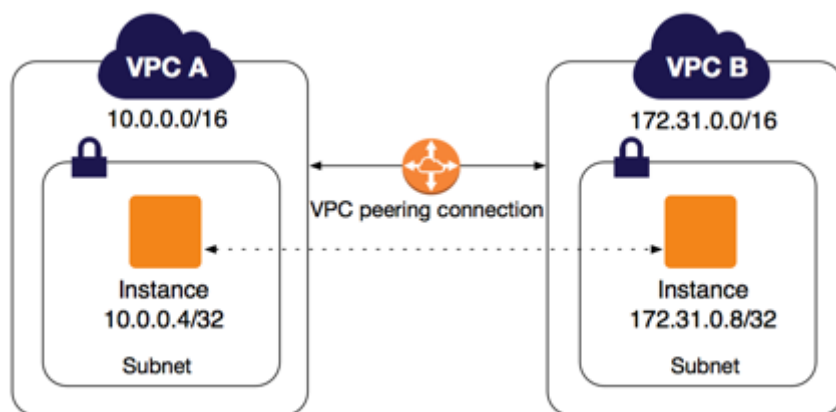


## AWS – Tutorial de Peering Connection

Fala galerinha da arquitetura de soluções em Cloud meu primeiro conteúdo sobre como montar uma rede VPC segura e criar seu primeiro Peering Connection.

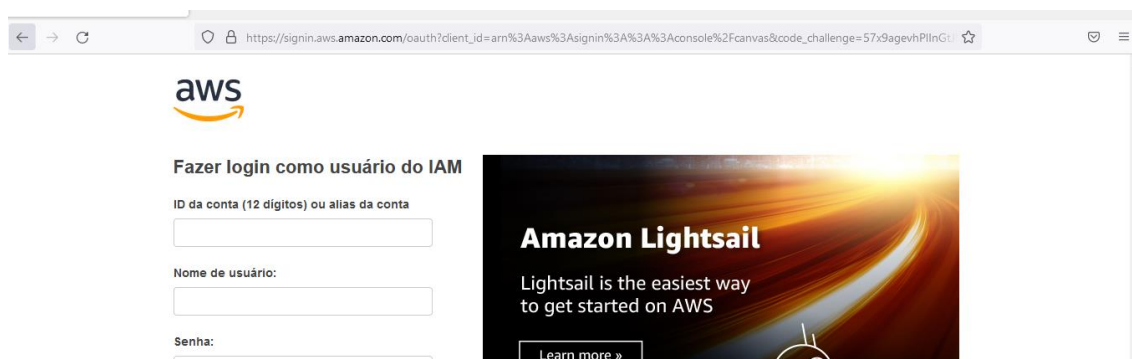
Primeiramente o que é um Peering Connection?

Segundo a AWS um PC (Peering Connection) é uma conexão de emparelhamento de VPC é uma conexão de rede entre duas VPCs que permite rotear o tráfego entre elas usando endereços IPv4 privados ou endereços IPv6. As instâncias em qualquer VPC podem se comunicar umas com as outras como se estivessem na mesma rede. Você pode criar uma conexão de emparelhamento de VPC entre suas próprias VPCs ou com uma VPC em outra conta da AWS. As VPCs podem estar em regiões diferentes (também conhecidas como conexão de emparelhamento de VPC entre regiões).



Neste tutorial mostrarei como criar uma conexão padrão e como testar se as conexões entre as VPCs estão conectando e respondendo a resposta. Vamos lá?

Primeiro passo, faça login no console:



Segundo passo, acesse o menu VPC na procura da console:

The screenshot shows the AWS Management Console VPC Dashboard. On the left is a navigation sidebar with 'VIRTUAL PRIVATE CLOUD' expanded, showing 'Your VPCs' as the selected option. The main content area is titled 'Resources by Region' and shows a grid of resource counts for 'US East'. The resources listed are: VPCs (1), NAT Gateways (0), Subnets (6), VPC Peering Connections (1), Route Tables (1), Network ACLs (1), Internet Gateways (1), Security Groups (2), Egress-only Internet Gateways (0), and Customer Gateways (0). On the right, the 'Service Health' section shows 'Amazon EC2 - US East' with a status of 'Service is operating normally'. Below that are 'Settings' and 'Additional Information' sections.

Resource Type	Count
VPCs	1
NAT Gateways	0
Subnets	6
VPC Peering Connections	1
Route Tables	1
Network ACLs	1
Internet Gateways	1
Security Groups	2
Egress-only Internet Gateways	0
Customer Gateways	0

Por padrão a AWS já disponibiliza uma VPC inicial para você inicializar os seus recursos na cloud, porém o que vamos fazer como boa prática neste material é a conta AWS de produção irá se conectar com a conta AWS de desenvolvimento que deixaremos a VPC padrão criada pela própria AWS, facilitando assim a conexão, pois existem questões de overlap de CIDR neste ambiente.

Terceiro passo, acesse o menu Your VPCs e clique em create VPC

The screenshot shows the 'Your VPCs' page in the AWS Management Console. The page title is 'Your VPCs (2)'. There is a search bar and a 'Filter VPCs' button. Below the search bar is a table with the following columns: Name, VPC ID, State, IPv4 CIDR, and IPv6 CIDR. The table contains two rows: 'Csa\_Manager' with VPC ID 'vpc-0df320fd894581437' and 'Default' with VPC ID 'vpc-083be973e780cc05a'. Both are in 'Available' state. Below the table is a section titled 'Select a VPC above' with three icons.

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
Csa_Manager	vpc-0df320fd894581437	Available	10.0.0.0/16	-
Default	vpc-083be973e780cc05a	Available	172.31.0.0/16	-

Quarto passo vamos definir as primeiras configurações da nossa VPC, utilize o site [IP Subnet Calculator for IPv4 | Online Subnet Mask Calculator - Site24x7](https://www.site24x7.com/tools/ip-subnet-calculator.html) para criar ideias de subnet e IPs para associação.

Após definição de qual IP utilizar conforme a imagem, vamos configurar no console AWS

**Subnet Calculator for IPv4**

The IP Subnet Calculator performs subnet calculations for the given network address block, subnet mask, maximum required hosts per subnet and determines the resulting broadcast address, subnet, Cisco wildcard mask and host range.

Network Address Block	10.0.0.0/16	Host Address Range	10.0.0.1 - 10.0.63.254
Subnet Mask	255.255.192.0/18	Broadcast Address	10.0.63.255
No. of Hosts/Subnet	16384	Wildcard Mask	0.0.63.255
Number of Subnets	4	CIDR Notation	10.0.0.0/18

**Subnet Details**

Subnet ID	Subnet Address	Host Address Range	Broadcast Address
1	10.0.0.0	10.0.0.1 - 10.0.63.254	10.0.63.255
2	10.0.64.0	10.0.64.1 - 10.0.127.254	10.0.127.255
3	10.0.128.0	10.0.128.1 - 10.0.191.254	10.0.191.255
4	10.0.192.0	10.0.192.1 - 10.0.255.254	10.0.255.255

Quinto passo vamos configurar as definições realizadas no console AWS

Create only the VPC resource or create VPC, subnets, etc.

☒ VPC only ☐ VPC, subnets, etc.

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

vpc\_teste

IPv4 CIDR block [Info](#)

☒ IPv4 CIDR manual input ☐ IPAM-allocated IPv4 CIDR block

IPv4 CIDR

10.0.0.0/16

IPv6 CIDR block [Info](#)

☒ No IPv6 CIDR block ☐ IPAM-allocated IPv6 CIDR block ☐ Amazon-provided IPv6 CIDR block ☐ IPv6 CIDR owned by me

Tenancy [Info](#)

Default

**Tags**

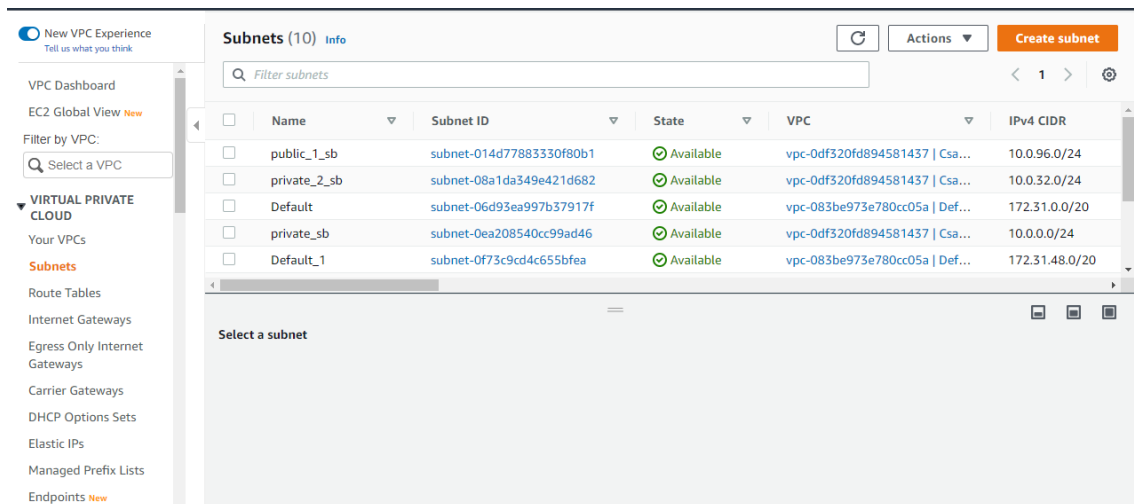
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - <i>optional</i>	
Q Name	Q vpc_teste	Remove

Add new tag

Clique em create VPC que retornara para o menu principal com a sua vpc configurada.

Sexto passo clique em subnet conforme imagem abaixo:



Ao clicar no menu subnets veremos todas as subnets que temos criadas pela AWS, mas como criamos a nossa própria VPC, vamos criar novas subnets para aplicar o conceito do Peering, clique em create subnet

**Subnet settings**  
Specify the CIDR blocks and Availability Zone for the subnet.

**Subnet 1 of 1**

**Subnet name**  
Create a tag with a key of 'Name' and a value that you specify.  
  
The name can be up to 256 characters long.

**Availability Zone** [Info](#)  
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

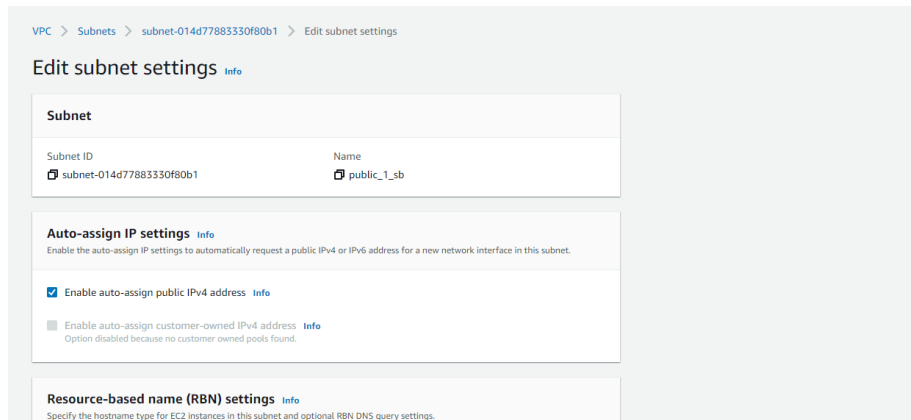
**IPv4 CIDR block** [Info](#)

**Tags - optional**

Key	Value - optional	
<input type="text" value="Name"/>	<input type="text" value="Private"/>	<input type="button" value="Remove"/>

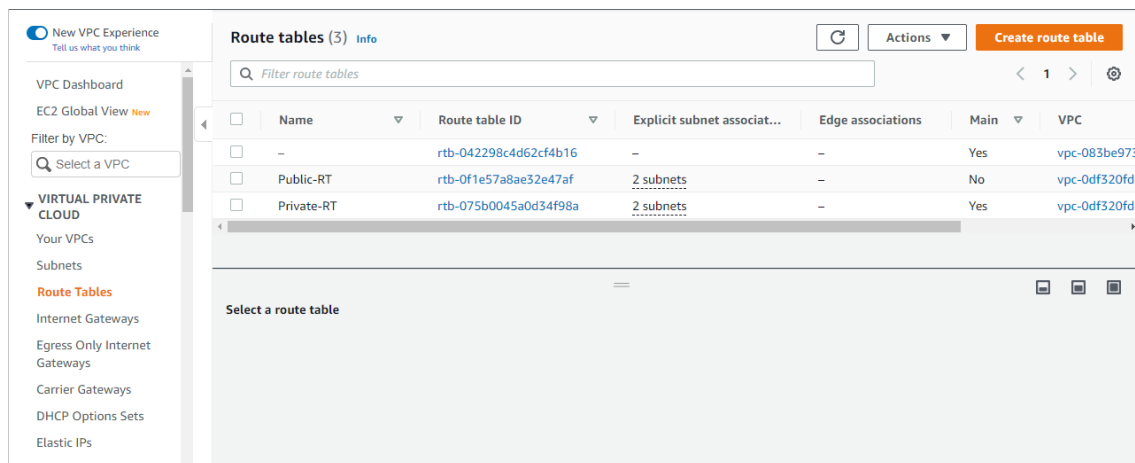
You can add 49 more tags.

Faça este processo diversas vezes dependendo de quantas subnets você gostaria de ter em sua nova VPC, caso você queria criar VPC abertas para a internet após configurar a sua Subnet selecione-a e vá no menu Edit subnet setting aparecera a configuração da sua subnet, deixe selecionado a opção conforme imagem



Após configurar as Subnets vamos configurar as Rout Tables, elas são as veias do nosso corpo humano, direcionando onde cada tráfego e como cada recurso irá se comunicar uns com os outros, aqui vamos selecionar 2 subnets como privadas e 2 como públicas, portanto vamos ter que configurar duas Route Tables para cada Subnet selecionada, Bora lá? 😊

Passo sétimo clique em Route Tables



Após configurar todas as informações selecione a sua VPC criada e clique em Creat Route table

**Route table settings**

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

Private-RT

**VPC**  
The VPC to use for this route table.

vpc-0df320fd894581437 (Csa\_Manager)

**Tags**

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key

Value - optional

Q Name X

Q Private-RT X

Remove

Add new tag

You can add 49 more tags.

Cancel

Create route table

Com as suas Route tables configuradas vamos para definição de quem será exposta para internet e qual não será, lembrete neste ponto, subnet exposta podemos expor nossos recursos a vulnerabilidades de segurança da informação, configure sempre o máximo de segurança e tente deixar o ambiente o mais blindado possível, para que cada instância não seja violada.

Oitavo passo, clique na sua RT criada e selecione a opção Edit subnet association

New VPC Experience  
Tell us what you think

VPC Dashboard

EC2 Global View New

Filter by VPC:  
Select a VPC

VIRTUAL PRIVATE CLOUD

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

Carrier Gateways

DHCP Options Sets

Elastic IPs

Managed Prefix Lists

Endpoints New

Route tables (1/3) Info

Filter route tables

< 1 > ⚙

	Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC
<input type="checkbox"/>	-	rtb-042298c4d62cf4b16	-	-	Yes	vpc-083be9
<input checked="" type="checkbox"/>	Public-RT	rtb-0f1e57a8ae32e47af	2 subnets	-	No	vpc-0df320f

rtb-0f1e57a8ae32e47af / Public-RT

Details

Routes

Subnet associations

Edge associations

Route propagation

Tags

Explicit subnet associations (2)

Edit subnet associations

Find subnet association

< 1 > ⚙

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-014d77883330f80b1 / public_1_sb	10.0.96.0/24	-

Após selecionar a opção teremos a seguinte mensagem:

VPC > Route tables > rtb-0f1e57a8ae32e47af > Edit subnet associations

### Edit subnet associations

Change which subnets are associated with this route table.

**Available subnets (2/4)**

Filter subnet associations

	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/>	public_1_sb	subnet-014d7788330f80b1	10.0.96.0/24	-	rtb-0f1e57a8ae32e47af / Public-RT
<input type="checkbox"/>	private_2_sb	subnet-08a1da349e421d682	10.0.32.0/24	-	rtb-075b0045a0d34f98a / Private-RT
<input type="checkbox"/>	private_sb	subnet-0ea208540cc99ad46	10.0.0.0/24	-	rtb-075b0045a0d34f98a / Private-RT
<input checked="" type="checkbox"/>	Public_sb	subnet-0c80ca04a7b7fca1d	10.0.64.0/24	-	rtb-0f1e57a8ae32e47af / Public-RT

**Selected subnets**

subnet-0c80ca04a7b7fca1d / Public\_sb X subnet-014d7788330f80b1 / public\_1\_sb X

Selecione as subnets que você queira definir na sua RT após selecionar clique em save.

Nono passo, vamos coletar as informações das duas VPCs criadas para nossa conexão de Peering, você localiza estas informações conforme imagem abaixo:

New VPC Experience Not used previously

VPC Dashboard EC2 Global View

Filter by VPC: Select a VPC

**Virtual Private Cloud**

- Your VPCs
- Subnets
- Route Tables
- Internet Gateways
- Egress Only Internet Gateways
- Carrier Gateways
- DHCP Options Sets
- Elastic IPs
- Managed Prefix Lists
- Endpoints
- Endpoint Services
- NAT Gateways
- Peering Connections

**SECURITY**

- Network ACLs
- Security Groups

**NETWORK ANALYSIS**

- Reachability Analyzer
- Network Anomaly

### Your VPCs (1/2)

Filter VPCs

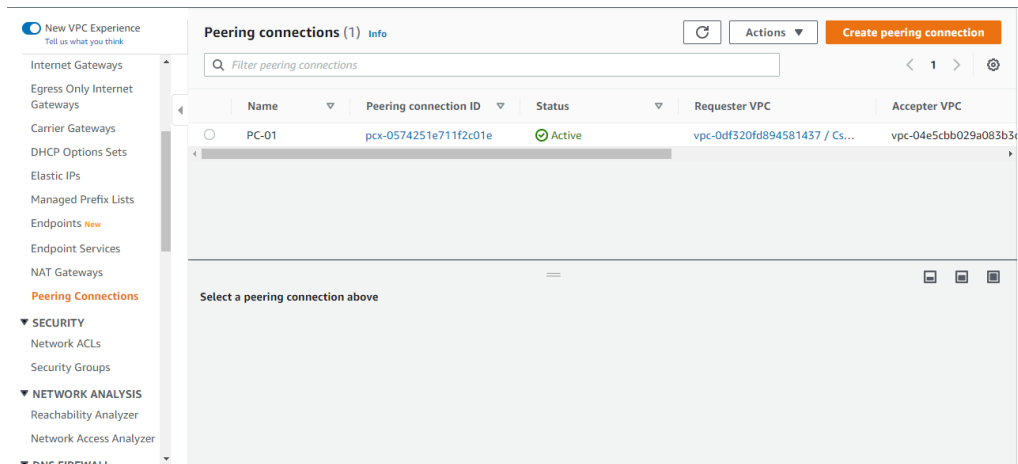
	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Main route table	Main network
<input checked="" type="checkbox"/>	Csa_Manager	vpc-0df320f894581437	Available	10.0.0.0/16	-	dopt-0cfa0650bb8459...	rtb-075b0045a0d34f98a ...	acl-09f4f4267
<input type="checkbox"/>	Default	vpc-083be973e780cc05a	Available	172.31.0.0/16	-	dopt-0cfa0650bb8459...	rtb-042298c4d62cfd016	acl-0a700a8a5

### Details

VPC ID vpc-0df320f894581437	State Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP options set dopt-0cfa0650bb8459b1e	Main route table rtb-075b0045a0d34f98a / Private-RT	Main network ACL acl-09f4f4267ce24f9ae
Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool -	IPv6 CIDR (Network border group) -
Route 53 Resolver DNS Firewall rule groups -	Owner ID 402152557044		

As opções que vamos coletar são: VPC ID e Owner ID.

Décimo passo com as informações coletadas conforme passo nono, vamos configurar o nosso Peering connection, no menu clique em Peering connection



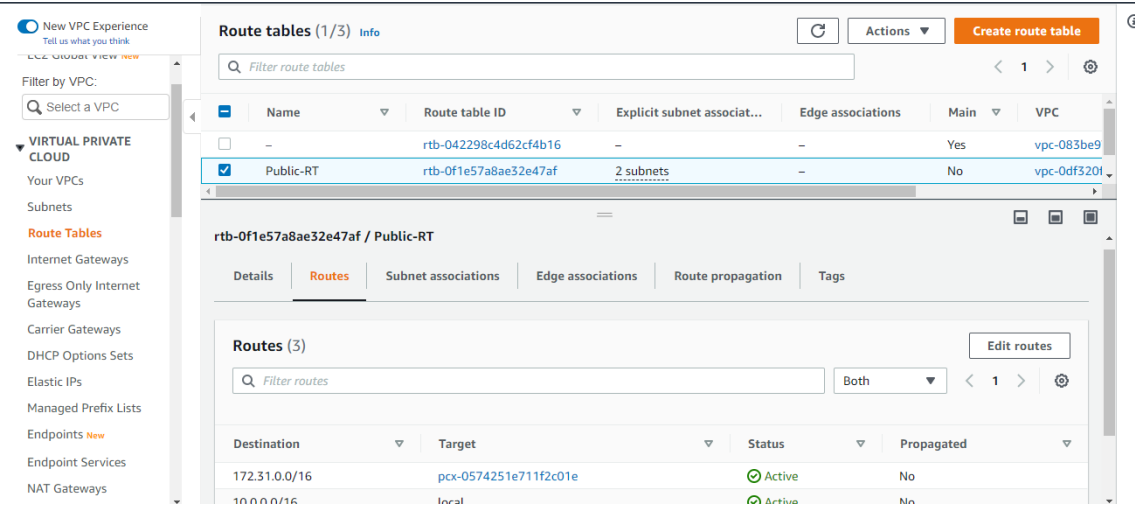
Clique em Create Peering connection, após isto vamos configurar as informações:

The 'Peering connection settings' form contains the following fields and options:

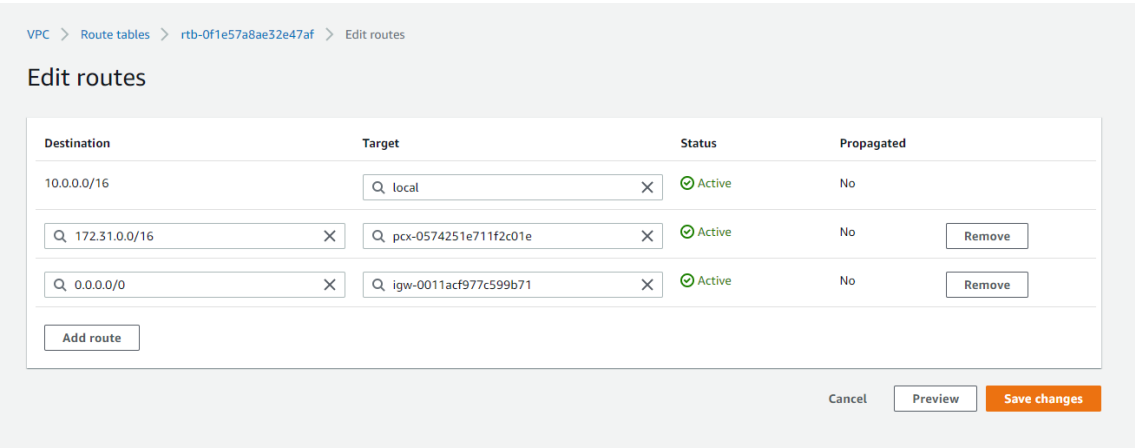
- Name - optional:** A text input field with the value 'my-pc-01'.
- Select a local VPC to peer with:** A dropdown menu labeled 'VPC ID (Requester)' with the option 'Select a VPC'.
- Select another VPC to peer with:**
  - Account:** Radio buttons for 'My account' and 'Another account' (selected).
  - Account ID:** A text input field with the value 'Account ID'.
  - Region:** Radio buttons for 'This Region (us-east-1)' (selected) and 'Another Region'.
  - VPC ID (Accepter):** A text input field with the value 'VPC ID'.



Defina um nome, na VPC selecione a VPC que queria conectar da sua conta, após isto, selecionar a opção Another account e adicione as informações coletadas no item nono. Com o Peering definido acessa a sua outra conta AWS e aceite a conexão solicitada, siga o próximo passo para que possamos conectar as duas contas, acesse novamente a opção Route Table, e clique em Routes e edite a regras da RT, conforme imagem:



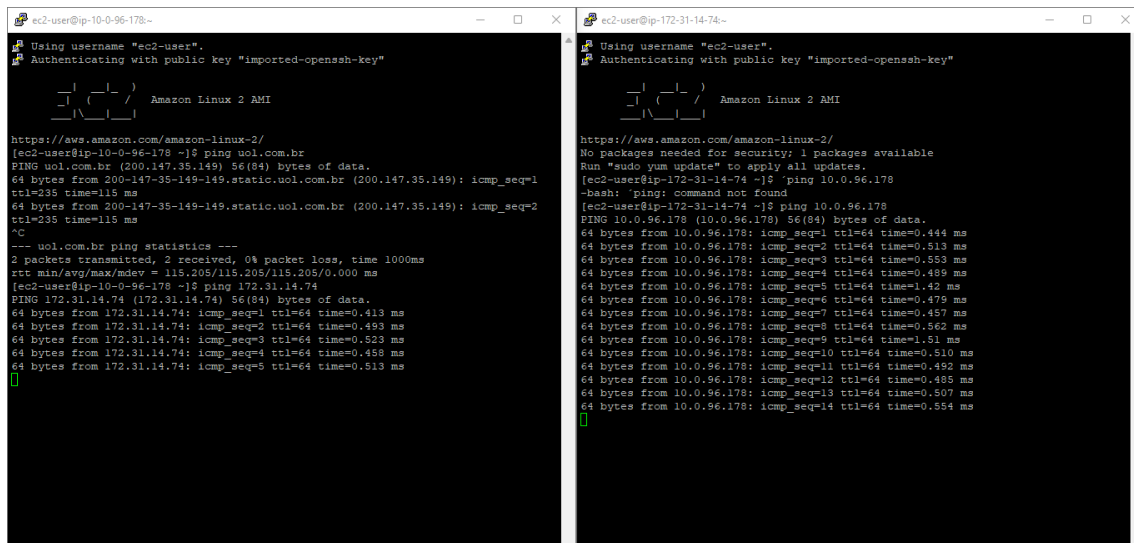
Vamos editar as regras,



Conforme imagem acima, configure conforme a imagem em ambas as suas contas AWS, fazendo a leitura temos:

- 1 IP da VPC
- 2 IP da nossa conexão Peering connection que criamos
- 3 IP Padrão do Internet gateway que ajudará na configuração e acesso aos nossos recursos.

Com as configurações realizadas vamos testar a conexão, lance duas instâncias uma em cada conta AWS, e vamos validar se as conexões estão sendo aceitas,



The image shows two terminal windows side-by-side. The left window is titled 'ec2-user@ip-10-0-96-178~' and shows a successful ping test to 'uol.com.br' with a response time of 115ms. The right window is titled 'ec2-user@ip-172-31-14-74~' and shows a successful ping test to '10.0.96.178' with a response time of 0.444ms. Both windows show the Amazon Linux 2 AMI logo and the command prompt history.

```
ec2-user@ip-10-0-96-178~  
Using username "ec2-user".  
Authenticating with public key "imported-openssh-key"  
  
Amazon Linux 2 AMI  
  
https://aws.amazon.com/amazon-linux-2/  
[ec2-user@ip-10-0-96-178 ~]$ ping uol.com.br  
PING uol.com.br (200.147.35.149) 56(84) bytes of data.  
64 bytes from 200-147-35-149.static.uol.com.br (200.147.35.149): icmp_seq=1  
ttl=235 time=115 ms  
64 bytes from 200-147-35-149.static.uol.com.br (200.147.35.149): icmp_seq=2  
ttl=235 time=115 ms  
^C  
--- uol.com.br ping statistics ---  
2 packets transmitted, 2 received, 0% packet loss, time 1000ms  
rtt min/avg/max/mdev = 115.205/115.205/115.205/0.000 ms  
[ec2-user@ip-10-0-96-178 ~]$ ping 172.31.14.74  
PING 172.31.14.74 (172.31.14.74) 56(84) bytes of data.  
64 bytes from 172.31.14.74: icmp_seq=1 ttl=64 time=0.413 ms  
64 bytes from 172.31.14.74: icmp_seq=2 ttl=64 time=0.493 ms  
64 bytes from 172.31.14.74: icmp_seq=3 ttl=64 time=0.523 ms  
64 bytes from 172.31.14.74: icmp_seq=4 ttl=64 time=0.458 ms  
64 bytes from 172.31.14.74: icmp_seq=5 ttl=64 time=0.513 ms  
^C  
[
```

```
ec2-user@ip-172-31-14-74~  
Using username "ec2-user".  
Authenticating with public key "imported-openssh-key"  
  
Amazon Linux 2 AMI  
  
https://aws.amazon.com/amazon-linux-2/  
No packages needed for security: 1 packages available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-14-74 ~]$ ping 10.0.96.178  
-bash: ping: command not found  
[ec2-user@ip-172-31-14-74 ~]$ ping 10.0.96.178  
PING 10.0.96.178 (10.0.96.178) 56(84) bytes of data.  
64 bytes from 10.0.96.178: icmp_seq=1 ttl=64 time=0.444 ms  
64 bytes from 10.0.96.178: icmp_seq=2 ttl=64 time=0.513 ms  
64 bytes from 10.0.96.178: icmp_seq=3 ttl=64 time=0.553 ms  
64 bytes from 10.0.96.178: icmp_seq=4 ttl=64 time=0.489 ms  
64 bytes from 10.0.96.178: icmp_seq=5 ttl=64 time=1.42 ms  
64 bytes from 10.0.96.178: icmp_seq=6 ttl=64 time=0.479 ms  
64 bytes from 10.0.96.178: icmp_seq=7 ttl=64 time=0.457 ms  
64 bytes from 10.0.96.178: icmp_seq=8 ttl=64 time=0.562 ms  
64 bytes from 10.0.96.178: icmp_seq=9 ttl=64 time=1.51 ms  
64 bytes from 10.0.96.178: icmp_seq=10 ttl=64 time=0.510 ms  
64 bytes from 10.0.96.178: icmp_seq=11 ttl=64 time=0.492 ms  
64 bytes from 10.0.96.178: icmp_seq=12 ttl=64 time=0.485 ms  
64 bytes from 10.0.96.178: icmp_seq=13 ttl=64 time=0.507 ms  
64 bytes from 10.0.96.178: icmp_seq=14 ttl=64 time=0.554 ms  
^C  
[
```

Obtivemos sucesso na conexão a máquina ao lado esquerdo é a máquina de produção e a do lado direito é do ambiente de desenvolvimento, com isto conseguiremos criar a nossa peering connection entre as contas e os nossos recursos.

Gostou? Avalia e comenta o que posso fazer para melhorar este manual !!!

Abraço! ☺