



Peer Connection Transit Gateway

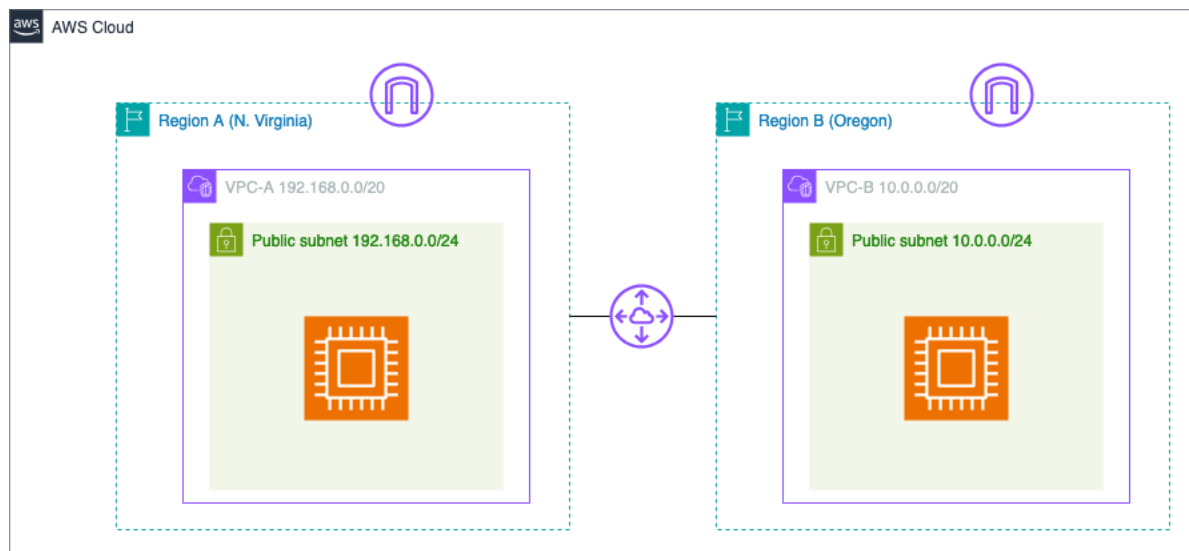
Peer Connection	3
Escenario.....	3
Peer connection	4
Transit Gateway	11
Escenario.....	11
Transit Gateway	12
Tareas	17

Peer Connection

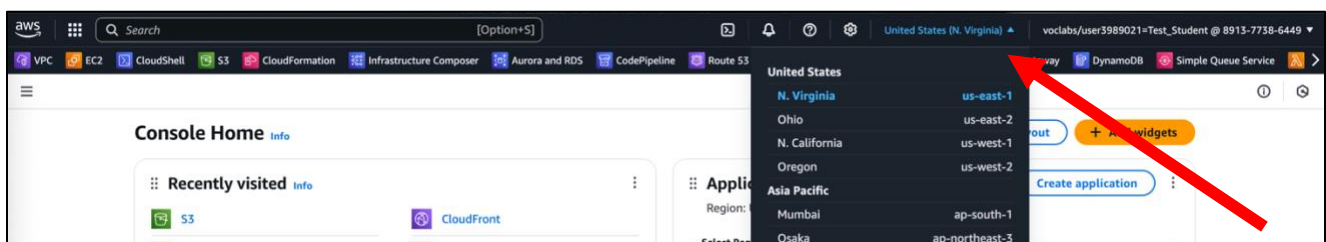
Es un servicio que establece una conexión entre dos VPCs dentro de la misma región o para lo que más se usa entre diferentes regiones. El objetivo de la práctica es lanzar dos instancias EC2, cada una de ellas en una región diferente y lograr comunicarse mediante el servicio de peer connection y no lograr comunicarse desde fuera de ellas.

Escenario

A definir en la instancia los siguientes elementos:



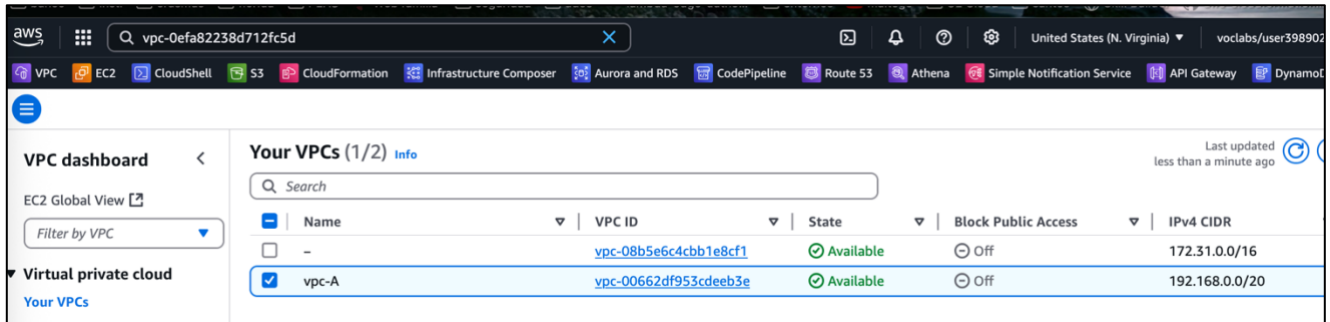
AWS divide el mundo en regiones principalmente por temas latencia y seguridad, con este dato, ahora podemos indicar que siempre hemos estado trabajando en la región del Norte de Virginia (us-east-1), esto lo podemos saber mirando en la barra de menú y desplegando la zona en la parte derecha.



Cabe destacar que en la capa gratuita del lab sólo tenemos acceso a dos regiones para hacer pruebas: N. Virginia (us-east-1) y Oregon (us-west-2).

Peer connection

Lo primero que haremos es en las dos regiones N. Virginia y Oregón crear una VPC con una subred pública. Además, crearemos también un grupo de seguridad en cada región para por ejemplo permitir el acceso por ssh, las VPC se mostrarían como se muestran a continuación:



United States (N. Virginia) | voclabs/user398902

Your VPCs (1/2) Info

Name	VPC ID	State	Block Public Access	IPv4 CIDR
-	vpc-08b5e6c4cbb1e8cf1	Available	Off	172.31.0.0/16
vpc-A	vpc-00662df953cdeeb3e	Available	Off	192.168.0.0/20

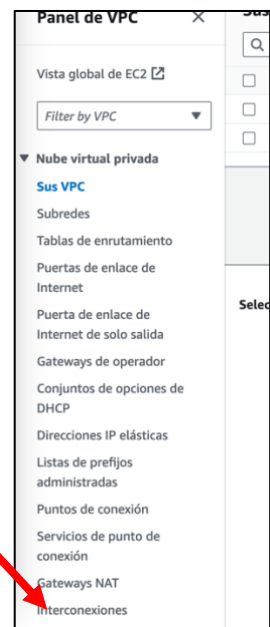


United States (Oregon) | voclabs/user3989021

Your VPCs (1/2) Info

Name	VPC ID	State	Block Public...	IPv4 CIDR
-	vpc-01fcccfbf5b0ff806	Available	Off	172.31.0.0/16
vpc-B	vpc-0efa82238d712fc5d	Available	Off	10.0.0.0/20

Una vez creada la infraestructura (VPC y subred), el EC2 aun no nos corre prisa, vamos a interconectar las dos VPCs con el elemento "Interconexión" o "peer connection" y lo haremos desde la región us-east-1.



Ahora creamos el elemento desde la región A, para ello hay que copiar el id del VPC destino porque lo usaremos en la configuración.

VPC > Peering connections > Create peering connection

Create peering connection

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them privately. [Info](#)

Peering connection settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

intervpcA-B

Select a local VPC to peer with

VPC ID (Requester)
vpc-00662df953cdeeb3e (vpc-A)

VPC CIDRs for vpc-00662df953cdeeb3e (vpc-A)

CIDR	Status	Status reason
192.168.0.0/20	✓ Associated	-

Select another VPC to peer with

Account
☒ My account
☐ Another account

Region
☐ This Region (us-east-1)
☒ Another Region

United States (Oregon) (us-west-2)

VPC ID (Acceptor)
vpc-0efa82238d712fc5d

Ahora en la región us-east-1, podemos comprobar que se ha iniciado la petición de conexión:

[Option+S] United States (N. Virginia) voclabs

S3 CloudFormation Infrastructure Composer Aurora and RDS CodePipeline Route 53 Athena Simple Notification Service API Gateway

✓ A VPC peering connection **pcx-0251d3b5b1831df41 / intervpcA-B** has been requested. Remember to change your region to **us-west-2** to accept the peering connection.

Peering connections (1/1) Info

Find resources by attribute or tag

Name	Peering connection ID	Status	Requester VPC
intervpcA-B	pcx-0251d3b5b1831df41	Initiating request	vpc-00662df953cdeeb3e / vpc-A

Ahora en la otra región tendremos que aceptar o rechazar la solicitud de conexión:

aws [Option+S] United States (Oregon) voclabs/user3989

VPC EC2 CloudShell S3 CloudFormation Infrastructure Composer Aurora and RDS CodePipeline Route 53 Athena Simple Notification Service

VPC dashboard < EC2 Global View Filter by VPC

Peering connections (1) Info

Find resources by attribute or tag

Name	Peering connection ID	Status	Requester VPC
-	pcx-0251d3b5b1831df41	Pending acceptance	vpc-00662df953cdeeb3e

Virtual private cloud

Y para aceptar se selecciona la Acción de aceptar.

Peering connections (1/1) [Info](#)

Find resources by attribute or tag

Name	Peering connection ID	Status	Requester VPC	Accepter VPC
-	pcx-0251d3b5b1831df41	Pending acceptance	vpc-00662df953cdeeb3e	vpc-0efa82238d712fc5d

Actions

- View details
- Accept request
- Reject request
- Edit DNS settings
- Manage tags
- Delete peering connection

Y confirmamos la acción:

Accept VPC peering connection request [Info](#)

Are you sure you want to accept this VPC peering connection request? (pcx-0251d3b5b1831df41)

Requester VPC vpc-00662df953cdeeb3e	Accepter VPC vpc-0efa82238d712fc5d / vpc-B	Requester CIDRs 192.168.0.0/20
Accepter CIDRs -	Requester Region N. Virginia (us-east-1)	Accepter Region Oregon (us-west-2)
Requester owner ID 891377386449 (This account)	Accepter owner ID 891377386449 (This account)	

[Cancel](#) [Accept request](#)

Para que no haya problemas con la resolución de nombres en el servicio de interconexión en la pestaña de dns habilitaremos la opción de resolver el dns.

Peering connections (1/1) [Info](#)

Find resources by attribute or tag

Name	Peering connection ID	Status	Requester VPC	Accepter VPC
-	pcx-0251d3b5b1831df41	Provisioning	vpc-00662df953cdeeb3e	vpc-0efa82238d712fc5d

pcx-0251d3b5b1831df41

[Details](#) [DNS](#) [Route tables](#) [Tags](#)

Details

pcx-0251d3b5b1831df41

Details **DNS** Route tables Tags

DNS settings

Requester VPC (vpc-00662df953cdeeb3e) [Info](#)

Allow accepter VPC to resolve DNS of hosts in requester VPC to private IP addresses
⊖ Disabled

Accepter VPC (vpc-0efa82238d712fc5d / vpc-B) [Info](#)

Allow requester VPC to resolve DNS of hosts in accepter VPC to private IP addresses
⊖ Disabled

[Edit DNS settings](#)

Edit DNS settings [Info](#)

Summary

Peering connection ID pcx-0251d3b5b1831df41	Name -	Requester VPC vpc-00662df953cdeeb3e
---	------------------	---

Edit DNS settings

The settings below control how your peered VPCs will work with DNS resolution.

Accepter DNS resolution

If enabled, the DNS hostname of an instance in the accepter VPC resolves to its private IP address when queried from instances in the requester VPC.

☒ Allow requester VPC (vpc-00662df953cdeeb3e) to resolve DNS of accepter VPC (vpc-0efa82238d712fc5d / vpc-B) hosts to private IP.

To use DNS resolution over peering you must enable 'DNS Hostname' on the VPCs involved in peering: vpc-0efa82238d712fc5d / vpc-B, vpc-00662df953cdeeb3e (us-east-1). [Learn more](#)

► **AWS Command Line Interface command**

Ahora ya tenemos las VPCs en conexión, nos queda indicar en las tablas de enrutamiento, cuáles son las redes y a donde pueden ir, por ello en la región A (us-east-1):

VPC dashboard <

EC2 Global View [↗](#)

Filter by VPC [▼](#)

▼ **Virtual private cloud**

Your VPCs

Subnets

[Route tables](#)

Route tables (1/3) [Info](#)

Find resources by attribute or tag

	Name	Route table ID	Explicit subnet associations	Edge associat
<input type="checkbox"/>	-	rtb-02a8f3dde05afcc26	-	-
<input checked="" type="checkbox"/>	vpc-A-rtb-public	rtb-0a2930a718447818a	subnet-09273e31a8d90f99c / vpc-A-subnet-...	-
<input type="checkbox"/>	-	rtb-0f56eac5546f2591e	-	-

Last updated less than a minute ago

Indicaremos que para ir a la red de la región B podemos ir utilizando el servicio de peer connection, con lo cual va a utilizar la red troncal de Amazon (para empezar los datos van cifrados), y en la otra región (B):

Edit routes

Destination	Target	Status
192.168.0.0/20	local	Active
0.0.0.0/0	Internet Gateway	Active
10.0.0.0/20	igw-0063fd9fde732c375	
	Peering Connection	
	pcx-0251d3b5b1831df41	

[Add route](#)

VPC
vpc-00662df953cdeeb3e | vpc-A

Owner ID
891377386449

public1-us-east-1a

[Routes](#) | [Subnet associations](#) | [Edge associations](#) | [Route propagation](#) | [Tags](#)

Routes (3)

Destination	Target	Status
0.0.0.0/0	igw-0063fd9fde732c375	Active
10.0.0.0/20	pcx-0251d3b5b1831df41	Active
192.168.0.0/20	local	Active

Una vez tenemos el enrutamiento configurado en las dos regiones, vamos a preparar el grupo de seguridad para que solo acepte mensajes ICMP de las redes privadas internas, y lo hacemos en las dos regiones:

aws

Search [Option+S]

United States (Oregon) | voclabs/user3989021=Test_Student

VPC > Security Groups > Create security group

Basic details

Security group name [Info](#)
gs-icmp-vpc-b
Name cannot be edited after creation.

Description [Info](#)
permitir mensajes icmp con el peer connection

VPC [Info](#)
vpc-0efa82238d712fc5d (vpc-B)

Inbound rules [Info](#)

Type	Protocol	Port range	Source	Description - optional
All ICMP - IPv4	ICMP	All	Cust... 192.168.0.0/20 192.168.0.0/20	

[Add rule](#)

Y en la otra región:

aws

Search

[Option+S]

United States (N. Virginia)

VPC

EC2

CloudShell

S3

CloudFormation

Infrastructure Composer

Aurora and RDS

CodePipeline

Route 53

Athena

Simple Notification Service

API Gateway

[VPC](#) > [Security Groups](#) > Create security group

Create security group [Info](#)

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name [Info](#)

gs-icmp-vpc-a

Name cannot be edited after creation.

Description [Info](#)

permitir mensajes icmp con el peer connection

VPC [Info](#)

vpc-00662df953cdeeb3e (vpc-A)

Inbound rules [Info](#)

Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
All ICMP - IPv4	ICMP	All	Custom	10.0.0.0/20
				10.0.0.0/20

Ahora que parece que está todo conectado vamos a intentar hacer un ping desde la VPC A a la B, para ello crearemos un EC2 en cada región con la opción de auto asignar ip públicamente. Para poder conectar al EC2 deberemos tener también habilitado SSH en el grupo de seguridad.

```
App Store
```

```
~\#### Amazon Linux 2023
~~~\#####
~~~\###|
~~~\#/ https://aws.amazon.com/linux/amazon-linux-2023
~~~v-'-'>
~~~~
~~~.~.~
~~~/~\
~~~/m/'-
```

```
[ec2-user@ip-192-168-0-70 ~]$ ping 10.0.0.245
PING 10.0.0.245 (10.0.0.245) 56(84) bytes of data.
64 bytes from 10.0.0.245: icmp_seq=1 ttl=127 time=55.1 ms
64 bytes from 10.0.0.245: icmp_seq=2 ttl=127 time=56.0 ms
^C
--- 10.0.0.245 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 55.058/55.512/55.966/0.454 ms
[ec2-user@ip-192-168-0-70 ~]$ █
```

Y luego lo mismo desde la vpc-B

```

#
./##### Amazon Linux 2023
~~~\#####\
~~~\###|
~~~\#/ https://aws.amazon.com/linux/amazon-linux-2023
~~~v~'-'->
~~~~
~~~.-.
~~~/_/
~~~/_m/'-'->
ec2-user@ip-10-0-0-245 ~]$ ping 192.168.0.70
PING 192.168.0.70 (192.168.0.70) 56(84) bytes of data.
64 bytes from 192.168.0.70: icmp_seq=1 ttl=127 time=55.2 ms
64 bytes from 192.168.0.70: icmp_seq=2 ttl=127 time=55.1 ms
^C
--- 192.168.0.70 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 55.077/55.117/55.158/0.040 ms
ec2-user@ip-10-0-0-245 ~]$

```

E incluso podemos resolver por el nombre dns interno:

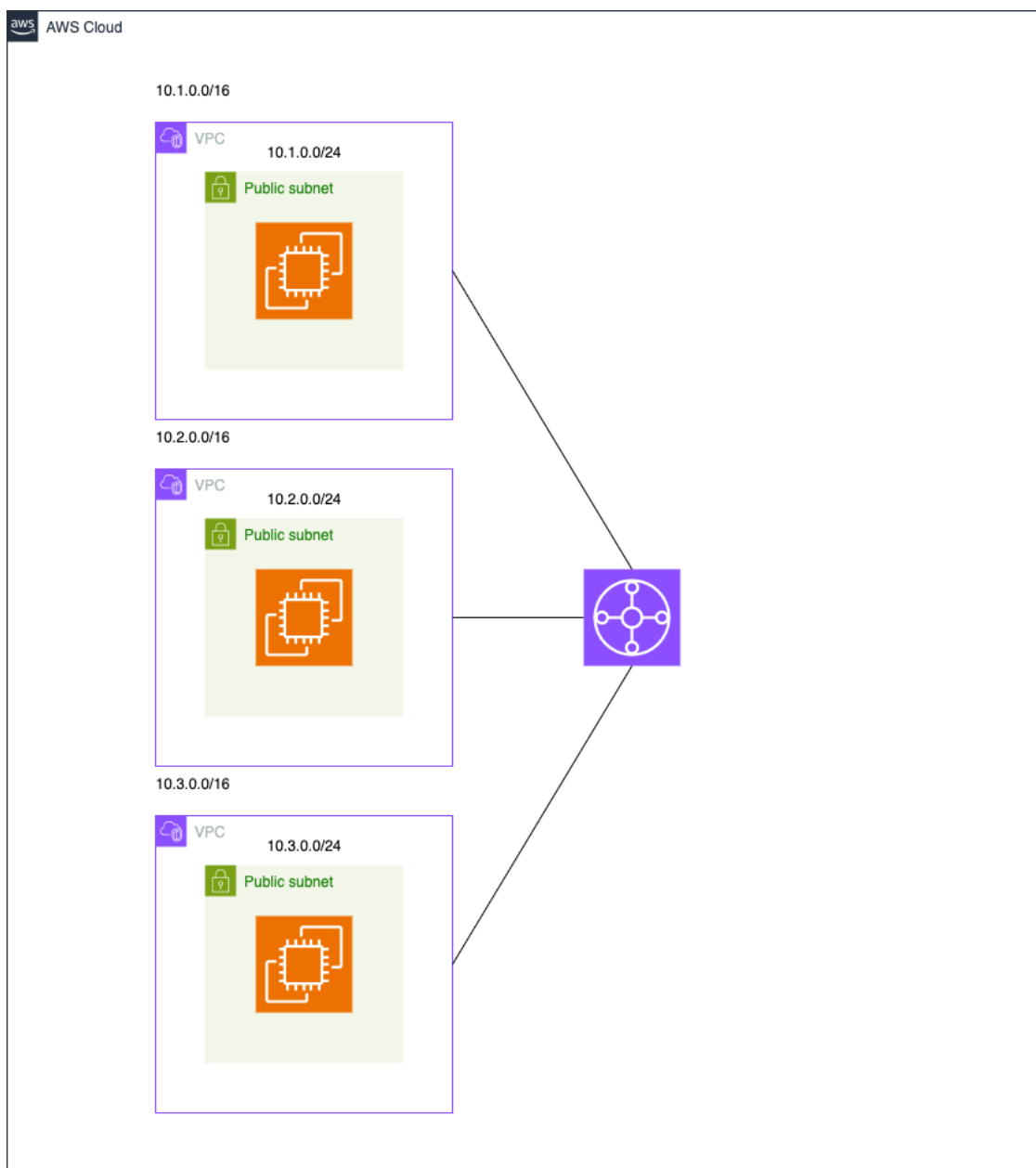
```
[ec2-user@ip-192-168-0-70 ~]$ ping ec2-54-202-138-39.us-west-2.compute.amazonaws.com
PING ec2-54-202-138-39.us-west-2.compute.amazonaws.com (10.0.0.245) 56(84) bytes of data:
64 bytes from ip-10-0-0-245.us-west-2.compute.internal (10.0.0.245): icmp_seq=1 ttl=127 time=55.0 ms
64 bytes from ip-10-0-0-245.us-west-2.compute.internal (10.0.0.245): icmp_seq=2 ttl=127 time=55.0 ms
64 bytes from ip-10-0-0-245.us-west-2.compute.internal (10.0.0.245): icmp_seq=3 ttl=127 time=55.0 ms
```

Transit Gateway

Es un servicio de enrutamiento centralizado que permite interconectar múltiples VPCs, redes on premises y VPNs en AWS. El objetivo de la práctica es lanzar tres instancias cada una dentro de su subred y en sus respectivas VPCs e interconectarlas utilizando un Transit Gateway. Cuando usamos el servicio peer connection no es una relación transitiva, por cada región que quiero interconectar debo de tener un peer connection lo cual en escenarios más complejos de redes dispara el número de elementos a gestionar.

Escenario

A definir en la instancia los siguientes elementos:



Datos de usuario de la instancia:

En el grupo de seguridad habilitaremos la conexión ssh y el ping para comprobar si podemos acceder desde cada VPC al resto.

Transit Gateway

La creación de la arquitectura la dejaremos ya por asumida, aunque las VPCs juegan en el siguiente rango:

VPC1: 10.1.0.0/16
Subred pública 1: 10.1.0.0/24

VPC2: 10.2.0.0/16
Subred pública 1: 10.2.0.0/24

VPC3: 10.3.0.0/16
Subred pública 1: 10.3.0.0/24

Una vez preparado en el entorno pasaremos a configurar el Transit gateway para poder relacionar las diferentes VPCs entre sí. En el servicio VPC encontraremos un apartado de menú donde podremos encontrar los gateways de tránsito.

Seleccionaremos el apartado de crear y empezaremos a crear uno, éste se crea vacío y habrá que configurarlo, lo único que pondremos es Name tag (nombrar etiqueta).



VPC > Gateways de tránsito > Crear gateway de tránsito

Crear gateway de tránsito Información

Una gateway de tránsito (TGW) es un centro de tránsito de red que interconecta conexiones (VPC y VPN) dentro de una cuenta o entre cuentas de AWS.

Detalles - opcional

Nombrar etiqueta
Crea una etiqueta con la clave establecida en Nombre y el valor establecido en la cadena especificada.

tgw_misvpcs

Descripción Información
Establezca la descripción de la gateway de tránsito para que pueda identificarla en el futuro.

transit gateway para interconectar los 3 vpcs

Cuando lo creemos se quedará un instante en el estado **pending**, pero enseguida se estabilizará.

Gateways de tránsito (1) información			
<input type="text" value="Buscar puerta de enlace de tránsito por atributo o etiqueta"/>			
<input type="checkbox"/>	Name ↗	ID de la puerta de enlace de tránsito	Estado
<input type="checkbox"/>	tgw_misvpcs	tgw-0ec2227a17d6d8fd8	Pending

▼ Gateways de tránsito

Gateways de tránsito

Conexiones de gateway de tránsito

Tablas de políticas de gateway de tránsito

Tablas de enrutamiento de gateway de tránsito

Multidifusión de gateway de tránsito

Una vez creado podremos comprobar que en el apartado de sus tablas de enrutamiento aparece una tabla de enrutamiento vacía, **desde aquí NO configuraremos las rutas**, este apartado se hace desde cada VPC y sus propias tablas de enrutamiento.

Antes de configurar las rutas de cada VPC, indicaremos las conexiones que puede tener el GT.

Conexiones de puerta de enlace de tránsito información			
<input type="text" value="Buscar conexión de puerta de enlace de tránsito por atributo o etiqueta"/>			
<input type="checkbox"/>	Name ↗	ID de conexión de gateway de ...	ID de la puerta de enlace ...
<p>No hay ninguna conexión de puerta de enlace de tránsito</p> <p>No tiene ninguna conexión de puerta de enlace de tránsito en esta región</p> <p>Crear conexión de puerta de enlace de tránsito</p>			
<p>Seleccionar una conexión de gateway de tránsito</p>			

En nuestro caso crearemos tres conexiones de GT (gateway de tránsito)

Detalles

Nombrar etiqueta - *opcional*
Crea una etiqueta con la clave establecida en Nombre y el valor establecido en la cadena especificada.

ID de la puerta de enlace de tránsito [Información](#)

Tipo de conexión [Información](#)

Y si hubiera más subredes dentro de la VPC, podríamos indicar el alcance a éstas y a las que no.

Vinculación de VPC
Seleccione y configure la conexión de VPC.

☒ Compatibilidad con DNS [Información](#)

☐ Compatibilidad con IPv6 [Información](#)

☐ Compatibilidad con el modo de dispositivo [Información](#)

ID de VPC
Seleccione la VPC que desea adjuntar a la gateway de tránsito.

vpc-0dc5a07bea62c0bcc

Los ID de subred [Información](#)
Seleccione las subredes en las que desea crear la conexión de VPC de gateway de tránsito.

☒ us-east-1a subnet-0ecc82f329f63921b

☐ us-east-1b No hay ninguna subred disponible

☐ us-east-1c No hay ninguna subred disponible

☐ us-east-1d No hay ninguna subred disponible

Cuando acabemos deberíamos tener algo parecido a la siguiente imagen:

Conexiones de puerta de enlace de tránsito (3) [información](#)

Buscar conexión de puerta de enlace de tránsito por atributo o etiqueta

<input type="checkbox"/>	Name	ID de conexión de gateway de ...	ID de la puerta de enlace ...	Estado
<input type="checkbox"/>	vpc3-attach	tgw-attach-058df6179957a7e08	tgw-0ec2227a17d6d8fd8	Pending
<input type="checkbox"/>	vpc1-attach	tgw-attach-05a4584c571a8d7a1	tgw-0ec2227a17d6d8fd8	Pending
<input type="checkbox"/>	vpc2-attach	tgw-attach-07928946dc0fddcc9	tgw-0ec2227a17d6d8fd8	Pending

Seleccionar una conexión de gateway de tránsito

Y las tablas de enrutamiento asociadas ya en el GT:

Tablas de enrutamiento de gateway de tránsito (1/1) [información](#)

Buscar tabla de enrutamiento de puerta de enlace de tránsito por atributo o etiqueta

<input checked="" type="checkbox"/>	Name	ID de la tabla de enrutamiento...	ID de la puerta de enlace ...	Estado	Tabla de en
<input checked="" type="checkbox"/>	default-TG	tgw-rtb-0a86cabdab6c10abb	tgw-0ec2227a17d6d8fd8	Available	Yes

Tablas de enrutamiento de puerta de enlace de tránsito: [tgw-rtb-0a86cabdab6c10abb](#) / default-TG

Detalles **Asociaciones** Propagaciones Referencias de lista de prefijos Rutas Etiquetas

Asociaciones (3) [información](#)

Buscar asociación por atributo o etiqueta

<input type="checkbox"/>	ID de conexión	Tipo de recurso	ID de recurso	Estado
<input type="checkbox"/>	tgw-attach-07928946dc0fddcc9	VPC	vpc-0fb435edf927abfda	Associated
<input type="checkbox"/>	tgw-attach-058df6179957a7e08	VPC	vpc-0db626a354eaf3db0	Associated
<input type="checkbox"/>	tgw-attach-05a4584c571a8d7a1	VPC	vpc-0dc5a07bea62c0bcc	Associated

Ahora, sí que podremos editar las rutas de cada VPC y cuando queramos enlazar con otras VPCs utilizaremos el Transit Gateway.

Vista global de EC2 [🔗](#)

Filter by VPC ▼

Nube virtual privada

Sus VPC

Subredes

Tablas de enrutamiento

Puertas de enlace de Internet

<input type="checkbox"/>	Name ▼	ID de tabla de enrutamiento ▼
<input type="checkbox"/>	vpc3-rtb-public	rtb-0888f856b8eefc090
<input type="checkbox"/>	-	rtb-03661c4d655feb645
<input type="checkbox"/>	-	rtb-068629238bee9f204
<input type="checkbox"/>	vpc2-rtb-public	rtb-0fa941459dfb8cb45
<input type="checkbox"/>	-	rtb-0615df7d70391a95f
<input type="checkbox"/>	vpc1-rtb-public	rtb-0e286221ee90dae5e
<input type="checkbox"/>	-	rtb-0357e4e4c9f125a41

Por ejemplo, para la VPC1:

<input type="checkbox"/>	-	rtb-0615df7d70391a95f	-	-	Sí	vpc-0e25fcb...	150114
<input checked="" type="checkbox"/>	vpc1-rtb-public	rtb-0e286221ee90dae5e	subnet-0ecc82f329f...	-	No	vpc-0dc5a07...	150114
<input type="checkbox"/>	-	rtb-0357e4e4c9f125a41	-	-	Sí	vpc-0db626a...	150114

rtb-0e286221ee90dae5e / vpc1-rtb-public

Detalles **Rutas** Asociaciones de subredes Asociaciones de borde Propagación de rutas Etiquetas

Rutas (2)

< 1 > ⚙️

Editaremos las rutas, y añadiremos una general:

Destino	Destino	Estado
10.1.0.0/16	local	🟢 Activo
<input type="text" value="0.0.0.0/0"/>	<input type="text" value="local"/>	
	Puerta de enlace de Internet	🟢 Activo
<input type="text" value="10.0.0.0/8"/>	<input type="text" value="igw-02be150eef5add96d"/>	
	Gateway de tránsito	-
	<input type="text" value="tgw-0ec2227a17d6d8fd8"/>	

Quedando tal y como se muestra:

<input type="text" value="10.0.0.0/8"/>	<input type="text" value="igw-02be150eef5add96d"/>	
	Gateway de tránsito	-
	<input type="text" value="tgw-0ec2227a17d6d8fd8"/>	

Una vez ya lo tengamos configurado podremos entrar en cualquier ec2 y desde ese acceder a cualquier otro ec2 de otras VPCs

```
Agent pid 41526
jorge@mjolinir Downloads % ssh-add vo1.pem
Identity added: vo1.pem (vo1.pem)
jorge@mjolinir Downloads % ssh-add vo2.pem
Identity added: vo2.pem (vo2.pem)
jorge@mjolinir Downloads % ssh-add vo3.pem
Identity added: vo3.pem (vo3.pem)
jorge@mjolinir Downloads % ssh-add -l
2048 SHA256:3wH3zMoeu0Q+xCGtAvYnhBxIUEqtTCX1LS3ARziVZ/g vo1.pem (RSA)
2048 SHA256:Qma90tImmXXprBE+4TitqELzrmgYFMxX01QXPVf0aos vo2.pem (RSA)
2048 SHA256:6jPg4jsheARB546AI71j69SeeIMbqCvGuB9gA7NpoOc vo3.pem (RSA)
jorge@mjolinir Downloads % ssh -A ec2-user@3.92.180.125

#_
~\_ #####_ Amazon Linux 2023
~~~ \_#####\
~~~ \_###|
~~~ \#/ ---- https://aws.amazon.com/linux/amazon-linux-2023
~~~ V~' '--->
~~~
~~~ * _ _ _ _ _
~~~ _/ _/ _/ _/ _/
~~~ _/m/ '

Last login: Sun Sep 22 17:11:55 2024 from 79.109.194.227
[ec2-user@ip-10-1-0-10 ~]$ ssh -A ec2-user@3.83.167.79

#_
~\_ #####_ Amazon Linux 2023
~~~ \_#####\
~~~ \_###|
~~~ \#/ ---- https://aws.amazon.com/linux/amazon-linux-2023
~~~ V~' '--->
```

Recordatorio:

La idea del Transit gateway es utilizarlo cuando tengo que interconectar muchas VPCs a modo de “switch” y centralizar toda esa faena es a efectos económicos más costoso, pero más fácil de centralizar todo el trabajo, mientras que el peer connection establece una relación uno a uno por lo que será más barato y mejor para entorno más sencillos pero cuando se complica parece una red multimalla donde todos los elementos se tienen que interconectar.

Tareas

1. Configurar la comunicación entre tres VPCs:

- **VPC-A y VPC-B** mediante un **Transit Gateway**.
- **VPC-A y VPC-C** mediante una **VPC Peering Connection**.

Tres VPCs en la misma o diferentes regiones:

- VPC-A: 10.0.0.0/16
- VPC-B: 10.1.0.0/16
- VPC-C: 10.2.0.0/16

Lanza instancias EC2 en cada VPC. Usa ping o telnet para verificar conectividad entre:

- EC2 en VPC-A ↔ VPC-B (mediante TGW).
- EC2 en VPC-A ↔ VPC-C (mediante Peering).

Tarea 2

2. Monta un escenario donde trabajando en dos regiones habrá en cada región: dos vpcs con sus correspondientes subredes interconectadas por un Transit Gateway. Para interconectar las regiones colocaremos un peer connection. Monta la infraestructura de manera que podamos hacer un ping entre cada dispositivo. En la tarea adjunta captura de las tablas de rutas, de los Transit gateways y peer connection creado y de los pings funcionando. Como pista decir que a los Transit habría que añadir rutas estáticas y que el peer se crea desde el Transit como se explicó en la primera parte del tema.

