1"  
}  
  
resource "aws_instance" "my_ec2_instance" {  
    ami          = "ami-0abcdef1234567890"  
    instance_type = var.instance_type  
    key_name      = "MyKeyPair"  
    security_groups = ["MySecurityGroup"]  
  
    tags = {  
        Name = "MyEC2Instance"  
    }  
}  
  
output "instance_id" {  
    value = aws_instance.my_ec2_instance.id  
}  
  
variable "instance_type" {  
    default = "t2.micro"  
}
```

## ■ Red virtual y Subredes

Crear una **Red Virtual (VPC)** y sus **subredes** en AWS es un paso fundamental para configurar el aislamiento y la seguridad de los recursos en la nube. La **VPC** (Virtual Private Cloud) te permite definir un espacio de red virtual dentro de AWS donde puedes lanzar recursos como instancias EC2, bases de datos RDS, y más, asegurando que solo los recursos dentro de la misma red puedan comunicarse entre sí, a menos que se configuren reglas explícitas para habilitar la comunicación con otros recursos.

Para crear una VPC podremos hacerlo desde tres métodos distintos:

- **Consola Gráfica (AWS Management Console):**

En primer lugar, accederemos al apartado de VPC, y simplemente seleccionaremos “crear VPC”.

En la configuración de la VPC, le asignaremos un nombre y a continuación una IP.

## COMPARATIVA AWS Y AZURE

En este mismo apartado, también podremos escoger las zonas de disponibilidad, y la cantidad de subredes públicas o privadas.

The screenshot shows the AWS VPC creation process. On the left, under 'Tenerencia' (Information), the 'Predeterminado' (Default) option is selected. The 'Número de zonas de disponibilidad (AZ)' (Number of availability zones) section shows a dropdown menu with options 1, 2, and 3, with '2' highlighted. Below it, the 'Personalizar las zonas de disponibilidad' (Customize availability zones) link is visible. The 'Cantidad de subredes públicas' (Number of public subnets) section shows a dropdown menu with options 0, 2, and 4, with '2' highlighted. Below it, the 'Cantidad de subredes privadas' (Number of private subnets) section shows a dropdown menu with options 0, 2, and 4, with '2' highlighted. The 'Personalizar bloques de CIDR de subredes' (Customize subnet CIDR blocks) link is also visible. On the right, the 'Vista previa' (Preview) section displays the VPC structure. It shows a central 'VPC' box labeled 'Net 7-vpc'. To its right, under 'Subredes (4)', are four subnets: 'us-east-1a', 'us-east-1b', 'us-east-1c', and 'us-east-1d'. Each subnet is associated with a specific CIDR range. To the right of the subnets is a 'Tablas de enrutamiento' (Routing tables) section with three entries: 'Net 7-rtb-public', 'Net 7-rtb-private1-us-east-1a', and 'Net 7-rtb-private2-us-east-1a'. Lines connect each subnet to its corresponding routing table entry.

Y, por último, podremos seleccionar el número de zonas de disponibilidad en las que crear Gateway, así como los puntos de enlace de la VPC.  
Finalmente, seleccionamos “Crear VPC”.

The screenshot shows the final steps of the AWS VPC creation wizard. On the left, under 'Personalizar bloques de CIDR de subredes' (Customize subnet CIDR blocks), the 'Gateway de S3' (S3 gateway) option is selected. Under 'Opciones de DNS' (DNS options), both 'Habilitar nombres de host DNS' (Enable host name DNS) and 'Habilitar la resolución de DNS' (Enable DNS resolution) are checked. At the bottom, there are 'Cancelar' (Cancel) and 'Crear VPC' (Create VPC) buttons, with 'Crear VPC' highlighted in red. On the right, the 'Vista previa' (Preview) section shows the completed VPC structure. It includes the 'VPC' box 'Net 7-vpc', 'Subredes (6)' with subnets 'us-east-1a' through 'us-east-1f', and 'Tablas de enrutamiento' with entries for each subnet. The 'Create VPC' button is prominently displayed at the bottom right of the preview area.

Comenzará el proceso de creación de la VPC.

VPC > Sus VPC > Crear VPC > Crear recursos de VPC

### Flujo de trabajo de creación de VPC

**Creación de recursos de VPC**  
Gracias por utilizar la nueva experiencia de creación de VPC. Díganos qué piensa al respecto.

Crear ruta

Detalles

- ✓ Crear VPC: vpc-0ebaeba8515fcc926
- ✓ Habilitar nombres de host DNS
- ✓ Habilitar la resolución de DNS
- ✓ Verificar la creación de una VPC: vpc-0ebaeba8515fcc926
- ✓ Crear punto de enlace de S3: vpce-06b6bdd41083a901b
- ✓ Crear subred: subnet-0ab8a7ac20300cec5
- ✓ Crear subred: subnet-01b668d150c484a49
- ✓ Crear subred: subnet-08cdb9066f9eb7a4e
- ✓ Crear subred: subnet-005ef7be023221586
- ✓ Crear subred: subnet-0c855f05e3620cf5f
- ✓ Crear subred: subnet-0fcc2bf79e835610a
- ✓ Crear una gateway de Internet: igw-00f3a4d225289f4be
- ✓ Adjuntar gateway de Internet a la VPC
- ✓ Crear tabla de enrutamiento: rtb-072d8d4acbb88fac4

44%

Una vez finalizado el proceso, nos devolverá a nuestro panel de VPC y nos aparecerá la información de nuestra VPC creada, con el nombre y los detalles de esta.

Panel de VPC

Vista global de EC2

Filtrar por VPC:

Seleccionar una VPC

Nube virtual privada

Sus VPC

Subredes

Tablas de enrutamiento

Puertas de enlace de Internet

Puerta de enlace de Internet de solo salida

Gateways de operador

Conjuntos de opciones de DHCP

Direcciones IP elásticas

Listas de prefijos

vpc-07131890db5abca55 / Net 7-vpc

Acciones

Detalles		Información	
ID de la VPC	vpc-07131890db5abca55	Estado	Available
Tenencia	Conjunto de opciones de DHCP	Nombres de host de DNS	Habilitado
Default	dopt-04c5c77b73fd61a4f	Tabla de enrutamiento principal	ACL de red principal
VPC predeterminada	CIDR IPv4	Grupo IPv6	acl-013051701bcab57d4
No	10.0.0.0/16	-	CIDR IPv6 (grupo de bordes de red)
Métricas de uso de direcciones de red	Grupos de reglas del firewall de DNS de Route 53 Resolver	ID de propietario	-
Desactivado	✗ No se pudieron cargar los grupos de reglas	680501345860	

Para comprobar las subredes que se han creado, simplemente en el panel de VPC, en “subredes” aparecerá la información de todas ellas.

Name	ID de subred	Estado	VPC	CIDR IPv4	CIDR IPv6
Net 7-subnet-p...	subnet-0305f32c83ce916b5	Available	vpc-07131890db5abca55   Ne...	10.0.16.0/20	-
Net 7-subnet-privat...	subnet-0701d15cab390c3f8	Available	vpc-07131890db5abca55   Ne...	10.0.128.0/20	-
proyecto-subnet-pr...	subnet-051fa12e117121cbd	Available	vpc-0a71383742f2fdidd   pro...	10.0.128.0/20	-
Net 7-subnet-publi...	subnet-0437fae38e204a851	Available	vpc-07131890db5abca55   Ne...	10.0.0.0/20	-
Net 7-subnet-privat...	subnet-0cdaea16f405258e	Available	vpc-07131890db5abca55   Ne...	10.0.144.0/20	-
proyecto-subnet-pr...	subnet-03ea7321e5b3912a0	Available	vpc-0a71383742f2fdidd   pro...	10.0.144.0/20	-

Una vez comprobadas las subredes, procedemos a configurar la tabla de enrutamiento, podemos acceder a ella mediante el panel de VPC “Tablas de enrutamiento”.

AWS crea automáticamente una tabla de ruteo asociada a la VPC. Puedes modificar esta tabla para definir cómo se enrutan los paquetes dentro de la red.

Finalmente, nos aseguraremos de agregar rutas para que el tráfico de nuestra VPC se pueda enrutar hacia el internet (si es necesario).

Deberemos realizar este proceso tanto para la red que hemos creado como para las subredes.

Name	ID de tabla de enruta...	Asociaciones de subred...	Asociaciones de ...	Prin...	VPC	ID d...
Net 7-rtb-private	rtb-0b07faf4303a3f61e	3 subredes	-	Sí	vpc-07131890db5abca55   Ne...	68050...
Net 7-rtb-public	rtb-0cb914eaf6b4dd989	3 subredes	-	No	vpc-07131890db5abca55   Ne...	68050...
proyecto-rtb-privat...	rtb-0d1bb6e9329b013a3	-	-	Sí	vpc-0a71383742f2fdidd   pro...	68050...

- **CLI (AWS CLI):**

Como en el caso de las instancias, también podremos utilizar la consola CLI que AWS nos proporciona para crear nuestra VPC y subred.

- Creamos la VPC con el bloque de direcciones IP 10.0.0.0/16:

```
aws ec2 create-vpc --cidr-block 10.0.0.0/16
```

- Creamos una subred en la zona de disponibilidad (us-east-1a) con el rango de IP 10.0.1.0/24:

```
aws ec2 create-subnet --vpc-id vpc-xxxxxx --cidr-block 10.0.1.0/24 --availability-zone us-east-1a
```

- Creamos una Gateway y la asociaremos:

```
aws ec2 create-internet-gateway
```

```
aws ec2 attach-internet-gateway --vpc-id vpc-xxxxxx --internet-gateway-id igw-xxxxxx
```

- Finalmente, actualizaremos la tabla de enrutamiento:

```
aws ec2 create-route --route-table-id rtb-xxxxxx --destination-cidr-block 0.0.0.0/0 --gateway-id igw-xxxxxx
```

- **IaC (CloudFormation o Terraform):**

Podremos utilizar la herramienta CloudFormation para automatizar la creación de una VPC y subredes con una plantilla YAML, exactamente igual que hicimos con las instancias.

Proporcionaremos la información de la VPC con un rango de direcciones 10.0.0.0/16.

Una subred en la zona us-east-1a con el rango de direcciones 10.0.1.0/24. Y, por último, una Gateway asociada a la VPC.

```
AWSTemplateFormatVersion: '2010-09-09'

Resources:
  MyVPC:
    Type: 'AWS::EC2::VPC'
    Properties:
      CidrBlock: '10.0.0.0/16'
      Tags:
        - Key: 'Name'
          Value: 'MiVPC'

  MySubnet:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref MyVPC
      CidrBlock: '10.0.1.0/24'
      AvailabilityZone: 'us-east-1a'
      Tags:
        - Key: 'Name'
          Value: 'MiSubnet'

  MyInternetGateway:
    Type: 'AWS::EC2::InternetGateway'
    Properties:
      Tags:
        - Key: 'Name'
          Value: 'MiInternetGateway'
```

En el caso de Terraform, el proceso sería similar, pero utilizaríamos un archivo .tf, similar al YAML que hemos mencionado arriba.

En este caso, esto es un pequeño fragmento del archivo que sería necesario, en el que se especifica el proveedor, la zona de disponibilidad, el código tanto para la creación de la VPC y de las subredes y al final la creación del Gateway. Como digo, es un pequeño fragmento y habría que añadir las tablas de enrutamiento y asociarlas.

```
# Proveedor AWS
provider "aws" {
  region = "us-east-1"
}

# Crear la VPC
resource "aws_vpc" "my_vpc" {
  cidr_block = "10.0.0.0/16"
  enable_dns_support = true
  enable_dns_hostnames = true

  tags = {
    Name = "MyVPC"
  }
}

# Crear una Subred Pública
resource "aws_subnet" "my_subnet_public" {
  vpc_id           = aws_vpc.my_vpc.id
  cidr_block       = "10.0.1.0/24"
  availability_zone = "us-east-1a"
  map_public_ip_on_launch = true

  tags = {
    Name = "MyPublicSubnet"
  }
}

# Crear una Subred Privada
resource "aws_subnet" "my_subnet_private" {
  vpc_id           = aws_vpc.my_vpc.id
  cidr_block       = "10.0.2.0/24"
  availability_zone = "us-east-1b"

  tags = {
    Name = "MyPrivateSubnet"
  }
}

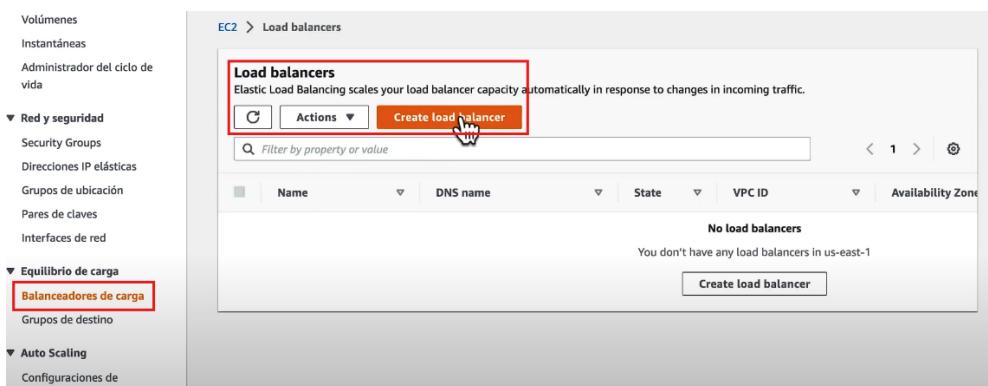
# Crear una Gateway de Internet
resource "aws_internet_gateway" "my_internet_gateway" {
```

## ■ Balanceador de Carga

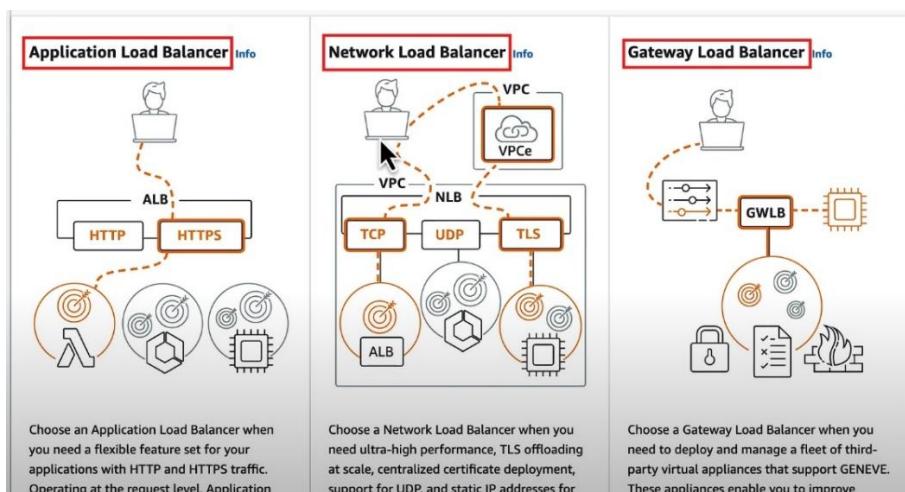
En AWS el balanceador de carga se denomina Elastic Load Balancing y su función es que distribuye automáticamente el tráfico entrante entre varios destinos, como EC2 instancias, contenedores y direcciones IP, en una o más zonas de disponibilidad. Monitorea el estado de los destinos registrados y enruta el tráfico solamente a destinos en buen estado. Elastic Load Balancing escala el equilibrador de carga a medida que el tráfico entrante va cambiando con el tiempo. Puede escalarse automáticamente para adaptarse a la mayoría de las cargas de trabajo.

- **Consola Gráfica (AWS Management Console):**

En primer lugar, en el apartado de EC2, accederemos al panel de equilibrio de carga, en el que aparecerá la opción “Balanceadores de carga”, lo seleccionamos y simplemente tendremos que clicar en la opción “Create load balancer”.



En este apartado, se nos mostrarán los diferentes tipos de balanceadores de carga, podremos elegir entre “Application” que está enfocado para aplicaciones HTTP/HTTPS, “Network” usado para tráfico TCP o UDP y “Gateway” enfocado para un tráfico más general. Seleccionamos en “crear” en el que sea más apropiado para nosotros.



Primeramente, le daremos un nombre a nuestro balanceador. También podremos escoger el tipo de esquema o tipo de IP que utilizaremos.

**Create Application Load Balancer** Info

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

▶ How Elastic Load balancing works

**Basic configuration**

**Load balancer name**  
Name must be unique within your AWS account and cannot be changed after the load balancer is created.  
  
A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

**Scheme** Info  
Scheme cannot be changed after the load balancer is created.  
 **Internet-facing**  
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. Learn more [?]

También podemos configurar las zonas de disponibilidad.

**Network mapping** Info

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

**VPC** Info  
Select the virtual private cloud (VPC) for your targets. Only VPCs with an internet gateway are enabled for selection. The selected VPC cannot be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#) [?].

-  
vpc-0c92e17a74fa2c786  
IPv4: 172.31.0.0/16

**Mappings** Info  
Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer, or the VPC are not available for selection.

us-east-1a (use1-az6)  
 us-east-1b (use1-az1)  
 us-east-1c (use1-az2)

En este punto, podemos configurar los grupos de seguridad que mencionamos anteriormente, así como los “listeners and routing”

**Security groups** Info

A security group is a set of firewall rules that control the traffic to your load balancer.

**Security groups**  
Select up to 5 security groups  
 [?]

default sg-0c9cd260f0d34283 X  
VPC vpc-0c92e17a74fa2c786

**Listeners and routing** Info

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener HTTP:80	Remove		
Protocol	Port	Default action	Info

Este punto es importante, pues aquí, deberemos crear nuestro grupo objetivo, es decir, agrupar nuestras instancias para que trabajen como un solo grupo objetivo y que así el balanceador pueda actuar de forma correcta.

**Listeners and routing** Info

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

**▼ Listener HTTP:80** Hand cursor icon Remove

Protocol	Port	Default action	Info
HTTP	: 80 1-65535	Forward to <span style="color: #800000;">Select a target group</span>	<span style="color: #800000;">C</span>
<span style="color: #800000;">Create target group</span>			

**Listener tags - optional**  
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add listener tag  
You can add up to 50 more tags.

Add listener

Una vez dentro de la configuración del grupo objetivo, como se puede observar, se pueden agrupar por instancias, direcciones IP, funciones lambda o por propios平衡adores de carga. En nuestro caso, los agruparemos por instancias. Simplemente habrá que añadir un nombre a ese grupo y procederemos al siguiente paso.

**Basic configuration**  
Settings in this section cannot be changed after the target group is created.

**Choose a target type**

Instances

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of Amazon EC2 Auto Scaling to manage and scale your EC2 capacity.

IP addresses

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

Lambda function

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

Application Load Balancer

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.

En los objetivos registrados, como se puede apreciar están disponibles dos instancias, en nuestro caso, serían cuatro, pero el proceso será el mismo.

Available instances (2)						
	Instance ID	Name	State	Security groups	Zone	Subnet ID
<input type="checkbox"/>	i-0bef895b21a3f43b0	Mi primera instancia	<span>running</span>	launch-wizard-1	us-east-1c	subnet-04ef741cb577ad9a6
<input type="checkbox"/>	i-07444cc0e557e21ed	Mi segunda instancia	<span>running</span>	launch-wizard-1	us-east-1c	subnet-04ef741cb577ad9a6

Seleccionaremos las instancias que queremos añadir al grupo objetivo y seleccionaremos “Include as pending below”.

Available instances (2/2)						
	Instance ID	Name	State	Security groups	Zone	Subnet ID
<input checked="" type="checkbox"/>	i-0bef895b21a3f43b0	Mi primera instancia	<span>running</span>	launch-wizard-1	us-east-1c	subnet-04ef741cb577ad9a6
<input checked="" type="checkbox"/>	i-07444cc0e557e21ed	Mi segunda instancia	<span>running</span>	launch-wizard-1	us-east-1c	subnet-04ef741cb577ad9a6

2 selected

Ports for the selected instances  
Ports for routing traffic to the selected instances.  
80  
1-65535 (separate multiple ports with commas)  
**Include as pending below**

Una vez creado el grupo objetivo, lo seleccionamos en el apartado de “listeners and routing”.

Listener HTTP:80			
Protocol	Port	Default action	
HTTP	: 80	Forward to <b>demo-tg-alb</b> Target type: Instance, IPv4	<span>Remove</span>
		<span>Create target group</span>	<span>C</span>

Listener tags - optional  
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add listener tag  
You can add up to 50 more tags.

Add listener

Se nos ofrecerá un resumen del balanceador con las características que hemos escogido y si todo es correcto procederemos a crearlo.

Summary			
Review and confirm your configurations. Estimate cost <a href="#">[?]</a>			
Basic configuration <a href="#">Edit</a>	Security groups <a href="#">Edit</a>	Network mapping <a href="#">Edit</a>	Listeners and routing <a href="#">Edit</a>
demo-alb • <a href="#">Internet-facing</a> • IPv4	• demo-sg-load-balancer sg-0c25eaa5e7bd0e803 <a href="#">[?]</a>	VPC <a href="#">vpc-0c92e17a74fa2c786</a> <a href="#">[?]</a> • us-east-1a subnet-081e0e659e665970e <a href="#">[?]</a> • us-east-1b subnet-0f239944e3d1fa078 <a href="#">[?]</a> • us-east-1c subnet-04ef741cb577ad9a6 <a href="#">[?]</a> • us-east-1d subnet-074196ac576deda6a <a href="#">[?]</a> • us-east-1e subnet-0c7021590b899402d <a href="#">[?]</a> • us-east-1f subnet-06d2d0d09ed41bd75 <a href="#">[?]</a>	• HTTP:80 defaults to <a href="#">demo-tg-alb</a> <a href="#">[?]</a>

- **CLI (AWS CLI):**

Como en el caso de las instancias, también podremos utilizar la consola CLI que AWS nos proporciona para crear nuestro balanceador de carga.

- Para crear un balanceador de carga, en este caso, del tipo Network, utilizaremos este comando:

```
aws elbv2 create-load-balancer --name my-network-lb \
--subnets subnet-xxxxxxxx subnet-yyyyyyy \
--security-groups sg-xxxxxxxx \
--scheme internet-facing \
--load-balancer-type network \
--listeners Protocol=TCP,Port=80,DefaultActions=[Type=fixed-
response,FixedResponseConfig={StatusCode=200,MessageBody="OK",C
ontentType="text/plain"}]
```

- El siguiente paso será crear el grupo objetivo para el balanceador:

```
aws elbv2 create-target-group --name my-targets \
--protocol TCP --port 80 \
--vpc-id vpc-xxxxxxxx --target-type instance
```

- Una vez creado el grupo objetivo, registraremos las instancias EC2:

```
aws elbv2 register-targets --target-group-arn
arn:aws:elasticloadbalancing:region:account-id:targetgroup/my-
targets/xxxxxxxx \
--targets Id=i-xxxxxxxx Id=i-yyyyyyy
```

- Finalmente, crearemos un listener y lo asociaremos al balanceador:

```
aws elbv2 create-listener --load-balancer-arn
arn:aws:elasticloadbalancing:region:account-id:loadbalancer/net/my-
network-lb/xxxxxxxx \
--protocol TCP --port 80 \
--default-actions
Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:region:acc
ount-id:targetgroup/my-targets/xxxxxxxx
```

- **IaC (CloudFormation o Terraform):**

Para crear un balanceador de carga, también podemos hacerlo mediante CloudFormation, en este caso, utilizaremos una plantilla en formato YAML. Como se puede observar, aparece la información sobre la VPC, la subred, el grupo de seguridad, el grupo objetivo y el balanceador que se va a crear.

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  MyVPC:
    Type: 'AWS::EC2::VPC'
    Properties:
      CidrBlock: '10.0.0.0/16'

  MySubnet:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref MyVPC
      CidrBlock: '10.0.1.0/24'
      AvailabilityZone: 'us-east-1a'

  MySecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: 'Allow TCP traffic'
      VpcId: !Ref MyVPC
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: '80'
          ToPort: '80'
          CidrIp: '0.0.0.0/0'

  MyTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: 'my-targets'
      Protocol: 'TCP'
      Port: '80'
      VpcId: !Ref MyVPC
      TargetType: 'instance'

  MyNetworkLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: 'MyNetworkLoadBalancer'
      Subnets:
        - Ref: MySubnet
```

En el caso de Terraform, el código que se utiliza es bastante similar e intuitivo.

Como se puede apreciar aparece el proveedor, la zona de disponibilidad, los datos del balanceador de carga, el grupo objetivo, los listener, etc.

```
provider "aws" {
    region = "us-east-1"
}

# Crear un Network Load Balancer
resource "aws_lb" "my_nlb" {
    name          = "my-network-lb"
    internal      = false
    load_balancer_type = "network"
    security_groups = [aws_security_group.my_sg.id]
    subnets        = [aws_subnet.my_subnet.id]

    enable_deletion_protection = false
    enable_cross_zone_load_balancing = true

    tags = {
        Name = "MyNetworkLB"
    }
}

# Crear un Target Group para el NLB
resource "aws_lb_target_group" "my_target_group" {
    name      = "my-targets"
    port      = 80
    protocol = "TCP"
    vpc_id   = aws_vpc.my_vpc.id
    target_type = "instance"
}

# Crear un Listener para el NLB
resource "aws_lb_listener" "my_listener" {
    load_balancer_arn = aws_lb.my_nlb.arn
    port            = 80
    protocol        = "TCP"

    default_action {
        type = "forward"
        target_group_arn = aws_lb_target_group.my_target_group.arn
    }
}
```

## 3.2. AZURE

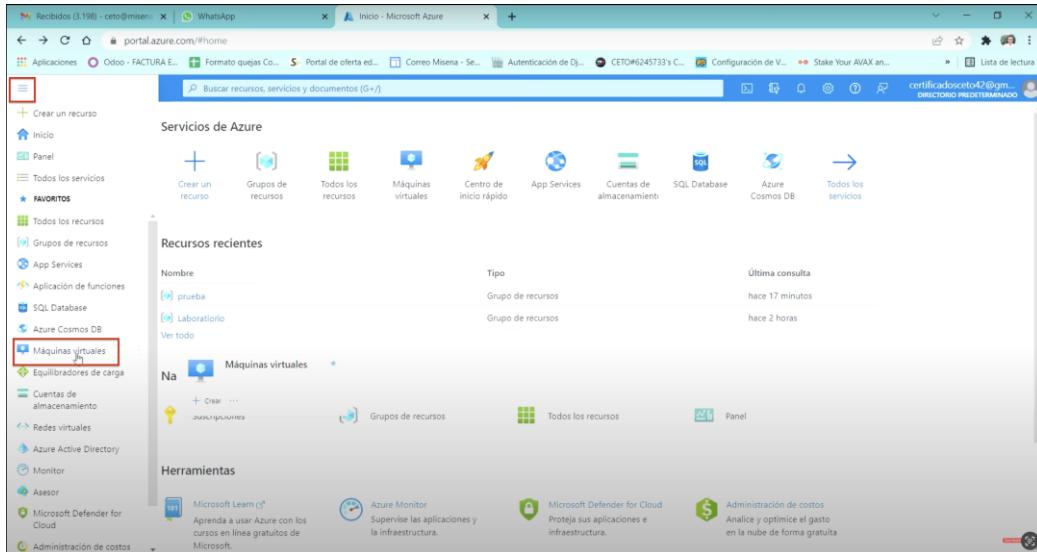
### 3.2.1. Creación de recursos en Azure

## ■ Máquinas virtuales

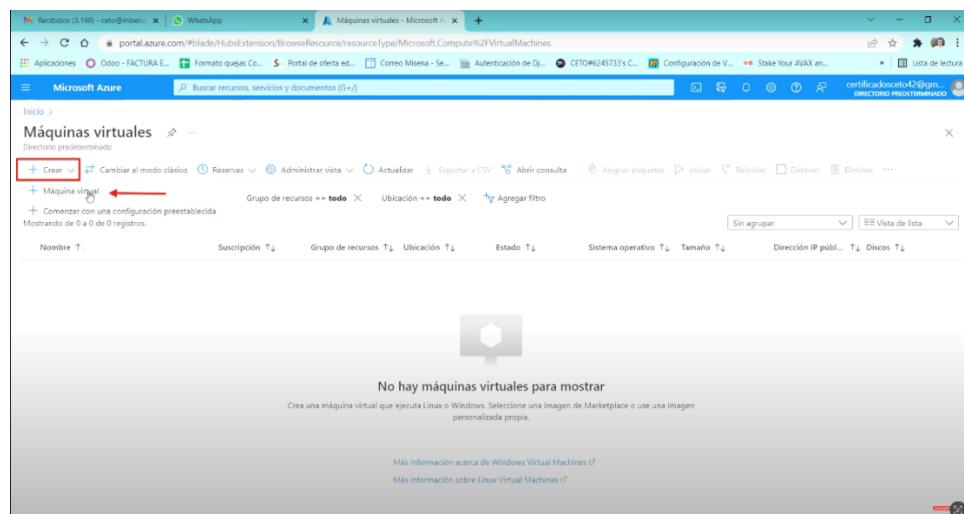
Es un recurso informático que emula una computadora física, pero que funciona dentro de la infraestructura de la nube de Azure. Es un entorno virtualizado que ejecuta sistemas operativos (como Windows, Linux, etc.)

- **Consola Gráfica (Azure Management Portal):**

En primer lugar, accederemos a el pequeño recuadro que tenemos en la parte superior de la izquierda y se nos abrirá un desplegable donde vamos a seleccionar “máquinas virtuales”.

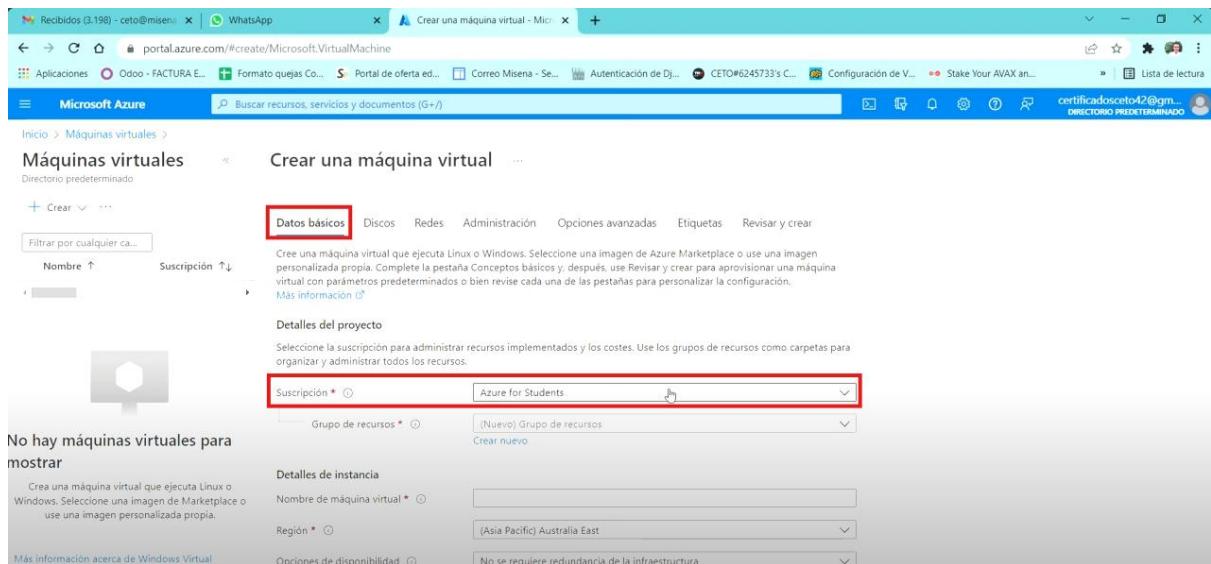


Como podemos ver, no tenemos ninguna máquina virtual creada, para empezar a crear una nueva nos iremos a la parte superior de la izquierda a donde dice “crear”. Nos saldrán dos opciones, nosotros la vamos a crear de cero, así que elegiremos “máquina virtual”.



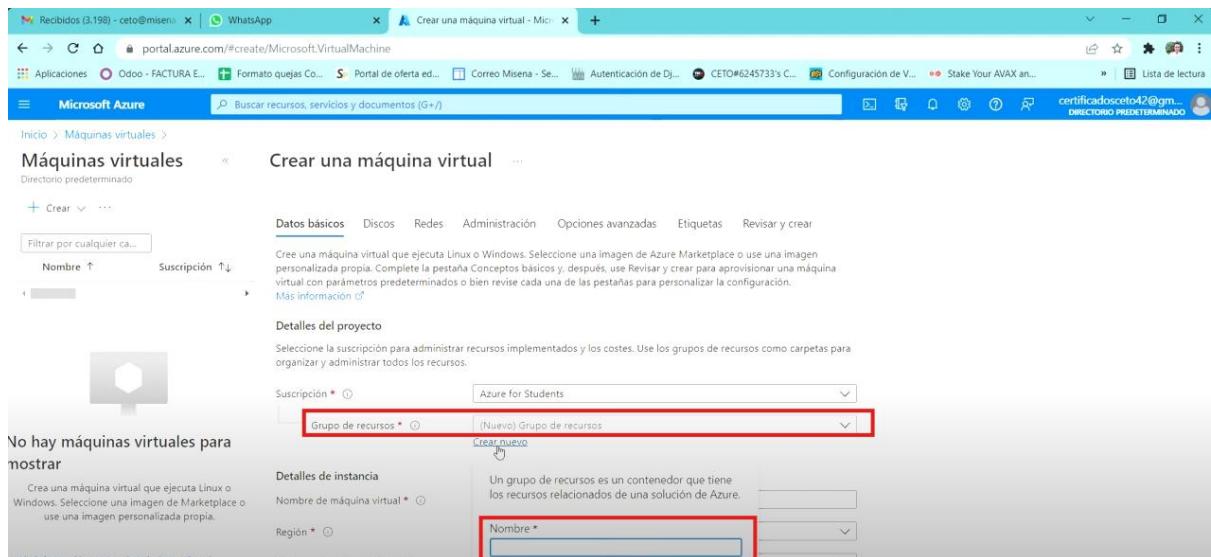
## COMPARATIVA AWS Y AZURE

Nos dirigirá a “datos básicos” donde veremos la suscripción que tenemos en Azure.



The screenshot shows the 'Create a virtual machine' wizard in the Microsoft Azure portal. The 'Basic' tab is selected. In the 'Subscription' dropdown, 'Azure for Students' is chosen and highlighted with a red box. Below it, the 'Resource group' dropdown shows '(Nuevo) Grupo de recursos' with a 'Create new' link. The rest of the form includes fields for 'Name', 'Region', and 'Availability options'.

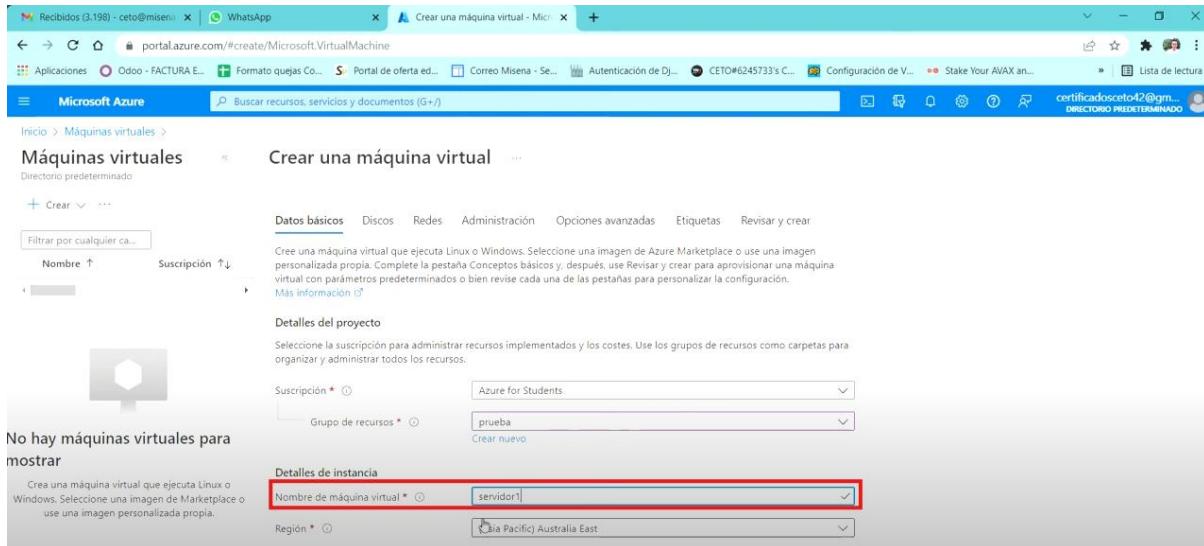
Si no tenemos un “grupo de recursos” procederemos a crear uno donde dice “crear nuevo”. Le daremos el nombre que queramos y le damos a “aceptar”. El grupo de recursos es donde podemos meter las máquinas virtuales que nosotros deseemos con unas configuraciones que establezcamos.



The screenshot shows the 'Create a virtual machine' wizard in the Microsoft Azure portal. The 'Basic' tab is selected. The 'Resource group' dropdown is highlighted with a red box, showing '(Nuevo) Grupo de recursos'. Below it, the 'Name' input field is also highlighted with a red box. The rest of the form includes fields for 'Region' and 'Availability options'.

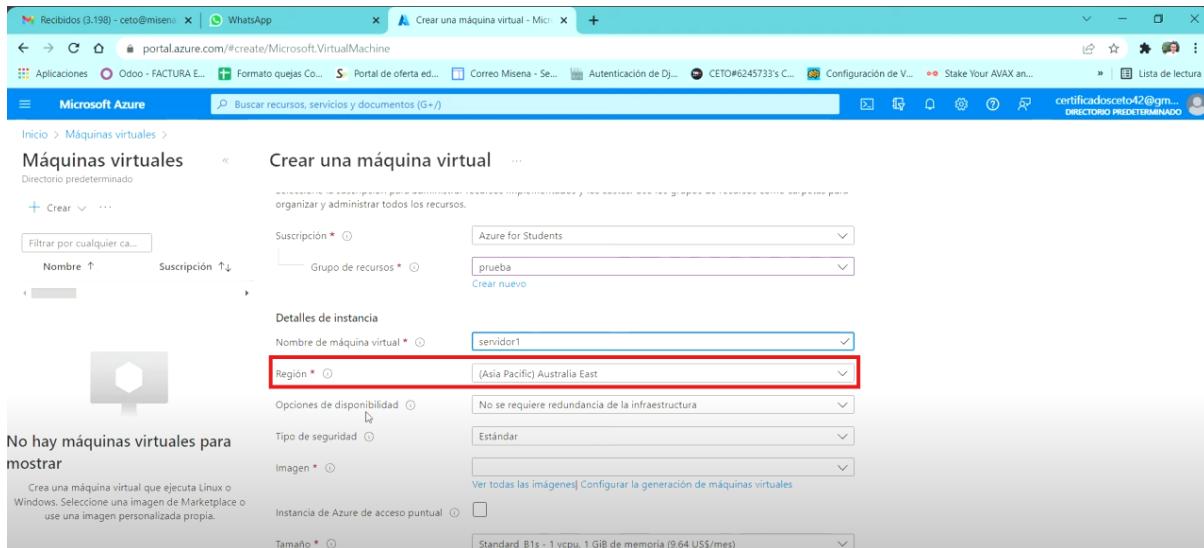
## COMPARATIVA AWS Y AZURE

En este caso se ha creado un grupo “prueba” donde se alojarán nuestras máquinas Linux. Ahora vamos a escribir el nombre que le vamos a dar a nuestra máquina virtual, en nuestro caso será “servidor1”.



The screenshot shows the Microsoft Azure portal interface for creating a new virtual machine. The top navigation bar includes links for Recibidos, WhatsApp, and the current page, 'Crear una máquina virtual - Microsoft Virtual Machine'. Below the navigation is a toolbar with icons for Applications, Odoo, FACTURA E..., etc. The main content area has a breadcrumb trail: Inicio > Máquinas virtuales. A sidebar on the left lists 'Máquinas virtuales' and 'Directorio predeterminado'. The main form is titled 'Crear una máquina virtual' and has tabs for Datos básicos, Discos, Redes, Administración, Opciones avanzadas, Etiquetas, and Revisar y crear. The 'Datos básicos' tab is active. It contains instructions to create a Linux or Windows VM using a Marketplace image or a personal image. Below this is the 'Detalles del proyecto' section, which asks for a subscription and resource group. The 'Subscription' dropdown is set to 'Azure for Students' and the 'Resource group' dropdown is set to 'prueba'. The 'Detalles de instancia' section is where the name and region are specified. The 'Name' field is filled with 'servidor1' and the 'Region' dropdown is set to '(Asia Pacific) Australia East'. Both the 'Name' and 'Region' fields are highlighted with a red border.

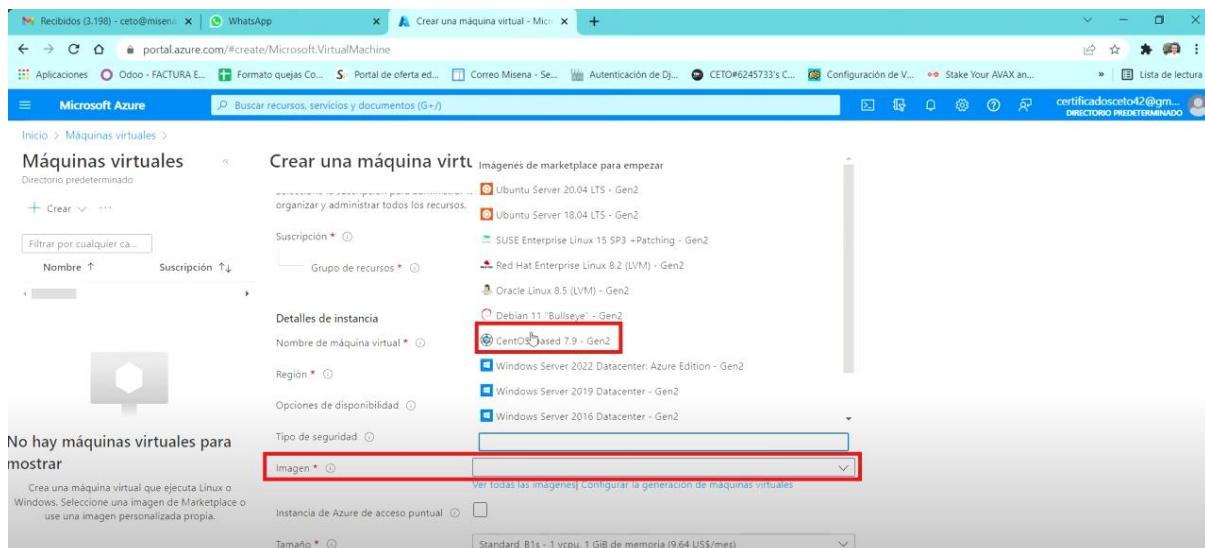
Ahora nos vamos a “región” donde escogeremos en que ubicación se encontrara nuestro servidor. Dependiendo de la zona varía el precio, en este caso lo dejaremos en Asia.



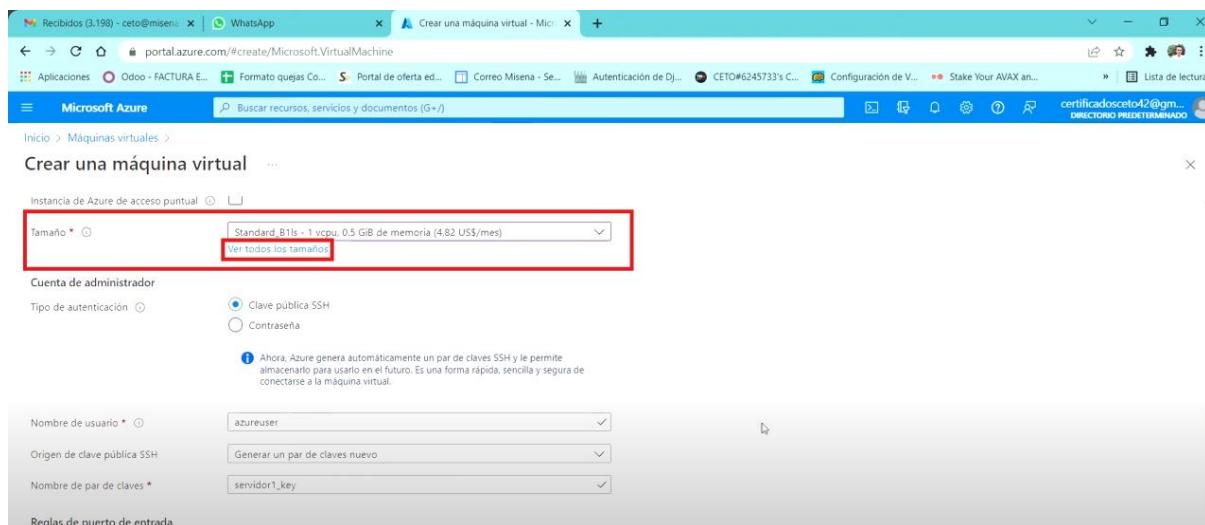
This screenshot shows the same 'Create a virtual machine' wizard as the previous one, but with a different focus. The 'Region' dropdown in the 'Instancia' section is highlighted with a red border. It is set to '(Asia Pacific) Australia East'. The other fields shown are 'Nombre' (servidor1), 'Opciones de disponibilidad' (No se requiere redundancia de la infraestructura), 'Tipo de seguridad' (Estándar), 'Imagen' (Ver todas las imágenes / Configurar la generación de máquinas virtuales), and 'Tamaño' (Standard\_B1s - 1 vcpu, 1 GiB de memoria (9.64 US\$/mes)). The rest of the interface is identical to the first screenshot, including the sidebar and top navigation.

## COMPARATIVA AWS Y AZURE

Si nos vamos a “imagen” nos sale un desplegable con muchas opciones de sistemas operativos que podremos darle a nuestra máquina virtual. Escogeremos la de CentOS (Linux).



En “tamaño” es donde podremos elegir el número de CPU, RAM etc.... que tendrá nuestra máquina virtual. Si nos fijamos más abajo pone “ver todos los tamaños”, si le damos clic ahí nos saldrán muchas más opciones.



## COMPARATIVA AWS Y AZURE

Nos muestra más opciones con su respectivo precio. En este caso vamos a seleccionar la de 4,82 dólares y daremos sonde pone “seleccionar” en la parte inferior de la izquierda.

Tamaño de VM ↑	Familia ↑	vCPU ↑	RAM (GiB) ↑	Discos de datos ↑	E/S máxima por s. ↑	Almacenamiento tem. ↑	Disco premium ↑	Costo/mes ↑
DS1_v2 ↗	Uso general	1	3.5	4	3200	7	Se admite	61.32 US\$
D2s_v3 ↗	Uso general	2	8	4	3200	16	Se admite	172.25 US\$
B2s ↗	Uso general	2	4	4	1280	8	Se admite	38.54 US\$
B1s ↗	Uso general	1	1	2	320	4	Se admite	9.64 US\$
B2ms ↗	Uso general	2	8	4	1920	16	Se admite	77.38 US\$
B1ls ↗	Uso general	1	0.5	2	160	4	Se admite	4.82 US\$
DS2_v2 ↗	Uso general	2	7	8	6400	14	Se admite	122.64 US\$
B4ms ↗	Uso general	4	16	8	2880	32	Se admite	154.03 US\$
D4s_v3 ↗	Uso general	4	16	8	6400	32	Se admite	182.50 US\$

“Tipo de autentificación” es la conexión que estableceremos. La clave SSH es la más segura, pero para el ejemplo y que sea algo más rápido usaremos “contraseña” para acceder a nuestra máquina virtual.

**Cuenta de administrador**

Tipo de autenticación:  Contraseña

Nombre de usuario \*: cliente1

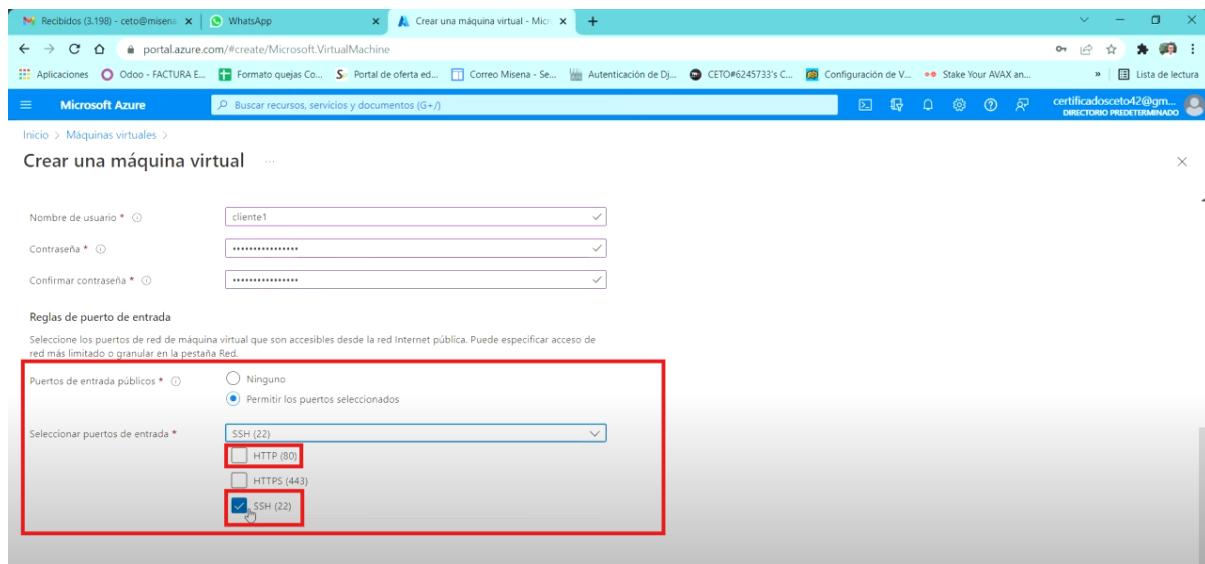
Contraseña \*: \*\*\*\*\*

Confirmar contraseña \*: \*\*\*\*\*

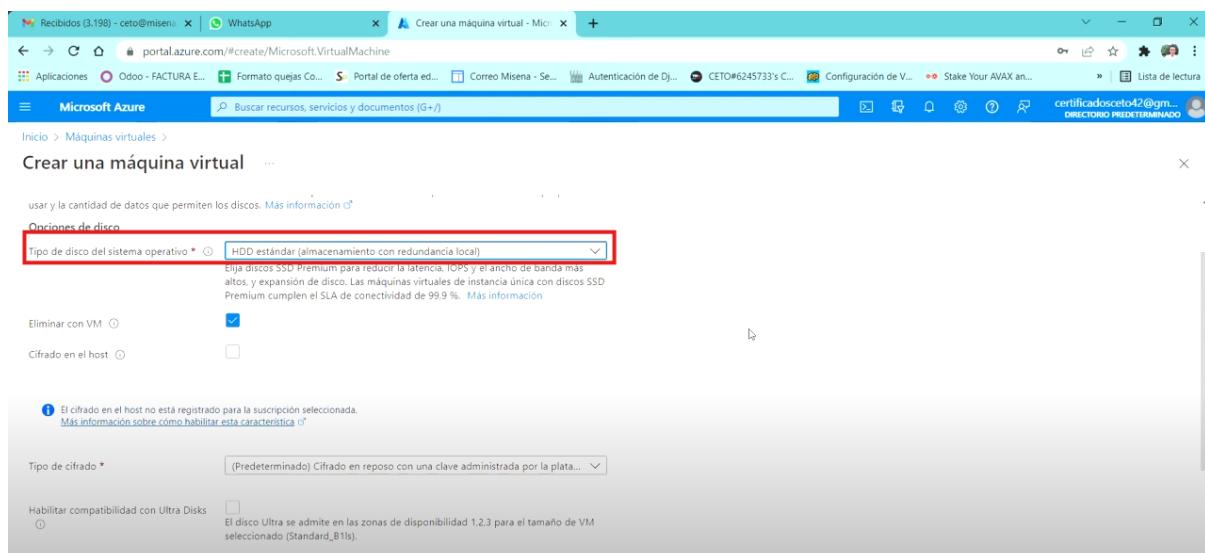
Nota: La contraseña y la confirmación de la contraseña deben coincidir.

## COMPARATIVA AWS Y AZURE

“Seleccionar puertos de entrada” vamos a desplegar las opciones y elegiremos “SSH” y “HTTP” con sus respectivos puertos (80 y 22). Damos a “siguiente: discos >”.

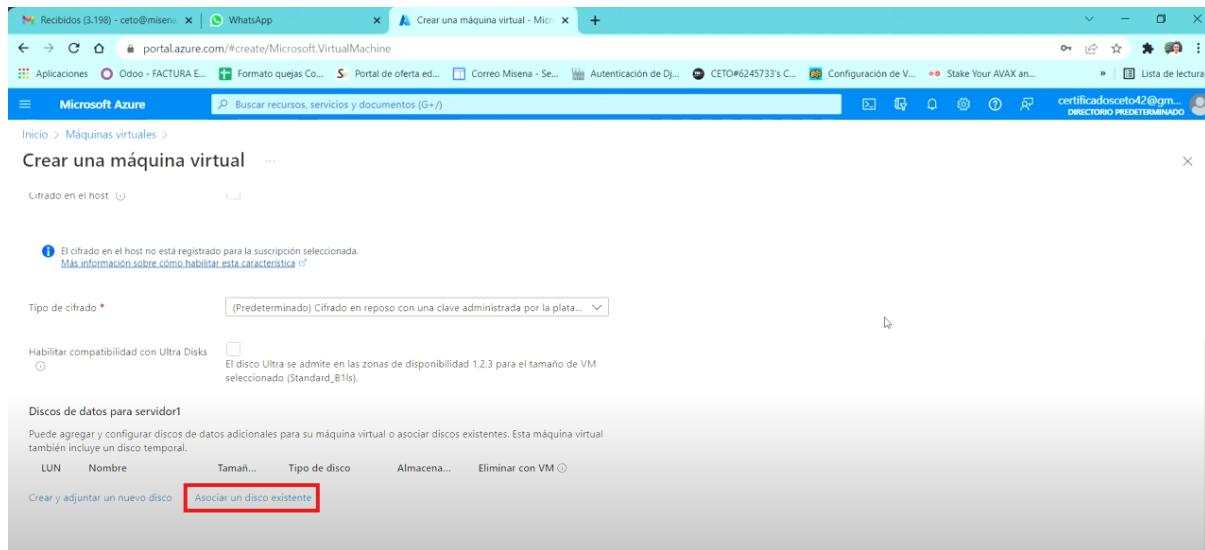


“Tipo del disco del sistema” donde tendremos un desplegable con varias opciones. En nuestro caso vamos a coger un disco duro HDD ya que es para una práctica pequeña y no se va a usar para grandes proyectos.



## COMPARATIVA AWS Y AZURE

Si por ejemplo se quiere utilizar algún disco duro que posea su máquina física para el servidor que se está creando, tendrá que elegir la opción “asociar un disco existente”. Damos a “siguiente: redes >”.



- **CLI:**

Este es el código para crear varias máquinas virtuales (VM) en **Azure** utilizando la **CLI de Azure**. El bucle for permite crear múltiples instancias de máquinas virtuales de forma automatizada.

Cada máquina virtual tendrá las siguientes características:

- **Nombre único:** MiVM1, MiVM2, MiVM3, MiVM4.
- **Sistema operativo:** Ubuntu LTS.
- **Tamaño de VM:** Standard\_B1s (pequeño, adecuado para cargas ligeras).
- **Red y subred:** Se conectan a la red virtual **MiRedVirtual** y la subred **MiSubred**.
- **Sin IP pública:** No tendrán direcciones IP públicas asignadas.
- **Sin NSG asignada:** No se aplican reglas de seguridad a las máquinas virtuales.
- **Acceso SSH:** Se genera un par de claves SSH para acceder de forma segura.

```
for i in {1..4}
do
az vm create \
--resource-group MiGrupoDeRecursos \
--name MiVM$i \
--image UbuntuLTS \
--size Standard_B1s \
--vnet-name MiRedVirtual \
--subnet MiSubred \
--public-ip-address "" \
--nsg "" \
--admin-username azureuser \
--generate-ssh-keys
done
```

- **IaC (Terraform o ARM Templates):**

Este código de Terraform crea tanto máquinas virtuales Linux como sus interfaces de red (NIC) asociadas en Azure.

- Se crean 4 máquinas virtuales, cada una con una interfaz de red asociada, utilizando **Ubuntu 20.04 LTS** como imagen.
- Cada VM tiene un nombre único, se crea en la región **East US**, con el tamaño **Standard\_B1s**, y se les asigna la misma red virtual/subred.
- Se crean 4 interfaces de red (una para cada VM). Cada NIC tiene una configuración IP dinámica dentro de la subred especificada.

```

resource "azurerm_linux_virtual_machine" "mi_vm" {
  count          = 4
  name           = "MiVM${count.index + 1}"
  resource_group_name = azurerm_resource_group.mi_grupo.name
  location        = "East US"
  size            = "Standard_Bis"
  admin_username   = "azureuser"
  network_interface_ids = [
    azurerm_network_interface.mi_nic[count.index].id,
  ]
  os_disk {
    caching          = "ReadWrite"
    storage_account_type = "Standard_LRS"
  }
  source_image_reference {
    publisher       = "Canonical"
    offer           = "UbuntuServer"
    sku             = "20.04-LTS"
    version         = "latest"
  }
  tags = {
    environment = "production"
  }
}

resource "azurerm_network_interface" "mi_nic" {
  count          = 4
  name           = "MINIC${count.index + 1}"
  location        = "East US"
  resource_group_name = azurerm_resource_group.mi_grupo.name
  ip_configuration {
    name                  = "MiIPConfig${count.index + 1}"
    subnet_id            = azurerm_subnet.mi_subnet.id
    private_ip_address_allocation = "Dynamic"
  }
}

```

Este código es de un archivo JSON de utilizado para desplegar y configurar una máquina virtual en Azure.

- Primero se especifica que el recurso que se está creando es una **máquina virtual**.
- Indica la versión de la API de Azure que se está utilizando para crear el recurso.
- Define la ubicación geográfica en la que se desplegará la máquina virtual.
- En “properties” se especifican las propiedades de la máquina virtual, tanto de hardware, software, credenciales y de red.

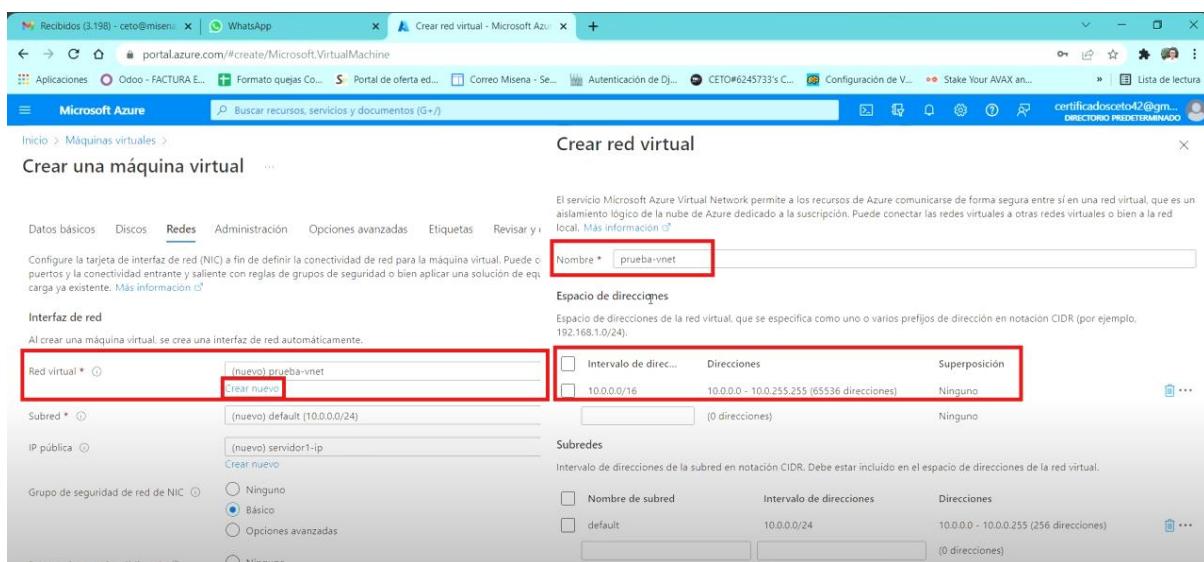
```
{  
    "type": "Microsoft.Compute/virtualMachines",  
    "apiVersion": "2021-03-01",  
    "location": "eastus",  
    "properties": {  
        "hardwareProfile": {  
            "vmSize": "Standard_B1s"  
        },  
        "osProfile": {  
            "computerName": "MiVM1",  
            "adminUsername": "azureuser",  
            "adminPassword": "P@ssw0rd!"  
        },  
        "networkProfile": {  
            "networkInterfaces": [  
                {  
                    "id": "[resourceId('Microsoft.Network/networkInterfaces', 'MiNIC1')]"  
                }  
            ]  
        }  
    }  
}
```

## ■ Red virtual y Subredes

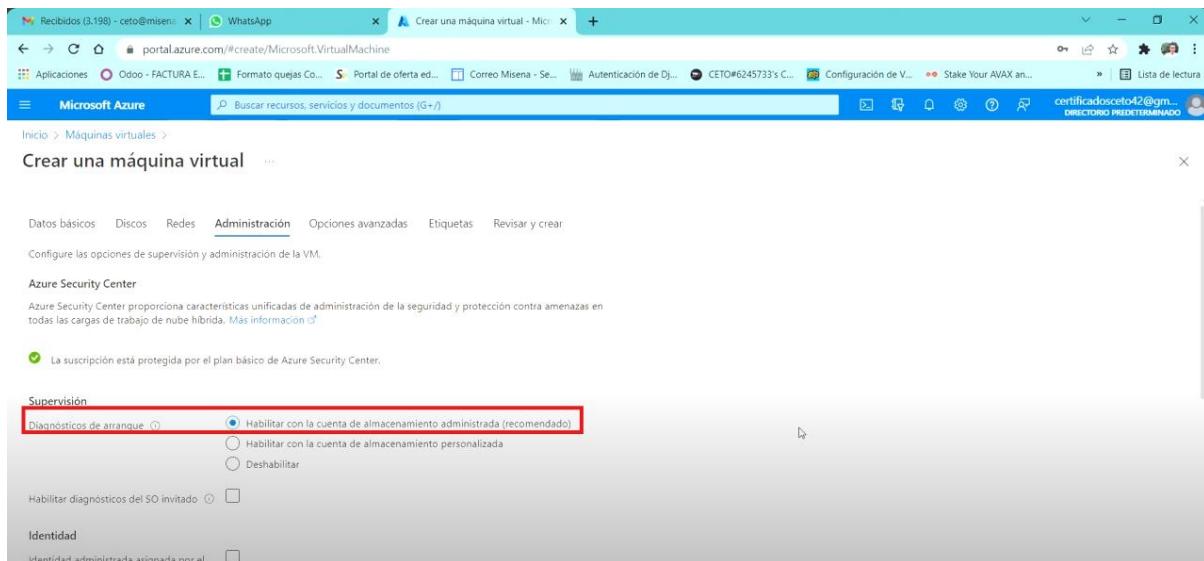
Es una red privada, definida por el usuario, que permite conectar de manera segura recursos en la nube, como máquinas virtuales, aplicaciones y otros servicios.

- **Consola Gráfica (AZURE):**

“Red virtual” --> “crear nuevo”, donde podremos crear nuestra red virtual dándole un nombre (ya sea un departamento, laboratorio...) y un intervalo de direcciones IP (que asignaremos a ese departamento, laboratorio...). Damos a “siguiente: administración >”.

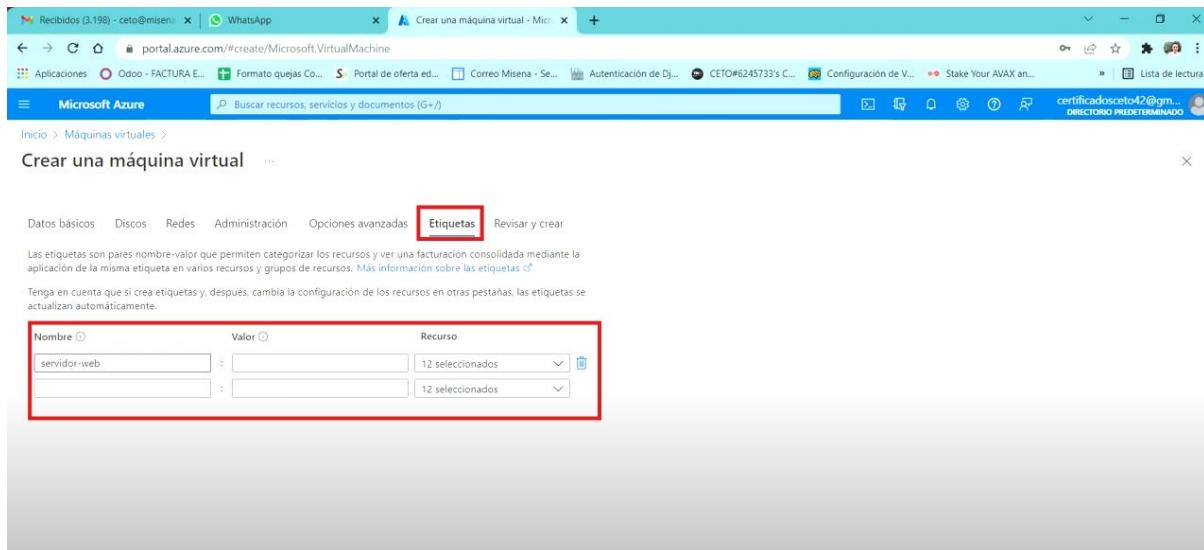


En la parte de “administración” nos vamos a “diagnóstico de arranque” y vemos que esté en “habilitar con la cuenta de almacenamiento administrada (recomendado)”. Y no tocaremos nada más. Nos vamos a “etiquetas” (situado en la parte superior).

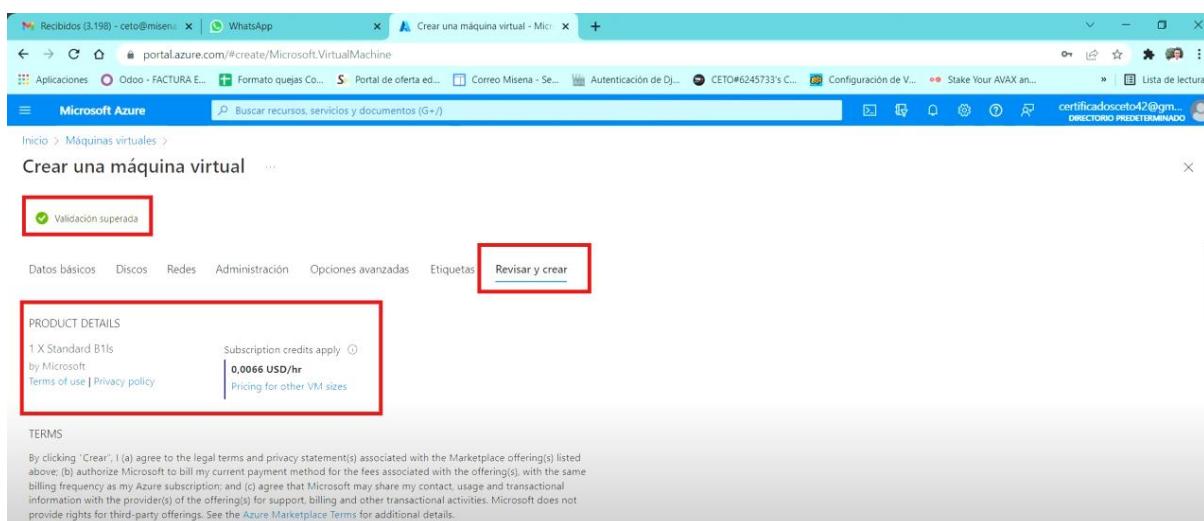


## COMPARATIVA AWS Y AZURE

“Nombre” donde indicaremos el nombre que deseamos darle al servidor. En este caso como va a ser uno dedicado a web se le llamará “servidor-web”. Se le ponen etiquetas por si tenemos más de uno, y así poder distinguirlos para saber cuál hace su función. Vamos a “siguiente: revisar y crear”.



Vemos que se ha validado correctamente sin ningún tipo de error. Muestra un pequeño resumen de la configuración del servidor con su precio. Nos dirigimos a “crear” (en la parte de abajo).



## COMPARATIVA AWS Y AZURE

Esperamos a que se complete.

The screenshot shows the Microsoft Azure portal with the URL <https://portal.azure.com/#blade/HubsExtension/DeploymentDetailsBlade/overview/id/%2Fsubscriptions%2F072a005c-0aaf-45ee-a9f7-ba827c034001%2FresourceGroups%2Fprueba%2Fprovider...>. The page title is 'CreateVm-OpenLogic.CentOS-7\_9-gen2-20220312153643 | Información general'. A red box highlights the message 'La implementación está en curso' (The implementation is in progress). Below this, there is a table showing the details of the implementation:

Recurso	Tipo	Estado	Detalles de la operación
servidor1	Microsoft.Compute/virtualMachines	Created	Detalles de la operación
servidor1808	Microsoft.Network/networkInterfaces	Created	Detalles de la operación
servidor1-ip	Microsoft.Network/publicIPAddresses	OK	Detalles de la operación
servidor1-nsg	Microsoft.Network/networkSecurityGroups	OK	Detalles de la operación
prueba-vnet	Microsoft.Network/virtualNetworks	OK	Detalles de la operación

Se ha completado correctamente. Ahora desde este menú podemos elegir si queremos “crear otra MV” donde podremos crear otras 3 máquinas virtuales más para diferentes usos o “ir al recurso” para ver el que hemos creado.

The screenshot shows the Microsoft Azure portal with the URL <https://portal.azure.com/#blade/HubsExtension/DeploymentDetailsBlade/overview/id/%2Fsubscriptions%2F072a005c-0aaf-45ee-a9f7-ba827c034001%2FresourceGroups%2Fprueba%2Fprovider...>. The page title is 'CreateVm-OpenLogic.CentOS-7\_9-gen2-20220312153643 | Información general'. A red box highlights the message 'Se completó la implementación' (The implementation completed). Below this, there are two buttons: 'Ir al recurso' (Go to resource) and 'Crear otra VM' (Create another VM), with 'Crear otra VM' being highlighted by a red box.

## COMPARATIVA AWS Y AZURE

Si vamos al recurso vemos nuestra máquina virtual creada correctamente con las configuraciones que hemos establecido. Arriba hay una opción “iniciar” donde se encenderá la máquina virtual creada.

The screenshot shows the Microsoft Azure portal interface. At the top, there are several tabs and a search bar. On the left, a sidebar lists options like 'Información general', 'Control de acceso (IAM)', 'Etiquetas', 'Diagnosticar y solucionar problemas', 'Configuración' (with 'Redes' selected), and 'Propiedades'. The main content area displays the properties of a VM named 'servidor1'. A red box highlights the 'Información esencial' section, which includes fields for 'Grupo de recursos', 'Estado', 'Ubicación', 'Suscripción', 'Id. de suscripción', and 'Etiquetas'. Another red box highlights the 'Máquina virtual' section under 'Propiedades', showing details like 'Nombre del equipo', 'Estado de mantenimiento', 'Sistema operativo', 'Publicador', 'Oferta', and 'Redes'. The 'Redes' section shows the VM's IP addresses and network configuration.

- **CLI:**

Como en el caso de las instancias, también podemos utilizar la consola CLI que AZURE nos proporciona para crear nuestra red y subred, mediante los siguientes comandos:

- **az** es el comando de la CLI de Azure y **network vnet create** especifica que estamos creando una VNet (Red Virtual).
- Se define el nombre de la red virtual que estamos creando. Este nombre debe ser único dentro de tu suscripción de Azure.
- Especifica el grupo de recursos donde se creará la VNet.
- Define la región geográfica en la que se creará la red virtual.
- Especifica el rango de direcciones IP para la red virtual.
- Define el nombre de la subred que se creará dentro de la red virtual.
- Especifica el rango de direcciones IP para la subred que estamos creando.

```
az network vnet create \
--name MiRedVirtual \
--resource-group MiGrupoDeRecursos \
--location eastus \
--address-prefix 10.0.0.0/16 \
--subnet-name MiSubred \
--subnet-prefix 10.0.1.0/24
```

- **IaC (Terraform o ARM Template):**

En el caso de Terraform, esto es un pequeño fragmento del archivo que sería necesario, en el que se especifica el nombre interno del recurso, la ubicación donde se creará la red, el grupo de recursos en el que se encontrará la red, y el rango de direcciones que la red utilizará. Además, definiremos en el bloque subnet el nombre de nuestra subred y el rango de direcciones de la subred.

```
resource "azurerm_virtual_network" "mi_red" {
  name                = "MiRedVirtual"
  location            = "East US"
  resource_group_name = azurerm_resource_group.mi_grupo.name
  address_space        = ["10.0.0.0/16"]

  subnet {
    name          = "MiSubred"
    address_prefix = "10.0.1.0/24"
  }
}
```

En el caso de ARM Templates, Este es un archivo JSON que se utiliza para describir los recursos que vamos a crear. Se especifica el tipo de recurso que estás creando, la información sobre la API que se utilizará, la ubicación donde se creará la red, y propiedades específicas de la red como los rangos de direcciones que se asignan a la red. A continuación, definimos el nombre de la/s subred/es y el rango de direcciones que tendrá la subred.

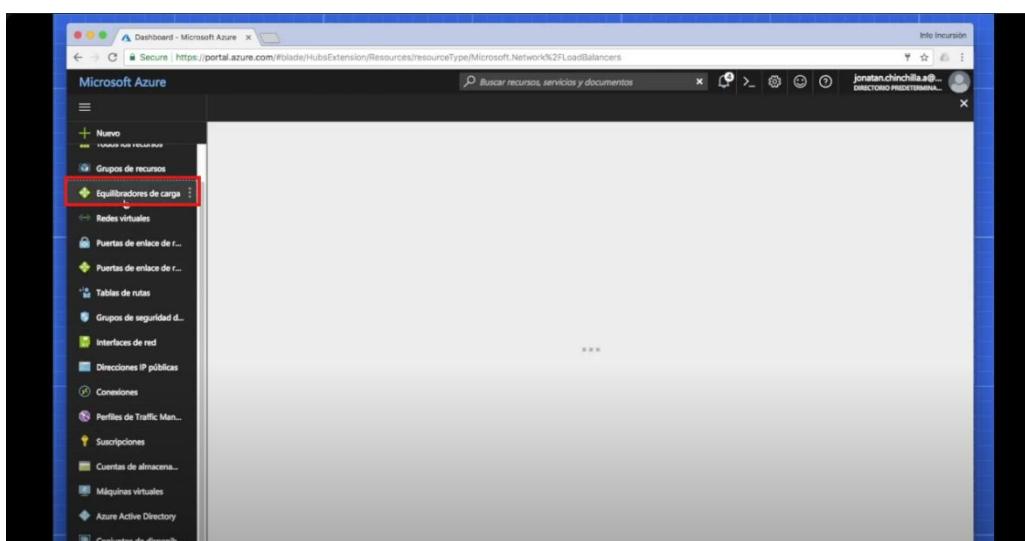
```
{
  "type": "Microsoft.Network/virtualNetworks",
  "apiVersion": "2021-03-01",
  "location": "eastus",
  "properties": {
    "addressSpace": {
      "addressPrefixes": [
        "10.0.0.0/16"
      ]
    },
    "subnets": [
      {
        "name": "MiSubred",
        "properties": {
          "addressPrefix": "10.0.1.0/24"
        }
      }
    ]
  }
}
```

## ■ Balanceador de carga

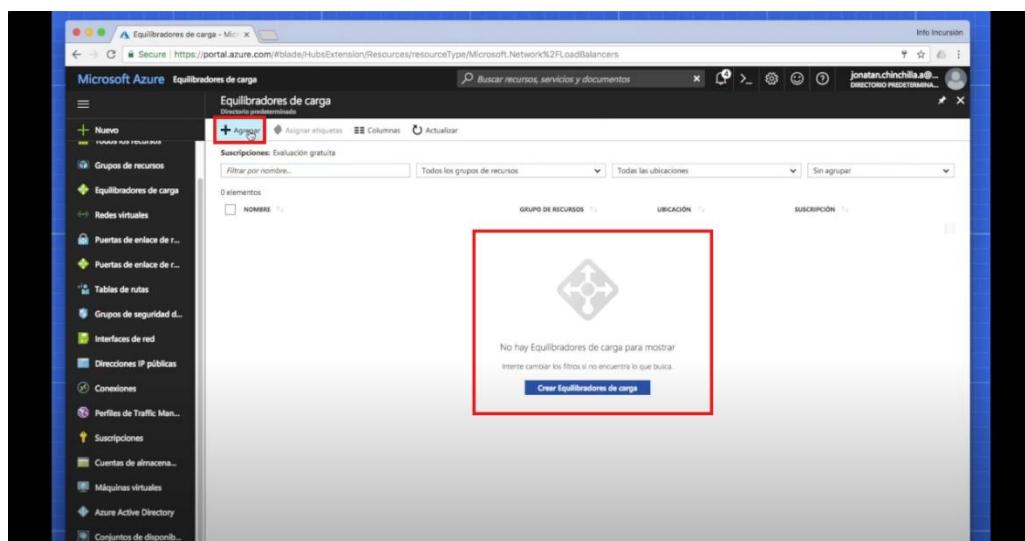
Es un servicio que distribuye el tráfico de red entrante entre varias instancias de máquinas virtuales o recursos, con el fin de optimizar el rendimiento, mejorar la disponibilidad y garantizar la resiliencia de las aplicaciones. Al distribuir la carga de manera eficiente, el balanceador de carga ayuda a evitar que algunos recursos se vean sobrecargados, mejorando así la escalabilidad y la confiabilidad de las aplicaciones.

- **Consola Gráfica (AZURE):**

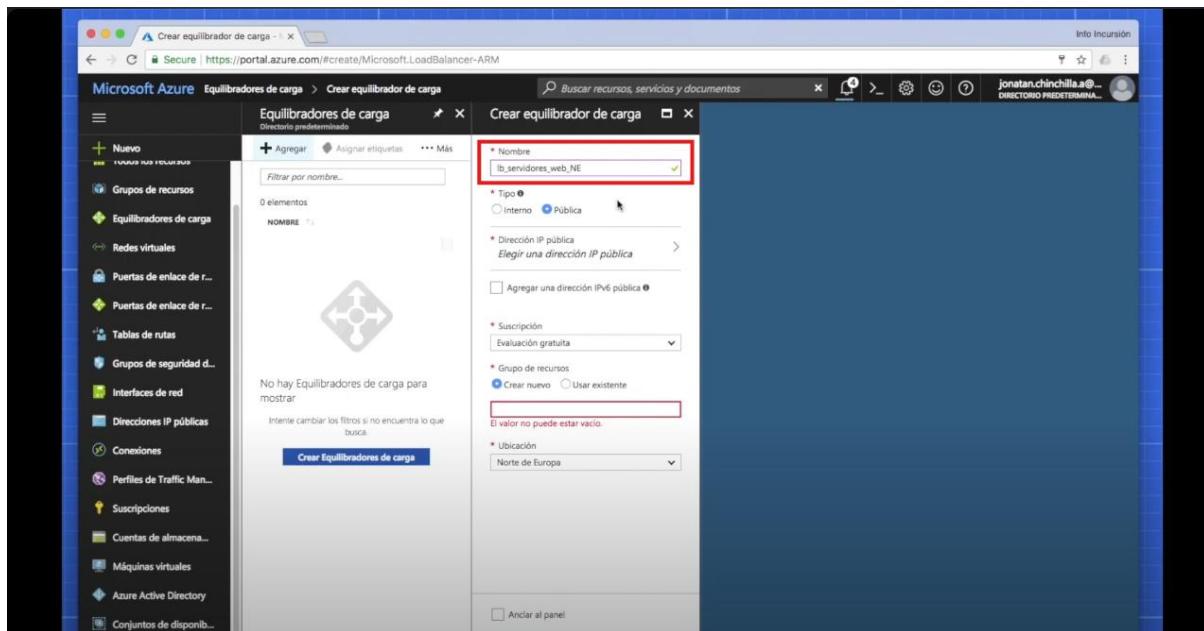
Para crear un balanceador de carga en el portal de Azure nos situaremos sobre el menú de la izquierda y seleccionaremos la opción “equilibradores de carga”.



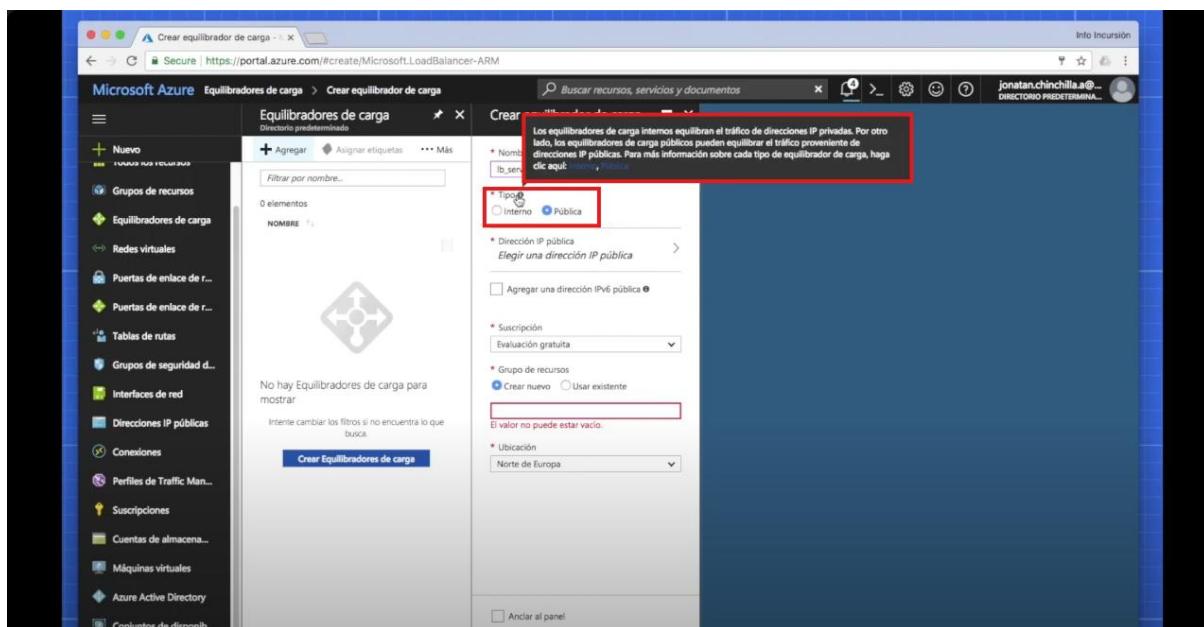
Como vemos, no tenemos ningún balanceador creado, así que le damos a la opción “agregar” situada en la parte de arriba.



Como hemos visto anteriormente (y creado la maquina Linux), va a ser un balanceador de servidores. En “nombre” ponemos la etiqueta de lo que se encargara el balanceador.

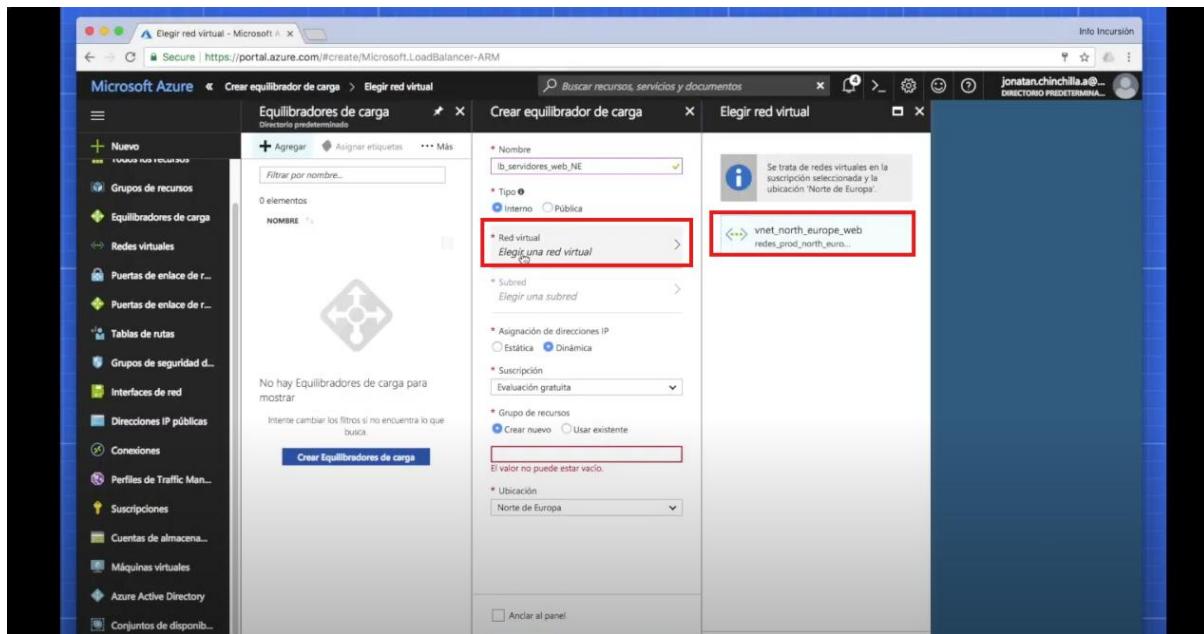


Nos muestra dos opciones en “tipo”. Una de ellas que es “interno” es básicamente que no tendrá salida a internet, seria por ejemplo si quisieramos utilizar esos servidores para una sala, un despacho, un laboratorio... la opción “publica” es que todo el mundo pueda acceder a ese servidor. Para nuestro caso práctico usaremos interno.

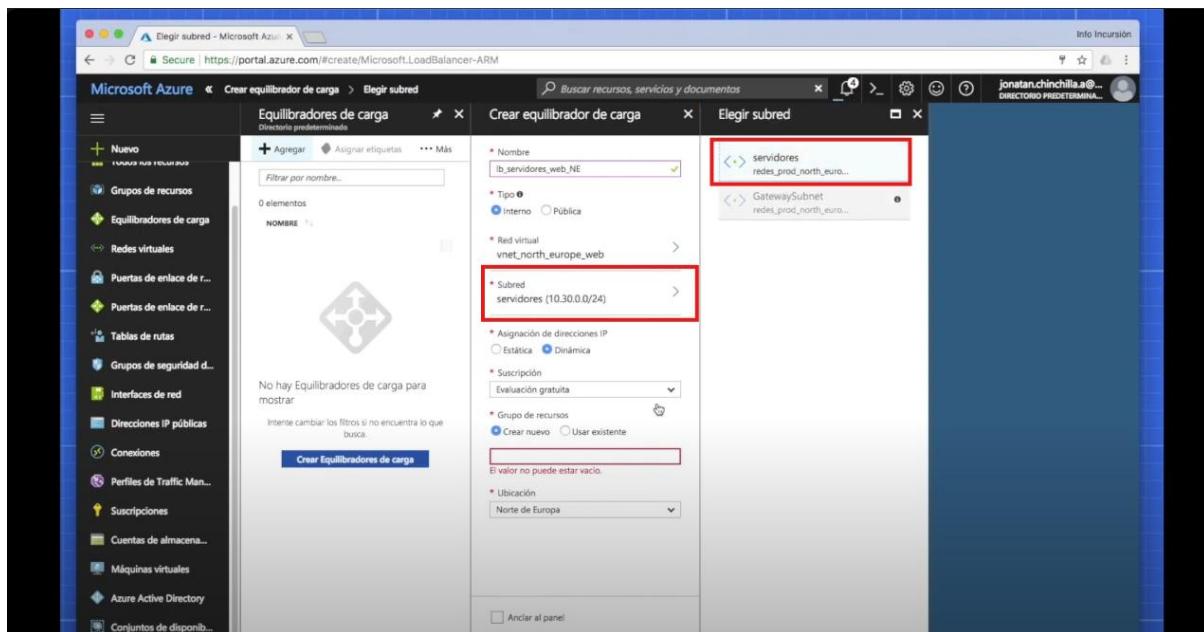


## COMPARATIVA AWS Y AZURE

“Red virtual” donde seleccionaremos la red que creamos con anterioridad de nuestra maquina Linux.

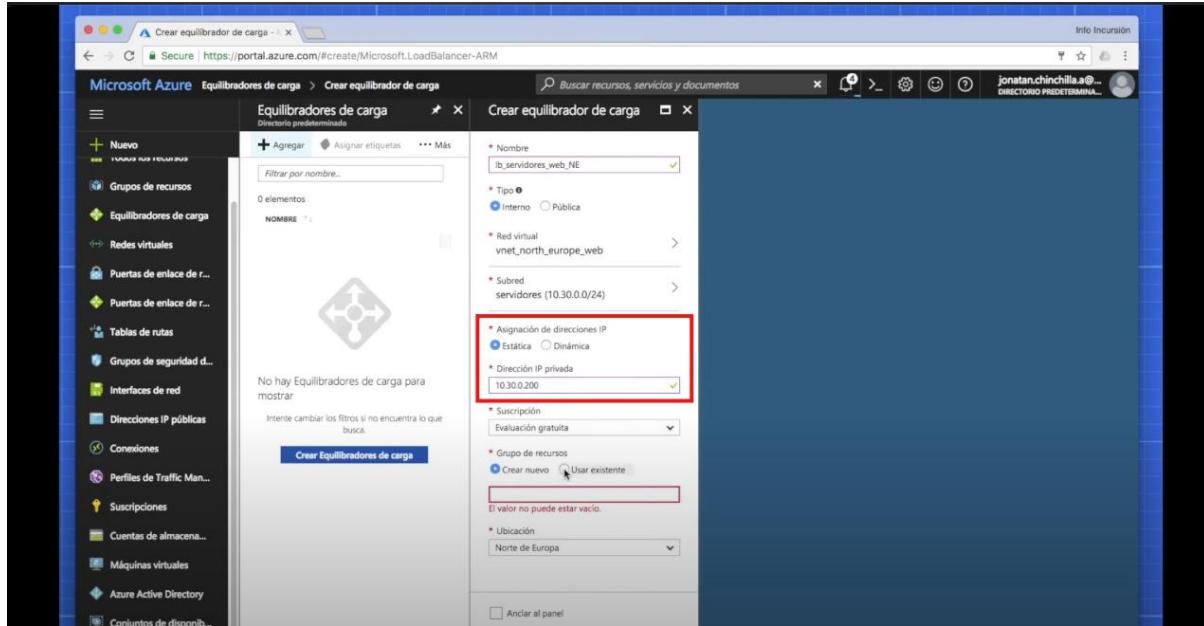


“Subred” donde está la configuración de red que configuramos previamente en nuestras maquinas Linux.

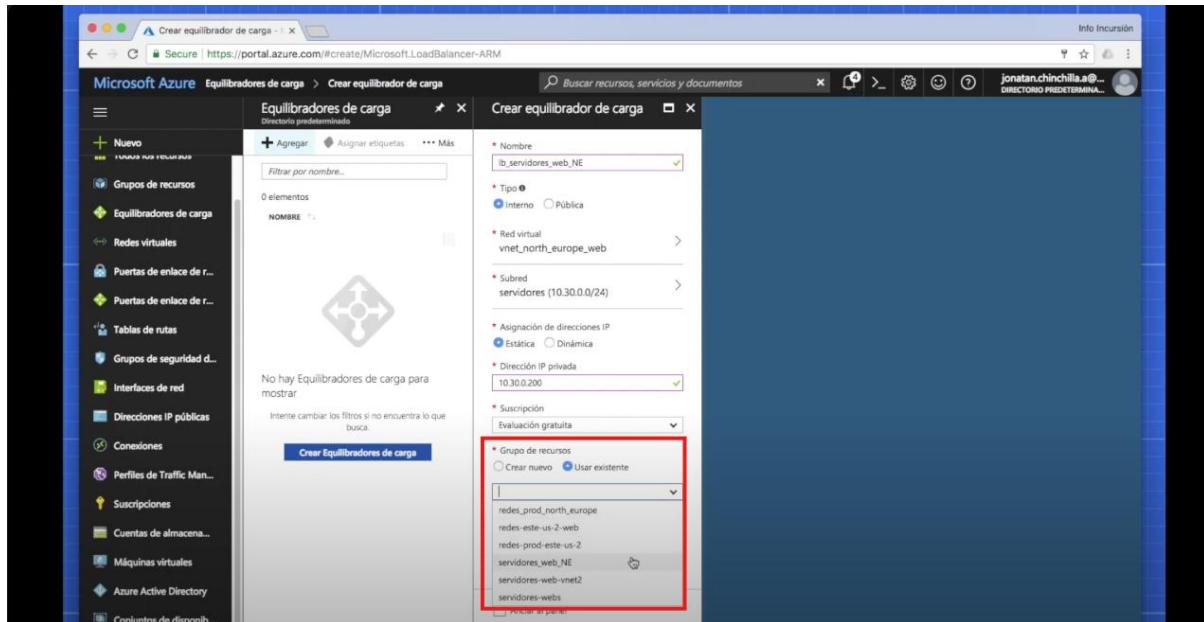


## COMPARATIVA AWS Y AZURE

“Asignación de direcciones IP” donde nos darán 2 opciones. “Estática” que pondremos nosotros a mano el direccionamiento IP que deseamos. “automático” donde nos darán una IP aleatoria y cambiara cada cierto tiempo. Usaremos “estática” ya que así te permite un control de tus servidores.



“Grupo de recursos” donde está nuestro grupo de máquinas virtuales (4 que creamos de Linux) llamada “servidores”. Le damos a “crear” la opción que esta más abajo.



- **CLI:**

Este es el código que se usa desde la CLI de Azure para crear un Balanceador de Carga. Este comando crea un balanceador de carga en la región East US, dentro del grupo de recursos MiGrupoDeRecursos, y se ubicará en una red virtual llamada MiRedVirtual y en una subred llamada MiSubred.

Este balanceador de carga podría ser utilizado para distribuir tráfico a varias instancias de máquinas virtuales u otros recursos.

```
az network lb create \
--name MiBalanceadorDeCarga \
--resource-group MiGrupoDeRecursos \
--location eastus \
--sku Standard \
--frontend-ip-name MiIPFrontend \
--backend-pool-name MiPoolBackend \
--vnet-name MiRedVirtual \
--subnet MiSubred
```

- **IaC (Terraform o ARM Templates):**

Este código de Terraform define la creación de un Balanceador de Carga en Azure utilizando el proveedor azurerm. En concreto, crea un balanceador de carga Standard en la región East US dentro de un grupo de recursos específico. Este balanceador de carga será útil para distribuir el tráfico de red hacia las máquinas virtuales.

```
resource "azurerm_lb" "mi_balanceador" {  
    name          = "MiBalanceadorDeCarga"  
    location      = "East US"  
    resource_group_name = azurerm_resource_group.mi_grupo.name  
    sku           = "Standard"  
  
    frontend_ip_configuration {  
        name          = "MiIPFrontend"  
        public_ip_address_id = azurerm_public_ip.mi_ip.id  
    }  
  
    backend_address_pool {  
        name = "MiPoolBackend"  
    }  
  
    inbound_nat_rule {  
        name          = "MiReglaNAT"  
        protocol      = "Tcp"  
        frontend_port = 22  
        backend_port   = 22  
        frontend_ip_configuration = azurerm_lb_frontend_ip_configuration.mi_ip_frontend.id  
    }  
}
```

Este código crea un Balanceador de Carga en Azure en la región East US. El balanceador tiene una configuración de IP frontal. El tráfico se distribuye entre un grupo de backend, con una regla de balanceo de carga que utiliza el puerto 80 para tráfico TCP, con un tiempo de inactividad de 4 minutos y sin IPs flotantes.

Este balanceador de carga ayudará a distribuir el tráfico de red entre las máquinas virtuales u otros recursos que estén en el grupo de backend y asegurará la disponibilidad y accesibilidad de los servicios a través de una IP pública.

```
{
  "type": "Microsoft.Network/loadBalancers",
  "apiVersion": "2021-03-01",
  "location": "eastus",
  "properties": {
    "frontendIPConfigurations": [
      {
        "name": "MiIPFrontend",
        "properties": {
          "publicIPAddress": {
            "id": "[resourceId('Microsoft.Network/publicIPAddresses', 'MiIPPublica')]"
          }
        }
      }
    ],
    "backendAddressPools": [
      {
        "name": "MiPoolBackend"
      }
    ],
    "loadBalancingRules": [
      {
        "name": "MiReglaDeBalanceo",
        "properties": {
          "frontendIPConfiguration": {
            "id": "[resourceId('Microsoft.Network/loadBalancers/frontendIPConfigurations', 'MiFrontend')]"
          },
          "backendAddressPool": {
            "id": "[resourceId('Microsoft.Network/loadBalancers/backendAddressPools', 'MiBackend')]"
          },
          "protocol": "Tcp",
          "frontendPort": 80,
          "backendPort": 80,
          "enableFloatingIP": false,
          "idleTimeoutInMinutes": 4,
          "loadDistribution": "Default"
        }
      }
    ]
  }
}
```

## 4. Comparación de características

### 4.1. AWS

#### ▪ Consola gráfica

La consola gráfica de AWS es altamente funcional y permite gestionar y configurar una gran cantidad de servicios. Su interfaz es muy completa, pero puede resultar algo abrumadora al principio debido a la gran cantidad de opciones y menús.

Un gran punto a su favor es el acceso a la configuración visual y control directo de los servicios.

Sin embargo, cuando se trata de proyectos grandes o cuando existen muchos servicios puedes llegar a ser una interfaz compleja y difícil de navegar.

#### ▪ CLI

El AWS CLI es muy completo y permite interactuar con casi todos los servicios de AWS desde la línea de comandos. Está basado en Python y es altamente automatizable. Los comandos son bastante detallados, y su estructura de uso puede ser compleja.

Algo favorable es que es flexible y bien documentada, adecuada para scripts y automatización.

Puede ser difícil para los novatos debido a la sintaxis compleja en algunos servicios.

#### ▪ Infrastructure as Code (IaC)

AWS CloudFormation permite gestionar la infraestructura como código usando plantillas **JSON** o **YAML**. Es altamente integrado con todos los servicios de AWS.

Su punto fuerte es la total integración con AWS, es muy adecuado para entornos de alta automatización y replicación.

El problema es que puede ser complejo de utilizar, sobre todo para los novatos, debido a sus sintaxis y estructuras.

Otra opción que existe ajena a CloudFormation y a AWS es Terraform.

**Terraform** es una herramienta que permite gestionar la infraestructura en AWS, Azure y otros proveedores. Su sintaxis en **HCL** es bastante legible y facilita la colaboración.

Su punto a favor es que es multinube, es decir, funciona con cualquier plataforma de servicios en la nube, es fácil de aprender y utilizar. Se adapta especialmente bien a múltiples proveedores.

El principal problema que presenta es que no está completamente integrado en todas las características de cada proveedor, y su ciclo de vida puede ser un poco más largo comparado con las soluciones nativas.

### ▪ **Facilidad de uso**

AWS tiene una curva de aprendizaje algo empinada, especialmente debido a su amplio catálogo de servicios y su complejidad al gestionar configuraciones avanzadas. Sin embargo, la comunidad de AWS es enorme y hay mucha documentación disponible.

Precisamente este es su mayor punto a favor, debido a esta gran cantidad de documentación y tutoriales. A parte de esto, es flexible para ajustarse a proyectos de diferentes tamaños y complejidades.

Como mencionamos en el apartado de la interfaz gráfica, en proyectos sobrecargados de servicios la configuración puede ser confusa y sobre todo para nuevos usuarios o para pequeñas empresas.

### ▪ **Costes aproximados**

AWS tiene un modelo de precios basado en el uso de recursos, y aunque puede ofrecer precios competitivos en muchos servicios, su modelo de precios es muy complejo, lo que hace que las estimaciones de costos sean difíciles de calcular sin herramientas específicas.

Su gran beneficio es que existe una amplia variedad de servicios que permiten adaptarse a diferentes presupuestos.

El problema es que los costes pueden crecer rápidamente si no se gestionan adecuadamente, debido a la gran cantidad de servicios y la facturación detallada por cada uno.

### ▪ **Diferencias en configuraciones y opciones disponibles**

- Amplia gama de servicios: AWS tiene una oferta de servicios más amplia que Azure, especialmente en áreas como Big Data, Machine Learning y Containers.
- Control granular: AWS ofrece un control más granular sobre configuraciones, lo que puede ser beneficioso para arquitecturas complejas, pero también puede complicar la gestión.
- Ecosistema: AWS tiene una mayor presencia en el mundo de startups, innovación y en empresas de tecnología.

**AWS** es generalmente preferido para infraestructuras puramente basadas en la nube con un enfoque en innovación y escalabilidad.

### ▪ **Implementación práctica**

En **AWS**, puedes empezar rápidamente con los **Servicios Básicos** como EC2 (máquinas virtuales), S3 (almacenamiento) y VPC (redes). AWS también tiene **Amazon Lightsail**, que es una opción más simple para implementar aplicaciones con configuraciones predefinidas. La implementación de recursos se maneja principalmente a través de la consola, CLI o IaC.

## 4.2. AZURE

### ▪ Consola gráfica

La consola gráfica de Azure, conocida como **Azure Portal**, es bastante intuitiva y también permite gestionar la infraestructura de manera visual. Ofrece dashboards personalizables, lo que facilita la visualización del estado de tus recursos.

Su punto fuerte es que ofrece una interfaz amigable, fácil de personalizar con diferentes vistas y dashboards.

Sin embargo, al igual que AWS, en configuraciones más avanzadas o en proyectos algo más voluminosos puede ser algo abrumadora.

### ▪ CLI

Azure también ofrece una CLI bastante robusta y eficiente para interactuar con sus servicios. Está escrita en Python y es similar a la AWS CLI en términos de funcionalidades.

Como punto a favor, cabe mencionar que es más fácil de aprender para usuarios nuevos, con una sintaxis clara y concisa.

El problema viene con ciertos comandos que pueden ser menos intuitivos, especialmente si ya tienes experiencia con AWS CLI.

### ▪ Infrastructure as Code (IaC)

Los **Azure Resource Manager (ARM)** Templates son equivalentes a los templates de CloudFormation, utilizando **JSON** para describir la infraestructura. Azure también soporta herramientas como **Bicep**, que simplifican la creación de plantillas.

El principal punto a favor de esta herramienta es un buen soporte y documentación además de una integración profunda con todos los servicios de Azure.

Sin embargo, como ocurre en CloudFormation, la sintaxis de JSON puede ser complicada y propensa a errores.

Otra opción que existe ajena a ARM Templates y a Azure es Terraform.

**Terraform** es una herramienta que permite gestionar la infraestructura en AWS, Azure y otros proveedores. Su sintaxis en **HCL** es bastante legible y facilita la colaboración.

Su punto a favor es que es multinube, es decir, funciona con cualquier plataforma de servicios en la nube, es fácil de aprender y utilizar. Se adapta especialmente bien a múltiples proveedores.

El principal problema que presenta es que no está completamente integrado en todas las características de cada proveedor, y su ciclo de vida puede ser un poco más largo comparado con las soluciones nativas.

#### ▪ **Facilidad de uso**

Azure tiene una curva de aprendizaje menos empinada, especialmente para empresas que ya están familiarizadas con productos de Microsoft. La interfaz del **Azure Portal** está diseñada para ser más amigable y directa. Azure también se integra muy bien con herramientas populares como **Active Directory** y **Office 365**.

Es muy fácil de usar, especialmente para usuarios ya familiarizados con productos de Microsoft. El **Azure Portal** tiene un enfoque muy accesible.

Aunque es fácil para usuarios de Microsoft, puede no ser tan flexible o escalable para necesidades más avanzadas en comparación con AWS.

#### ▪ **Costes aproximados**

El modelo de precios de Azure es algo similar al de AWS, pero generalmente, Azure tiende a ser más barato para ciertos servicios, especialmente en términos de máquinas virtuales y almacenamiento.

Tiene precios competitivos, especialmente para empresas que ya están en el ecosistema de Microsoft.

La principal contra es que el sistema de precios también es bastante complejo y es necesario realizar una buena planificación para evitar costos inesperados.

#### ▪ **Diferencias en configuraciones y opciones disponibles**

- Fuerte integración con Microsoft: Azure ofrece una integración más fluida con productos como Active Directory, Windows Server, Office 365, etc. Si tu infraestructura ya está basada en tecnologías Microsoft, Azure tiene una ventaja.
- Foco en híbrido: Azure sobresale en soluciones híbridas y es más fuerte en entornos de nube privada y on-premise (por ejemplo, mediante Azure Arc y Azure Stack).
- Facilidad para empresas: Muchas empresas prefieren Azure debido a su fácil integración con su infraestructura existente (como servidores Windows, licencias de SQL Server, etc.).

Azure es más fuerte en **integración híbrida** y en **soluciones empresariales** donde Microsoft es clave.

- **Implementación práctica**

En Azure, puedes usar **Azure Resource Groups** para organizar recursos. La implementación práctica es muy sencilla, especialmente para usuarios familiarizados con Microsoft. Azure también tiene **Azure App Services** para aplicaciones web, **Azure Functions** para serverless, y **Azure VM** para máquinas virtuales, similar a EC2 de AWS. Azure proporciona una interfaz amigable con **Azure Marketplace** para obtener soluciones prediseñadas.

## 5. Precios en Azure y AWS

- Azure

- Máquinas virtuales → 67,26 € /ud. (x4) → 269,04 € / mes
  - 4 Máquinas
  - 2 vCPUs
  - 8GB RAM
  - 730H (Bajo demanda)
- Red virtual → 3,84 € / mes
- Balanceador de carga → 22,31 € / mes
  - 5 reglas del balanceo
  - 1000 GB procesamiento de datos

**Virtual Machines**

Get €200 credit plus free monthly amounts of popular services for 12 months—including Virtual Machines. See free amounts [▼](#)

Region:	Operating system:	Type:	Tier:
East US	Linux	Ubuntu	Standard

Category:	Instance Series:	INSTANCE: <a href="#">Need help finding the right VM?</a>
All	All	D2 v3: 2 vCPUs, 8 GB RAM, 50 GB Temporary storage, €0.092/hour
<input type="button" value="4"/> Virtual machines	<input type="button" value="730"/> Hours	

**Savings Options**

Explore pricing models to help optimize your Azure costs. [Learn more](#)

<input checked="" type="radio"/> Compute (D2 v3)	<input type="radio"/> Pay as you go
--	-------------------------------------

**Virtual Network**

VNet 1 Region: VNet 2 Region:

East US	East US
---------	---------

[ⓘ Virtual Network in Azure is free of charge. Charges can be incurred by using VNET Peering between two networks or by deploying NAT Gateways.](#)

Virtual Network 1: East US VNET Peering: East US -> East US <a href="#">ⓘ</a>
100 GB

[View Data Transfer Costs](#) €1.92

Virtual Network 2: East US VNET Peering: East US -> East US <a href="#">ⓘ</a>
100 GB

[View Data Transfer Costs](#) €1.92

Upfront cost	€0.00
Monthly cost	€3.84

## COMPARATIVA AWS Y AZURE

Region: West US      Tier: Standard

**Load Balancer rules**

5 Rules      €17.51

**NAT rules**

(i) NAT rules are free.

**Data processed**

1000 GB      €0.005 Per GB      €4.80

(i) In addition to the Data Processed charges, Bandwidth charges are also applicable. [Learn more about Bandwidth pricing.](#)

Upfront cost	€0.00
Monthly cost	€22.31

Your Estimate +

**Your Estimate**

Virtual Network	(i) East US (Virtual Network 1): 100 GB Outbound Data...             Upfront: €0.00      Monthly: €3.84
Virtual Machines	(i) 4 D2 v3 (2 vCPUs, 8 GB RAM) x 730 Hours (Pay as you go)             Upfront: €0.00      Monthly: €269.02
Load Balancer	(i) Standard Tier: 5 Rules, 1,000 GB Data Processed             Upfront: €0.00      Monthly: €22.31

**Support**

SUPPORT:

Basic (Included)      €0.00

**Select your program/offer**

LICENSING PROGRAM:

Microsoft Customer Agreement (MCA)      (i) [Log in](#) to see your Azure agreement pricing.

Show Dev/Test Pricing (i)

**Estimated upfront cost**      €0.00

**Estimated monthly cost**      €295.17

- **TOTAL MENSUAL → 295,17€**
- **TOTAL ANUAL → 3542,04 €**

## COMPARATIVA AWS Y AZURE

- AWS

- Máquinas virtuales → 46,84 € /ud. (x4) → 187,36 € / mes
  - 4 Máquinas
  - 2 vCPUs
  - 8GB RAM
  - 730H (Bajo demanda)
- Red virtual → 3,48 € / mes
- Balanceador de carga → 23,33 € / mes
  - 5 reglas del balanceo
  - 1000 GB procesamiento de datos

The screenshot shows the AWS EC2 instance creation process across two main sections: 'Especificaciones de EC2' (EC2 specifications) and 'Instancias EC2 (714)' (List of EC2 instances).

**Especificaciones de EC2:**

- Tenencia:** Selected 'Este de EE. UU. (Ohio)'.
- Sistema operativo:** Selected 'Linux'.
- Cargas de trabajo:** Selected 'Uso constante'.
- Cantidad de instancias:** Specified as '4'.

**Instancias EC2 (714):**

- Summary:** Based on your inputs, this is the lowest-cost EC2 instance: **t4g.large**.
- Instance Details:** Instance selected: **t4g.large** | Family: t4g | 2vCPU | 8 GiB Memoria.
- Filters:** Set to 'Any Instance family' (2 vCPUs), '8 GiB' Memory, and 'Any Network Performance'.
- Results:** A table lists various EC2 instance types, with **t4g.large** highlighted in blue.

Nombre de instancia	Familia de instancias	Categoría de instancia	vCPU	Núcleos físicos	Memoria	Rendimiento de la red	Almacenamiento
<b>t4g.large</b>	t4g.large	General purpose	2	2	8 GiB	Up to 5 Gigabit	EBS only
t3a.large	t3a.large	General purpose	2	2	8 GiB	Up to 5 Gigabit	EBS only
m6g.large	m6g.large	General purpose	2	2	8 GiB	Up to 10 Gigabit	EBS only
m7g.large	m7g.large	General purpose	2	2	8 GiB	Up to 12500 Megabit	EBS only
t3.large	t3.large	General purpose	2	2	8 GiB	Up to 5 Gigabit	EBS only
m5a.large	m5a.large	General purpose	2	2	8 GiB	Up to 10 Gigabit	EBS only
m6a.large	m6a.large	General purpose	2	2	8 GiB	Up to 12500 Megabit	EBS only
m8g.large	m8g.large	General purpose	2	2	8 GiB	Up to 12.5 Gigabit	EBS only
m6gd.large	m6gd.large	General purpose	2	2	8 GiB	Up to 10 Gigabit	1 x 118 NVMe SSD
t2.large	t2.large	General purpose	2	2	8 GiB	Low to Moderate	EBS only

## COMPARATIVA AWS Y AZURE

**Opciones de pago**

Precio de compromiso estimado basado en las siguientes opciones:  
Tipo de instancia: **t4g.large** Sistema operativo: **Linux**

Seleccione el contenido y las opciones para encontrar su mejor precio

Compute Savings Plans  
Un plan que se aplica automáticamente a todo el uso de EC2, Fargate y Lambda. Hasta un 66 % de descuento. [Obtenga más información](#)

Plazo de reserva  
 1 year  
 3 year

Opciones de pago  
 Sin pagos iniciales  
 Pago inicial parcial  
 Pago inicial total

**Inicial: 0.00**  
**Mensual: 24.38/Mes**

EC2 Instance Savings Plans  
Obtenga más descuentos cuando solo necesita una región y familia de instancias. Hasta un 72 % de descuento. [Obtenga más información](#)

Plazo de reserva  
 1 year  
 3 year

Opciones de pago  
 Sin pagos iniciales  
 Pago inicial parcial  
 Pago inicial total

**Inicial: 0.00**  
**Mensual: 21.17/Mes**

Bajo demanda [Obtenga más información](#)  
Maximice la flexibilidad.

**Expected utilization**  
Introduzca el uso previsto de las instancias de Amazon EC2

Uso:   
Tipo de uso:

**Instancia: 0.0672/Hora**  
**Mensual: 49.06/Mes**

Instancias de spot  
Minimize los costos aprovechando la capacidad sobrante de EC2. Recomendado para aplicaciones tolerantes a errores y a interrupciones. [Obtenga más información](#)

El descuento promedio histórico del t4g.large es 71 %

Assume percentage discount for my estimate

**Instancia: 0.0672/Hora**  
**Mensual: 198.19/Mes**

▶ Otras opciones de compra

Elija un tipo de ubicación [Información](#)

Región

Elija una región

Selección los servicios de VPC que desea calcular

VPN Connection

Network Address Translation (NAT) Gateway

Public IPv4 Address

Transit Gateway

AWS PrivateLink

Data Transfer

Gateway Load Balancer

IPAM

Network Access Analyzer

Traffic Mirroring

Reachability Analyzer

Cloud WAN

**Public IPv4 Address característica**

Dirección IPv4 pública

Los cálculos que se muestran a continuación no incluyen los descuentos de la capa gratuita.

Cantidad de direcciones IPv4 públicas en uso

Cantidad de direcciones IPv4 públicas inactivas

▼ Mostrar cálculos  
1 dirección IPv4 pública en uso x 730 horas en un mes x 0,005 USD = 3,65 USD (Cargo total por dirección IPv4 pública en uso)  
Cargo total por dirección IPv4 pública (mensual): **3.65 USD**

## COMPARATIVA AWS Y AZURE

**Configuración de servicio información**

Número de balanceadores de carga de aplicaciones  
1

▼ Mostrar cálculos  
1 balanceadores de carga x 0,0225 USD por hora x 730 horas en un mes = 16,43 USD  
Cargos fijos por hora del Balanceador de carga de aplicaciones (mensual): 16.43 USD

**Unidades de capacidad de balanceador de carga (LCU) información**

Especifique al menos una de las siguientes dimensiones para determinar los precios de LCU.

Bytes procesados (funciones Lambda como objetivos)  
Escriba el total de datos procesados por ALB para las funciones Lambda como objetivos.

Value	0	Unidad	GB por hora
-------	---	--------	-------------

Bytes procesados (instancias EC2 y direcciones IP como objetivos)  
Escriba el número total de datos procesados por ALB para instancias EC2 y direcciones IP como objetivos.

Value	1000	Unidad	GB por mes
-------	------	--------	------------

Número promedio de conexiones nuevas por ALB  
Escriba el número promedio de conexiones nuevas por conexión.

Value	100	Unidad	por minuto
-------	-----	--------	------------

Duración promedio de conexión  
Escriba la duración promedio para cada conexión nueva (si la duración es inferior a 1 segundo, escriba 1 segundo).

Value	5	Unidad	segundos
-------	---	--------	----------

Número promedio de solicitudes por segundo por ALB  
Escriba el número promedio de solicitudes por conexión.

Value	100	Unidad	segundos
-------	-----	--------	----------

Número promedio de evaluaciones de las reglas por solicitud  
Escriba el número promedio de evaluaciones de las reglas por solicitud.

Value	5	Unidad	segundos
-------	---	--------	----------

▼ Mostrar cálculos

Conversión de unidades

Bytes procesados (instancias EC2 y direcciones IP como objetivos): 1000 GB por mes x 0.00136986 meses en una hora = 1.36986 GB por hora  
Número promedio de conexiones nuevas por ALB: 100 por minuto / (60 segundos en un minuto) = 1.67 por segundo

Cálculos de precios  
1.36986 GB por hora / 1 GB de bytes procesados por hora por LCU para instancias EC2 y direcciones IP como destinos = 1,36986 LCU de bytes procesados para instancias EC2 y direcciones IP como objetivos  
1,67 conexiones nuevas por segundo / 25 conexiones nuevas por segundo por LCU = 0,0668 LCU de las conexiones nuevas  
1,67 conexiones nuevas por segundo x 5 segundos = 8,35 conexiones activas  
8,35 conexiones activas / 3000 conexiones por LCU = 0,00278333333333334 LCU de las conexiones activas  
5 reglas por solicitud - 10 reglas gratuitas = -5 reglas de pago por solicitud después de las 10 reglas gratuitas  
Max (-5 USD, 0 USD) = 0,00 reglas de pago por solicitud  
Max (1.36986 LCU de bytes procesados , 0,00278333333333334 LCU de las conexiones activas , 0 LCU de las evaluaciones de las reglas ) = 1,36986 LCU máximo  
1 balanceadores de carga x 1,36986 LCU x 0,008 Precio por hora de LCU x 730 horas al mes = 8,00 USD  
Cargos por uso de la LCU del Balanceador de carga de aplicaciones (mensual): 8.00 USD

AWS Pricing Calculator > Mi estimación

**My Estimate** [Editar](#) [Exportar](#) [Compartir](#)

**Resumen de la estimación información**

Costo inicial 0,00 USD	Costo mensual 224,30 USD	Costo total de 12 months <b>2691,60 USD</b> Incluye el costo inicial
---------------------------	-----------------------------	--

**Comenzar con AWS**

[Comience de forma gratuita](#) [Comuníquese con el departamento de ventas](#)

**My Estimate**

Nombre del servicio	Estado	Costo inicial	Costo mensual	Descripción	Región	Resumen de la configuración
Amazon EC2	-	0,00 USD	196,22 USD	-	Este de EE. UU. (Ohio)	Tenencia (Instancias compart...
Amazon Virtual Private Clou...	-	0,00 USD	3,65 USD	-	Este de EE. UU. (Ohio)	Cantidad de direcciones IPv4 ...
Elastic Load Balancing	-	0,00 USD	24,43 USD	-	Este de EE. UU. (Ohio)	Número de balanceadores de...

- **TOTAL MENSUAL → 214,16 €**
- **TOTAL ANUAL → 2569,87 €**

## 6. Conclusiones y recomendaciones

**AWS** es ideal para organizaciones que buscan una **gran variedad de servicios y flexibilidad** en configuraciones avanzadas, especialmente en proyectos de **gran escala** y con necesidades complejas. Su **curva de aprendizaje** puede ser pronunciada, pero ofrece un control granular y opciones avanzadas que lo hacen perfecto para **entornos multi-nube y arquitecturas complejas**. Sin embargo, su **estructura de precios** puede resultar complicada.

**Azure**, por otro lado, destaca por su **facilidad de uso**, especialmente si ya estás integrado en el ecosistema de **Microsoft**. Es una excelente opción para **entornos híbridos** y empresas que ya utilizan **Active Directory, Office 365** y otras soluciones Microsoft. Aunque no tiene tanta variedad como AWS, sus precios pueden ser más competitivos en áreas clave y ofrece una **interfaz más amigable**.

Como recomendación, si se requiere una integración estrecha con servicios de Microsoft, Azure es la mejor opción.

Para mayor personalización y una oferta de servicios más amplia, AWS es preferible.  
Utilizar Terraform para IaC si se busca una solución multi-nube.  
Considerar el uso de CLI para automatizar tareas repetitivas.

Como resumen referente a las recomendaciones, la mejor opción será **AWS** si necesitas una **infraestructura más compleja y flexible**, y **Azure** si buscas **facilidad de uso**, integración con Microsoft y soluciones híbridas.