

VPC Peering

In this project, **VPC Peering** we're going to level up by setting up VPC Peering - we're going to play with TWO VPCs instead of one!

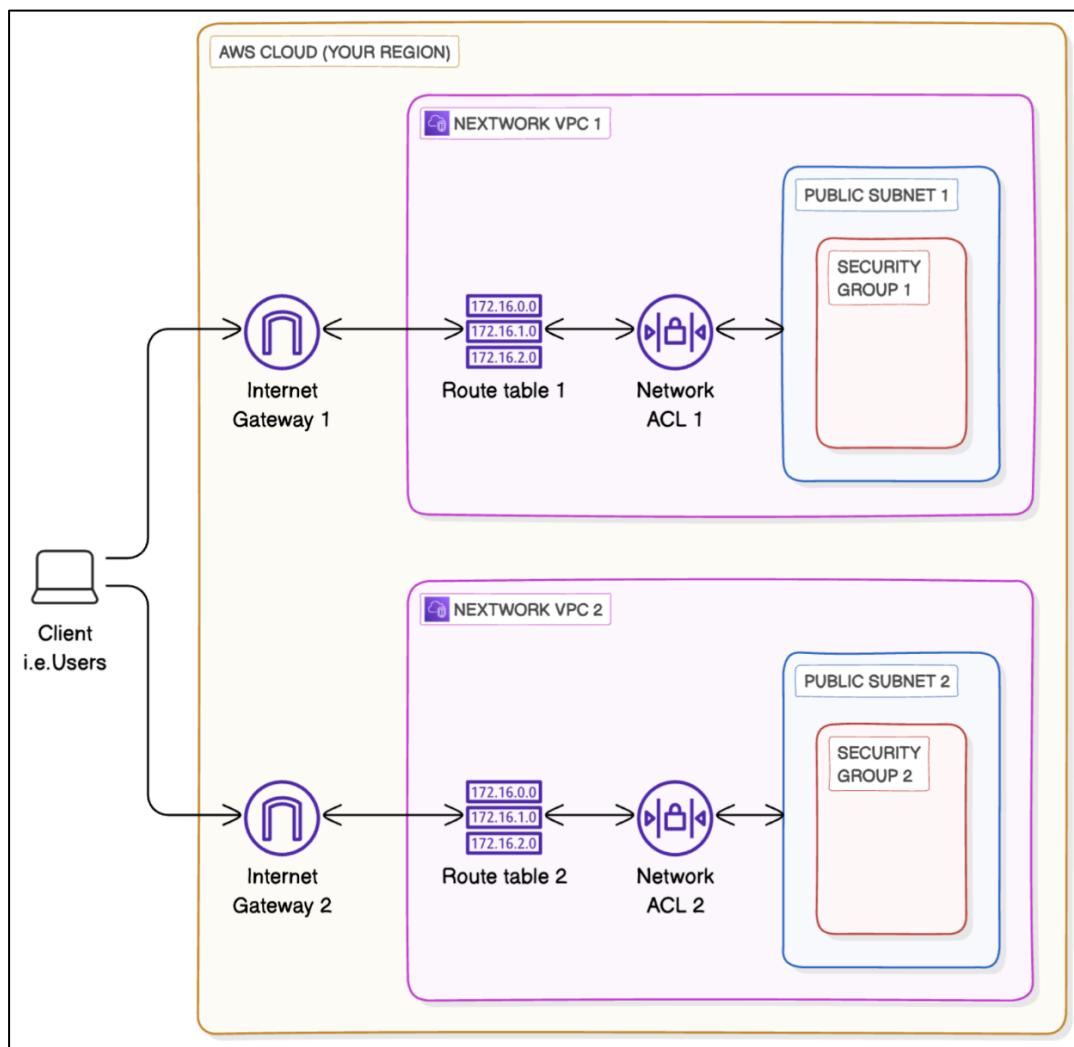
Aim:

1. Set up multiple VPCs.
2. Create a VPC peering connection - i.e. get two VPCs to talk to each other!
3. Test VPC peering with connectivity tests.

[Set up your VPCs](#)

In this step, you're going to:

1. Create two VPCs from scratch!
2. Use the visual VPC resource map to create your VPCs



- Log in to your AWS Account.
- Head to your **VPC** console - search for VPC at the search bar at top of your page.
- From the left hand navigation bar, select **Your VPCs**.
- Select **Create VPC**.
- let's select **VPC and more**.

Create VPC 1

- Under **Name tag auto-generation**, enter **NextWork-1**
- The VPC's **IPv4 CIDR block** is already pre-filled to **10.0.0.0/16** - change that to **10.1.0.0/16**
- For **IPv6 CIDR block**, we'll leave in the default option of **No IPv6 CIDR block**.
- For **Tenancy**, we'll keep the selection of **Default**.

VPC > Your VPCs > Create VPC

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances, Amazon RDS databases, and Amazon S3 buckets.

VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.

VPC only VPC and more

Name tag auto-generation Info

Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

Auto-generate
NextWork-1

IPv4 CIDR block Info

Determine the starting IP and the size of your VPC using CIDR notation.

10.1.0.0/16	65,536 IPs
-------------	------------

IPv6 CIDR block Info

No IPv6 CIDR block Amazon-provided IPv6 CIDR block

Tenancy Info

Default ▾

- For **Number of Availability Zones (AZs)**, we'll use just **1** Availability Zone.

- Make sure the **Number of public subnets** chosen is **1**.
- For **Number of private subnets**, we'll keep things simple today and go with **0** private subnets.

Number of Availability Zones (AZs) [Info](#)
Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.

1 **2** **3**

► Customize AZs

Number of public subnets [Info](#)
The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.

0 **1**

Number of private subnets [Info](#)
The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.

0 **1** **2**

- Next, for the **NAT gateways (\$)** option, make sure you've selected **None**. As the dollar sign suggests, NAT gateways cost money!
- Next, for the **VPC endpoints** option, select **None**.
- You can leave the **DNS options** checked.

EC2

[VPC](#) > [Your VPCs](#) > [Create VPC](#)

NAT gateways (\$) [Info](#)
Choose the number of Availability Zones (AZs) in which to create NAT gateways.
Note that there is a charge for each NAT gateway

None **In 1 AZ** **1 per AZ**

VPC endpoints [Info](#)
Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.

None **S3 Gateway**

DNS options [Info](#)
 Enable DNS hostnames
 Enable DNS resolution

► Additional tags

[Cancel](#) [Preview code](#) [Create VPC](#)

- Select **Create VPC**.

VPC > Your VPCs > Create VPC > Create VPC resources

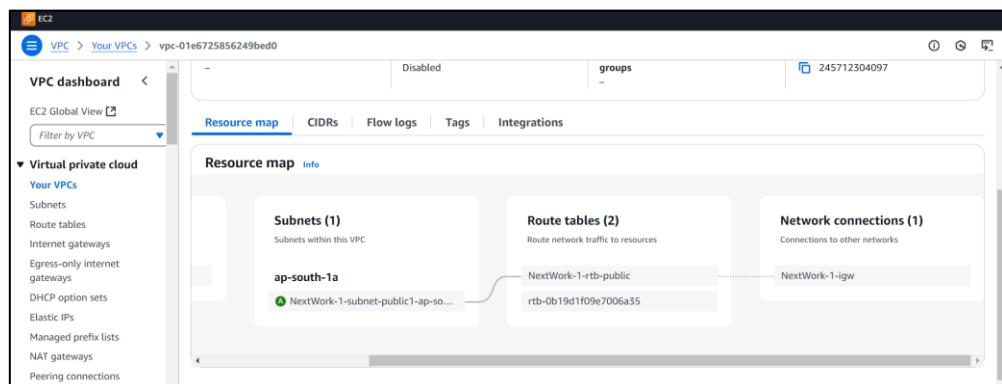
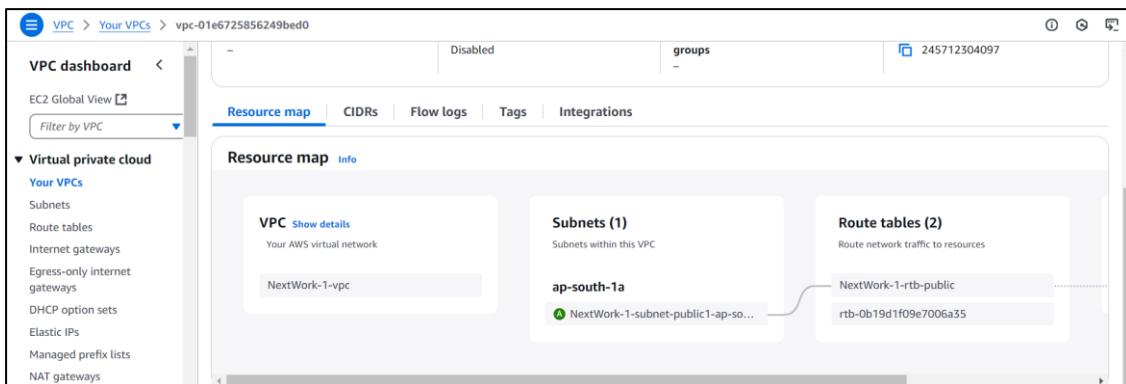
Create VPC workflow

Success

▼ Details

- ✓ Create VPC: vpc-01e6725856249bed0
- ✓ Enable DNS hostnames
- ✓ Enable DNS resolution
- ✓ Verifying VPC creation: vpc-01e6725856249bed0
- ✓ Create subnet: subnet-06bfed490a4382234
- ✓ Create internet gateway: igw-04d6646d1af356e1b
- ✓ Attach internet gateway to the VPC
- ✓ Create route table: rtb-00f6e413ea8c75d72
- ✓ Create route
- ✓ Associate route table
- ✓ Verifying route table creation

- Select **View VPC**.
- Select the **Resource map** tab - nice, all of these resources have been set up for you in a flash!



Set up VPC 2

We're going to set up another VPC with *slightly* different settings

- Select **Create VPC**.
- Select **VPC and more**.
- Under **Name tag auto-generation**, enter **NextWork-2**
- The VPC's **IPv4 CIDR block** should be unique! Make sure the CIDR block is NOT 10.1.0.0/16 - it should be **10.2.0.0/16**

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as EC2 instances, Amazon RDS databases, and Amazon S3 buckets.

VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.

VPC only VPC and more

Name tag auto-generation Info
Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

Auto-generate
NextWork-2

IPv4 CIDR block Info
Determine the starting IP and the size of your VPC using CIDR notation.

10.2.0.0/16 65,536 IPs

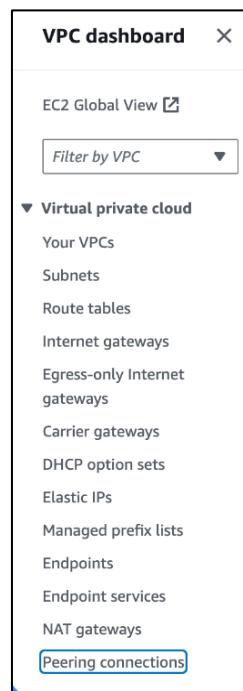
- For **IPv6 CIDR block**, we'll leave in the default option of **No IPv6 CIDR block**.
- For **Tenancy**, we'll keep the selection of **Default**.
- For **Number of Availability Zones (AZs)**, we'll use just **1** Availability Zone.
- Make sure the **Number of public subnets** chosen is **1**.
- For **Number of private subnets**, we'll go with **0** for today's project. Let's keep it simple with just a single subnet!
- For the **NAT gateways (\$)** option, select **None**.
- For the **VPC endpoints** option, select **None**.
- You can leave the **DNS options** checked.
- Select **Create VPC**.

Create a Peering Connection

Now that you have two VPCs ready to go, let's bridge them together with a peering connection.

In this step, you're going to:

1. Set up a connection link between your VPCs.
- Still in the VPC console, click on **Peering connections** on the left hand navigation panel.



- Click on **Create peering connection** in the right hand corner.



- Name your **Peering connection name** as VPC 1 \leftrightarrow VPC 2
- Select **NextWork-1-VPC** for your **VPC ID (Requester)**.

A screenshot of the 'Create peering connection' configuration form. It has three main sections: 'Peer connection settings' (with a 'Name - optional' field containing 'VPC 1 <=> VPC 2'), 'Select a local VPC to peer with' (with a dropdown menu showing 'vpc-01e6725856249bed0 (NextWork-1-vpc)'), and 'VPC CIDRs for vpc-01e6725856249bed0 (NextWork-1-vpc)' (a table with one row showing CIDR 10.1.0.0/16 and Status 'Associated').

- Under **Select another VPC to peer with**, make sure **My Account** is selected.
- For **Region**, select **This Region**.
- For **VPC ID (Acceptor)**, select **NextWork-2-VPC**

Select another VPC to peer with

Account		
<input checked="" type="radio"/> My account		
<input type="radio"/> Another account		
Region		
<input checked="" type="radio"/> This Region (ap-south-1)		
<input type="radio"/> Another Region		
VPC ID (Acceptor)		
vpc-08f51850f21517ffe (NextWork-2-vpc)		
VPC CIDRs for vpc-08f51850f21517ffe (NextWork-2-vpc)		
CIDR	Status	Status reason
10.2.0.0/16	<input checked="" type="checkbox"/> Associated	-

- Click on **Create peering connection**.
- Your newly created peering connection isn't finished yet! The green success bar says the peering connection **has been requested**.

A VPC peering connection pcx-036a963c1128ac95a / VPC 1 <> VPC 2 has been requested.

pcx-036a963c1128ac95a / VPC 1 <> VPC 2

Pending acceptance
You can accept or reject this peering connection request using the 'Actions' menu. You have until Monday, January 13, 2025 at 11:51:00 GMT+5:30 to accept or reject the request, otherwise it expires.

- On the next screen, select **Actions** and then select **Accept request**.
- Click on **Accept request** again on the pop up panel.

Accept VPC peering connection request [Info](#)

Are you sure you want to accept this VPC peering connection request? (pcx-036a963c1128ac95a / VPC 1 <> VPC 2)

Requester VPC vpc-01e6725856249bed0 / NextWork-1-vpc	Acceptor VPC vpc-08f51850f21517ffe / NextWork-2-vpc	Requester CIDRs <input type="checkbox"/> 10.1.0.0/16
Acceptor CIDRs -	Requester Region Mumbai (ap-south-1)	Acceptor Region Mumbai (ap-south-1)
Requester owner ID <input type="checkbox"/> 245712304097 (This account)	Acceptor owner ID <input type="checkbox"/> 245712304097 (This account)	Cancel Accept request

Your VPC peering connection (pcx-036a963c1128ac95a | VPC 1 <> VPC 2) has been established.

To send and receive traffic across this VPC peering connection, you must add a route to the peered VPC in one or more of your VPC route tables.

[Info](#) **Modify my route tables now** X

- Click on **Modify my route tables now** on the top right corner.

Update Route Tables

- With a peering connection all set up  , now it's time for traffic in your VPCs to learn how to use it.

In this step, you're going to:

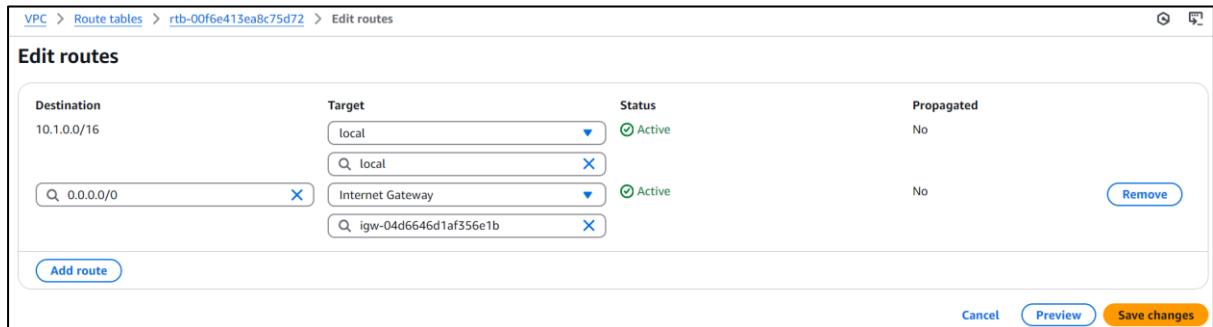
- Set up a way for traffic coming from VPC 1 to get to VPC 2.
- Set up a way for traffic coming from VPC 2 to get to VPC 1.

Update VPC 1's route table

- Select the checkbox next to VPC 1's route table i.e. called **NextWork-1-rtb-public**.

Route tables (7) Info						
<input type="checkbox"/>		Name	Route table ID	Explicit subnet associ...	Edge associations	Main
<input type="checkbox"/>	NextWork-2-rtb-public	rtb-04ee724bbcf3ee793	subnet-0d394ac3655859...	-	No	vpc-08f51...
<input type="checkbox"/>	NextWork-1-rtb-public	rtb-00f6e413ea8c75d72	subnet-06bfed490a4382...	-	No	vpc-01e67...

- Scroll down and click on the **Routes** tab.
- Click **Edit routes**.



VPC > Route tables > rtb-00f6e413ea8c75d72 > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.1.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway	Active	No

Add route

Cancel Preview Save changes

- Let's add a new route!
- Add a new route to **VPC 2** by entering the CIDR block 10.2.0.0/16 as our **Destination**.
- Under Target, select **Peering Connection**.
- Select **VPC 1 <> VPC 2**.

VPC > Route tables > rtb-00f6e413ea8c75d72 > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.1.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway	Active	No
10.2.0.0/16	Peering Connection	-	No
	pcx-036a963c1128ac95a		

[Add route](#) [Remove](#) [Remove](#)

[Cancel](#) [Preview](#) [Save changes](#)

- Click **Save changes**.
- Routes appear in VPC 1's **Routes** tab now!

Routes (3)

[Edit routes](#)

Destination	Target	Status	Propagated
0.0.0.0/0	igw-04d6646d1af356e1b	Active	No
10.1.0.0/16	local	Active	No
10.2.0.0/16	pcx-036a963c1128ac95a	Active	No

Update VPC 2's route table

set up the equivalent route in VPC 2's route table same as you have done above for VPC 1

- The route table you're updating is **NextWork-2-rtb-public**.
- The **Destination** is the CIDR block 10.1.0.0/16

Routes (3)

[Edit routes](#)

Destination	Target	Status	Propagated
0.0.0.0/0	igw-04241c7d57fe8b8b2	Active	No
10.1.0.0/16	pcx-036a963c1128ac95a	Active	No
10.2.0.0/16	local	Active	No

Launch EC2 Instances

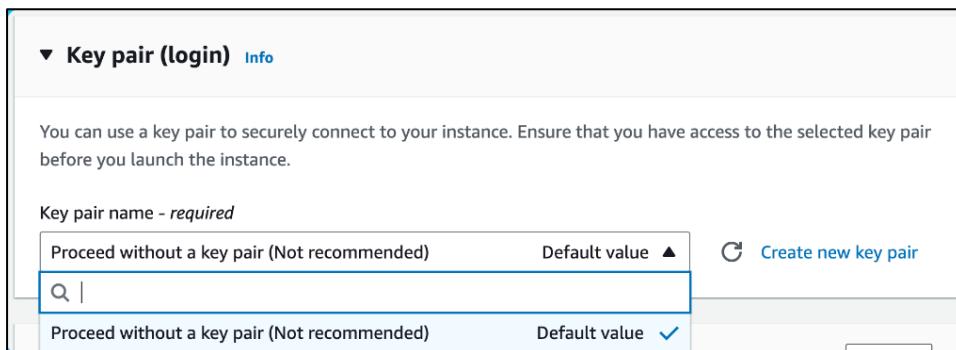
It's time to launch EC2 instances into your architecture!

In this step, you're going to:

1. Launch an EC2 instance in each VPC, so we can use them to test your VPC peering connection later.

Launch an instance in VPC 1

- Head to the **EC2 console** - search for EC2 in the search bar at the top of screen.
- Select **Instances** at the left hand navigation bar.
- Select **Launch instances**.
- Since your first EC2 instance will be launched in your first VPC, let's name it **Instance - NextWork VPC 1**
- For the **Amazon Machine Image**, select **Amazon Linux 2023 AMI**.
- For the **Instance type**, select **t2.micro**.
- For the **Key pair (login)** panel, select **Proceed without a key pair (not recommended)**.



- At the **Network settings** panel, select **Edit** at the right hand corner.
- Under **VPC**, select **NextWork-vpc-1**.
- Under **Subnet**, select your VPC's public subnet.
- Keep the **Auto-assign public IP** setting to **Disable**.
- For the **Firewall (security groups)** setting, Amazon VPC already created a security group for your VPC - let's use that!
- Choose **Select existing security group**.
- Select the **default** security group for your VPC.
- Select **Launch instance**.

VPC - required | [Info](#)

vpc-01e6725856249bed0 (NextWork-1-vpc)
10.1.0.0/16

Subnet | [Info](#)

subnet-06bfed490a4382234 NextWork-1-subnet-public1-ap-south-1a
VPC: vpc-01e6725856249bed0 Owner: 245712304097 Availability Zone: ap-south-1a
Zone type: Availability Zone IP addresses available: 4091 CIDR: 10.1.0.0/20

Create new subnet

Auto-assign public IP | [Info](#)

Disable

Firewall (security groups) | [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

Common security groups | [Info](#)

Select security groups

default sg-05dc57dea66d07d87
VPC: vpc-01e6725856249bed0

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Advanced network configuration

Launch an instance in VPC 2

set up an EC2 instance in VPC 2, use the same instructions above but make sure:

1. The Name is **Instance - NextWork VPC 2**
2. The VPC is **NextWork-vpc-2**.

Instances (2) Info		Last updated less than a minute ago	Connect	Instance state ▾	Actions ▾	Launch instances
<input type="text"/> Find Instance by attribute or tag (case-sensitive)		All states				
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/>	Instance - NextWork VPC 1	i-0bd771d297992ead7	Running	t2.micro	Initializing	View alarms +
<input type="checkbox"/>	Instance - NextWork VPC 2	i-0b065d4262c66036a	Running	t2.micro	Initializing	View alarms +

Connect to Instance 1

To test our VPC peering connection, we'll need to get one of our EC2 instances to try talk to the other.

In this step, you're going to:

1. Use EC2 Instance Connect to connect to your first EC2 instance.
 2. Fix a connection error!
- Still in your **EC2** console, select the checkbox next to **Instance - NextWork VPC 1**.
 - Select **Connect**.

Connect to instance [Info](#)

Connect to your instance i-0bd771d297992ead7 (Instance - NextWork VPC 1) using any of these options

[EC2 Instance Connect](#) [Session Manager](#) [SSH client](#) [EC2 serial console](#)

No public IPv4 or IPv6 address assigned
With no public IPv4 or IPv6 address, you can't use EC2 Instance Connect. Alternatively, you can try connecting using [EC2 Instance Connect Endpoint](#).

- Keeping **Disable** for the **Auto-assign IP address** option in our EC2 instance's network settings caused this error!
- Verify this by heading back to the **Instances** page in your EC2 console, and checking the Public IPv4 address field... it's empty

Instances (1/2) [Info](#)

Last updated 1 minute ago [C](#) [Connect](#) [Instance state](#) [Actions](#)

Find Instance by attribute or tag (case-sensitive) [All states](#)

Name	Instance ID	Instance state	Instance type
<input checked="" type="checkbox"/> Instance - NextWork VPC 1	i-0bd771d297992ead7	Running Q Q	t2.micro
<input type="checkbox"/> Instance - NextWork VPC 2	i-0b065d4262c66036a	Running Q Q	t2.micro

i-0bd771d297992ead7 (Instance - NextWork VPC 1)

Instance summary [Info](#)

Instance ID i-0bd771d297992ead7	Public IPv4 address	Private IPv4 addresses 10.1.7.1
--	---------------------	--

- On your EC2 console's left hand navigation panel, select **Elastic IPs**.

Network & Security

Security Groups

Elastic IPs

- Select **Allocate Elastic IP addresses**.
- Leave all default options.
- Select **Allocate**.
- Refresh your page, then select the new IP address you've set up.
- Select the **Actions** dropdown, then select **Associate Elastic IP address**.
- Under **Instance**, select **Instance - NextWork VPC 1**.

EC2 > Elastic IP addresses > Associate Elastic IP address

Associate Elastic IP address Info

Choose the instance or network interface to associate to this Elastic IP address (65.1.230.164)

Elastic IP address: 65.1.230.164

Resource type
Choose the type of resource with which to associate the Elastic IP address.

Instance
 Network interface

⚠️ If you associate an Elastic IP address with an instance that already has an Elastic IP address associated, the previously associated Elastic IP address will be disassociated, but the address will still be allocated to your account. [Learn more](#)

If no private IP address is specified, the Elastic IP address will be associated with the primary private IP address.

Instance

X @

aws EC2 Search [Alt+S] Instances Reassociations Asia Pacific (Mumbai) VivekVelturi

EC2 > Elastic IP addresses > Associate Elastic IP address

⚠️ but the address will still be allocated to your account. [Learn more](#)

If no private IP address is specified, the Elastic IP address will be associated with the primary private IP address.

Instance

X @

i-0bd771d297992ead7 (Instance - NextWork VPC 1) - running
i-0b065d4262c66036a (Instance - NextWork VPC 2) - running

Reassociation
Specify whether the Elastic IP address can be reassigned with a different resource if it's already associated with a resource.
 Allow this Elastic IP address to be reassigned

Cancel Associate

- Click **Associate**.
- Your EC2 Instance should have a public IP address now.
- To check this, select **Instances** from the left hand navigation panel.
- Select the checkbox next to **Instance - NextWork VPC 1**.
- You will see a **Public IPv4 address** for your EC2 instance now

Instances (1/2) Info Last updated 1 minute ago Connect Instance state Actions Launch instances

Find Instance by attribute or tag (case-sensitive) All states

Name	Instance ID	Instance state	Instance type	Status check
<input checked="" type="checkbox"/> Instance - NextWork VPC 1	i-0bd771d297992ead7	Running Q E	t2.micro	2/2 checks passed
<input type="checkbox"/> Instance - NextWork VPC 2	i-0b065d4262c66036a	Running Q E	t2.micro	2/2 checks passed

i-0bd771d297992ead7 (Instance - NextWork VPC 1)

Instance summary Info

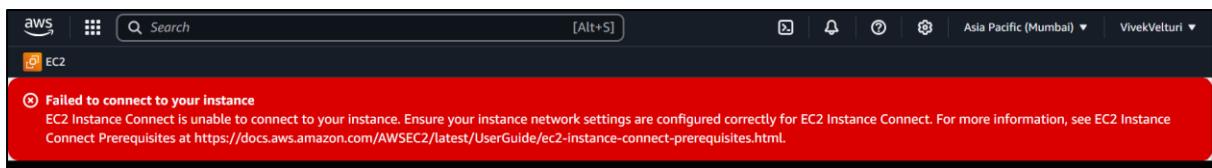
Instance ID i-0bd771d297992ead7	Public IPv4 address 65.1.230.164 open address	Private IPv4 addresses 10.1.7.1
---	---	---

Connect to Instance 1 (round two!)

IP address should be all resolved now... let's try connecting to your EC2 instance again

In this step, you're going to:

1. Use EC2 Instance Connect to connect to Instance 1 (one more time)!
 2. Fix (another) error.
- Select **Connect**.
 - In the EC2 Instance Connect set up page, select **Connect** again.



- We've failed to connect to our instance
- Head back to your **VPC console**.
- Select **Subnets** from the left hand navigation panel.
- Select the checkbox next to **NextWork-1-subnet-public1...**
- Investigate the **Route table** and **Network ACL** tabs

Route table: rtb-00f6e413ea8c75d72 / NextWork-1-rtb-public

Edit route table association

Destination	Target
10.1.0.0/16	local
10.2.0.0/16	pcx-036a963c1128ac95a
0.0.0.0/0	igw-04d6646d1af356e1b

subnet-06bfed490a4382234 / NextWork-1-subnet-public1-ap-south-1a

Details | Flow logs | Route table | **Network ACL** | CIDR reservations | Sharing | Tags

Network ACL: acl-0453c484534ad4f42

Inbound rules (2)

Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

Outbound rules (2)

Rule number	Type	Protocol	Port range	Destination	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

- leaves one more thing to investigate..., Copy the **VPC ID** of **NextWork-1-vpc**.



- Head into the **Security groups** page from the left hand navigation panel.
- There will be a bunch of nameless security groups!
- To find the VPC 1's default security group, let's use the handy search bar.
- Click into the search bar, and paste the ID you copied into the search bar. Make sure there aren't any empty spaces before your text.
- Select the filter!

Security Groups (12) Info			
Actions Export security groups to CSV Create security group			
<input type="text" value="vpc-01e6725856249bed0"/> X			
Group ID	Security group name	VPC ID	
6e5b682ab8b	launch-wizard-2	vpc-096d0213ee8f579	
5bbe3cbede2c	default	vpc-0343bfe016dcf245	
1df60a375f	default	vpc-08f51850f21517ff	

- Now that it narrowed it nicely for us. Select the checkbox next to VPC 1's **default** security group.
- Select the **Inbound rules** tab.

Inbound rules (1)					
Edit inbound rules Manage tags					
<input type="text" value="Search"/> X					
Name	Security group rule ID	IP version	Type	Protocol	
-	sgr-06d717eb836b3640c	-	All traffic	All	

- The answer lies in the **Source** of your security group's inbound rules. We're trying to access Instance - NextWork VPC 1 using SSH through EC2 Instance Connect, which is trying to connect to your instance over the internet. Your default security group only allows inbound traffic from within the VPC, so traffic from the internet is being cut off! **That's a key learning to take away:** The default security group for a new VPC does not allow incoming traffic from outside of the VPC. You have to allow inbound SSH traffic on port 22 yourself!

- In the **Inbound rules** tab, select **Edit inbound rules**.
- Select **Add rule**.
- For your new rule, configure the **Type** as **SSH**.
- Then, under **Source type**, select **Anywhere-IPv4**.

VPC > Security Groups > sg-05dc57dea66d07d87 - default > Edit inbound rules

Inbound rules [Info](#)

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
sgr-06d717eb836b3640c	All traffic ▼	All	All	Custom Anywhere-IPv4 <input checked="" type="checkbox"/> 7dea66d07d8 X	Delete
-	SSH ▼	TCP	22	An... ▲ 🔍 0.0.0.0/0 X	Delete

[Add rule](#)

- Select **Save rules**.

Inbound rules (2)					 Manage tags	Edit inbound rules
<input type="checkbox"/>		Name	Security group rule ID	IP version	Type	Protocol
<input type="checkbox"/>	-	sgr-06d717eb836b3640c	-		All traffic	All
<input type="checkbox"/>	-	sgr-0e63de432f5c5f8a2		IPv4	SSH	TCP

- With that modified, refresh your EC2 console's **Instances** page.
 - Select your **Instance-NextWork VPC 1** and select **Connect** again.
 - Select **Connect** in the EC2 Instance Connect setup page.

- Success.

Test VPC Peering

we've figured out how to connect with VPC 1's Instance!

Let's see if we can connect with VPC 2's instance from here.

In this step, you're going to:

1. Get Instance 1 to send test messages to Instance 2.
 2. Solve connection errors until Instance 2 is able to send messages back.
- Leave open the **EC2 Instance Connect** tab, but head back to your **EC2** console in a new tab.
 - Select **Instance - NextWork VPC 2**.
 - Copy Instance - NextWork VPC 2's **Private IPv4 address**.

Name	Instance ID	Instance state	Instance type	Status check
Instance - NextWork VPC 1	i-0bd771d297992ead7	Running	t2.micro	2/2 checks passed
Instance - NextWork VPC 2	i-0b065d4262c66036a	Running	t2.micro	2/2 checks passed

i-0b065d4262c66036a (Instance - NextWork VPC 2)

Instance summary

Instance ID i-0b065d4262c66036a	Public IPv4 address -	Private IPv4 addresses 10.2.4.196
------------------------------------	--------------------------	--------------------------------------

- Switch back to the **EC2 Instance Connect** tab.
- Run ping [the Private IPv4 address you just copied] in the terminal.
 - Eg: in my case: ping 10.2.4.196
- You should see a response similar to this

```
[ec2-user@ip-10-1-7-1 ~]$ ping 10.2.4.196
PING 10.2.4.196 (10.2.4.196) 56(84) bytes of data.
```

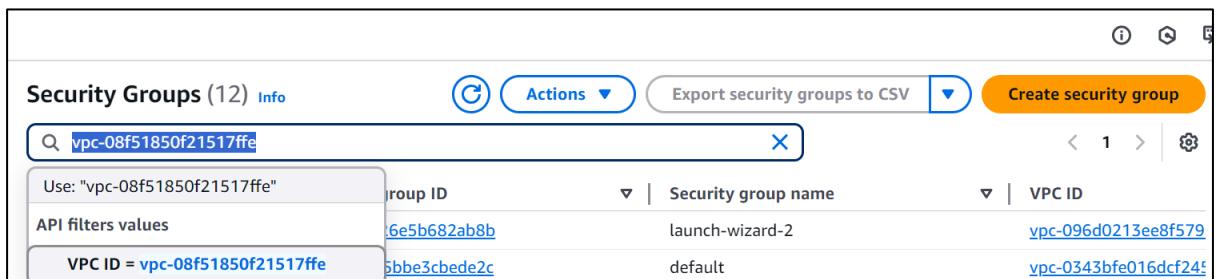
- This single line indicates that your Instance - NextWork VPC 1 has sent out a ping message... and that's about it.
Usually, when you ping another computer successfully, you should see **several** replies back instantly. Each reply tells you how long it took for the message to go to the Instance - NextWork VPC 2 and come back.
If you don't get any replies (that's our situation right now), or if the replies stop suddenly, it's usually a sign that there's a problem with the connection.
One common reason for these issues is that the target server (Instance - NextWork VPC 2) or its network might be blocking the type of messages used in ping, which are known as **ICMP (Internet Control Message Protocol) traffic**.

Blocking ICMP traffic is often done to prevent network attacks, like attackers can overwhelming a server with ping messages so it can't respond to real users wanting to use your application. Fair enough that ICMP traffic is blocked by default!

- To resolve this connectivity error, let's investigate whether **Instance - NextWork VPC 2** is allowing inbound ICMP traffic.
- Leave open the **EC2 Instance Connect** tab, but head back to your **VPC** console in a new tab.
- In the VPC console, select the **Subnets** page.
- Select VPC 2's subnet i.e. **NextWork-2-subnet-public1-...**



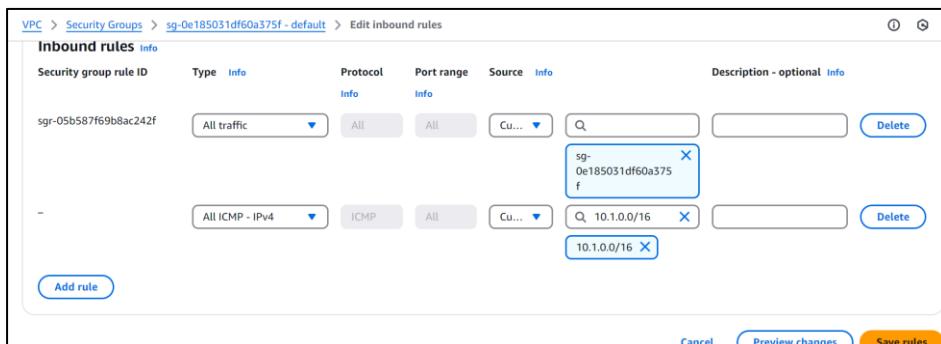
- Let's investigate the **Route tables** and **Network ACL** tabs for your public subnet
- The network ACL allows all types of inbound traffic from anywhere! So this looks perfectly fine.
- Before we finish, let's check the security groups!
- Copy the **VPC ID** of VPC 2.
- Select **Security groups** from the left hand navigation panel.
- Paste the **VPC ID** in the search bar, and select the suggested filter.



- Check your security group's **Inbound rules** tab - does this security group allow ICMP traffic from sources outside of VPC 2? (Nope!)

Let's fix this by letting inbound ICMP traffic from VPC 1.

- Select **Add new rule**.
- Change the **Type** to **All ICMP - IPv4**.
- Select **Edit inbound rule**
- Set the **Source** to traffic coming from VPC 1 - **10.1.0.0/16**



- Select **Save rules**.

Inbound rules (2)					Manage tags	Edit inbound rules
	Name	Security group rule ID	IP version	Type	Protocol	
<input type="checkbox"/>	-	sgr-05b587f69b8ac242f	-	All traffic	All	
<input type="checkbox"/>	-	sgr-089a55514892ffdad	IPv4	All ICMP - IPv4	ICMP	

- Revisit the **EC2 Instance Connect** tab that's connected to Instance - NextWork VPC 1.
- Lots of new lines coming through in the terminal.

The screenshot shows a terminal window within the AWS EC2 Instance Connect interface. The terminal output is as follows:

```

/ec2-user@ip-10-1-7-1 ~]$ ping 10.2.4.196
PING 10.2.4.196 (10.2.4.196) 56(84) bytes of data.
64 bytes from 10.2.4.196: icmp_seq=1572 ttl=127 time=0.464 ms
64 bytes from 10.2.4.196: icmp_seq=1573 ttl=127 time=0.971 ms
64 bytes from 10.2.4.196: icmp_seq=1574 ttl=127 time=0.481 ms
64 bytes from 10.2.4.196: icmp_seq=1575 ttl=127 time=0.825 ms
64 bytes from 10.2.4.196: icmp_seq=1576 ttl=127 time=0.764 ms
64 bytes from 10.2.4.196: icmp_seq=1577 ttl=127 time=1.10 ms
64 bytes from 10.2.4.196: icmp_seq=1578 ttl=127 time=0.717 ms
64 bytes from 10.2.4.196: icmp_seq=1579 ttl=127 time=0.388 ms
64 bytes from 10.2.4.196: icmp_seq=1580 ttl=127 time=1.05 ms
64 bytes from 10.2.4.196: icmp_seq=1581 ttl=127 time=0.540 ms
64 bytes from 10.2.4.196: icmp_seq=1582 ttl=127 time=0.273 ms
64 bytes from 10.2.4.196: icmp_seq=1583 ttl=127 time=1.09 ms
64 bytes from 10.2.4.196: icmp_seq=1584 ttl=127 time=0.824 ms
64 bytes from 10.2.4.196: icmp_seq=1585 ttl=127 time=0.779 ms
64 bytes from 10.2.4.196: icmp_seq=1586 ttl=127 time=0.450 ms

```

i-0bd771d297992ead7 (Instance - NextWork VPC 1)

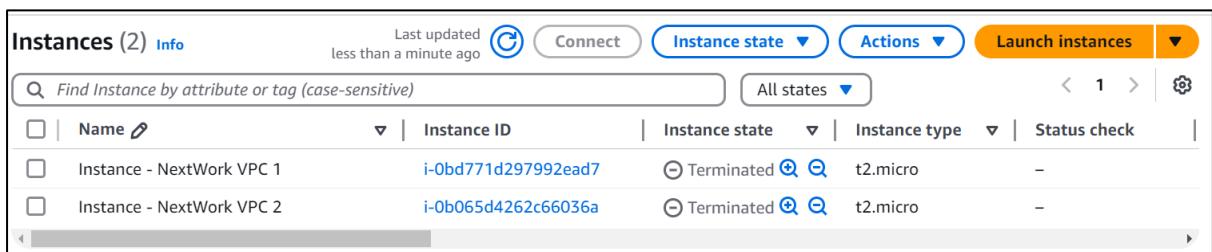
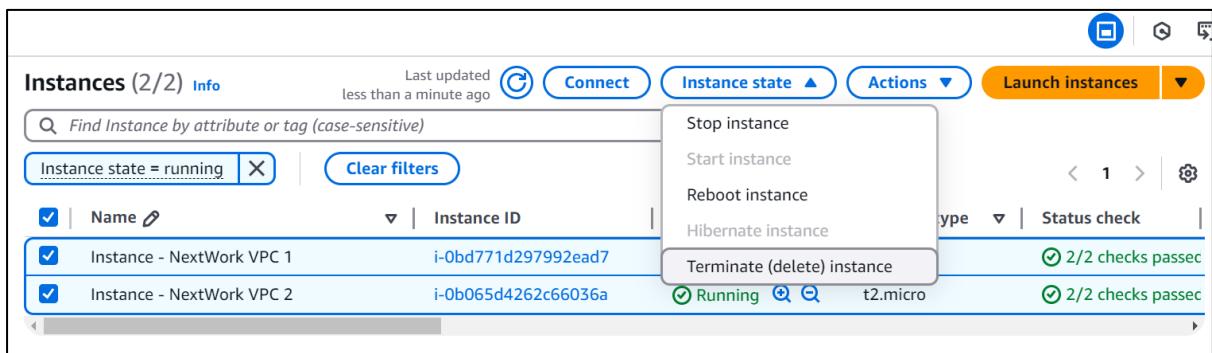
Public IPs: 65.1.230.164 Private IPs: 10.1.7.1

Congratulations! You've set up a peering architecture that connects VPC 1 to VPC 2 AND validated it with ping

Delete Your Resources

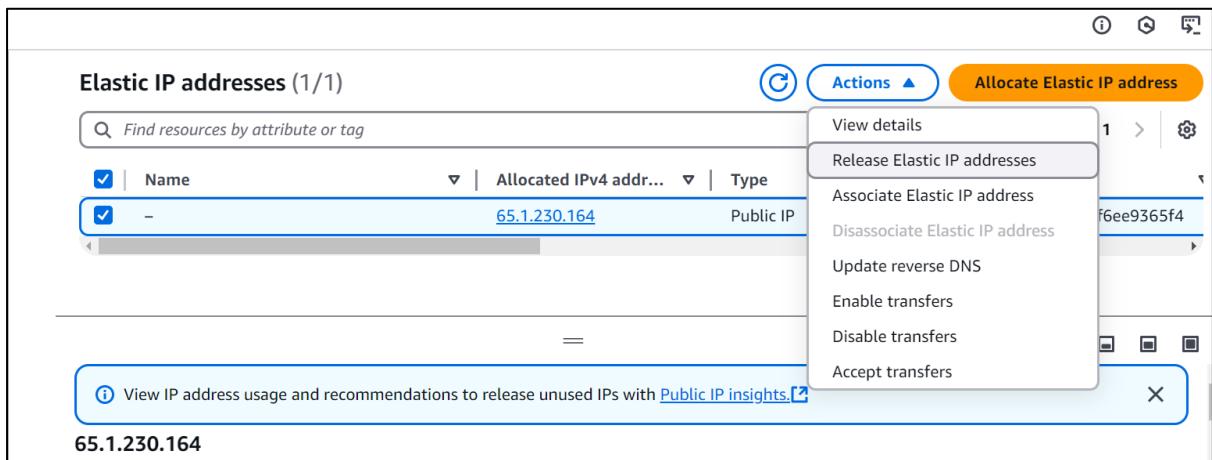
Delete your EC2 Instances

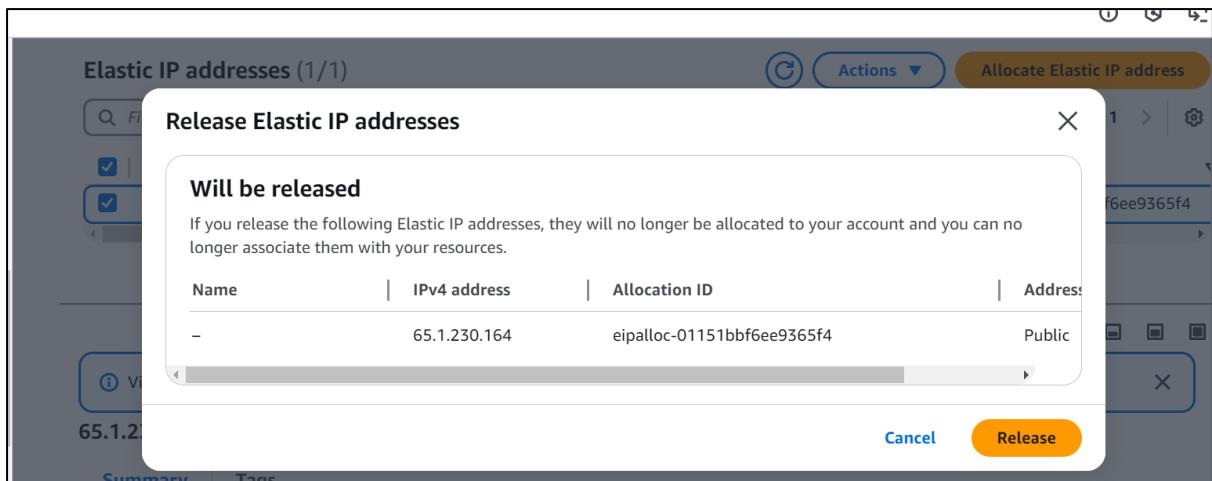
- Head back to the **Instances** page of your EC2 console.
- Select the checkboxes next to **Instance - NextWork VPC 1** and **Instance - NextWork VPC 2**.
- Select **Instance state**, then select **Terminate Instance**.
- Select **Terminate**.



Delete your Elastic IP address

- Select **Elastic IPs** from the left hand navigation panel.
- Select the IP address you've created.
- Select Actions, then **Release Elastic IP addresses**.

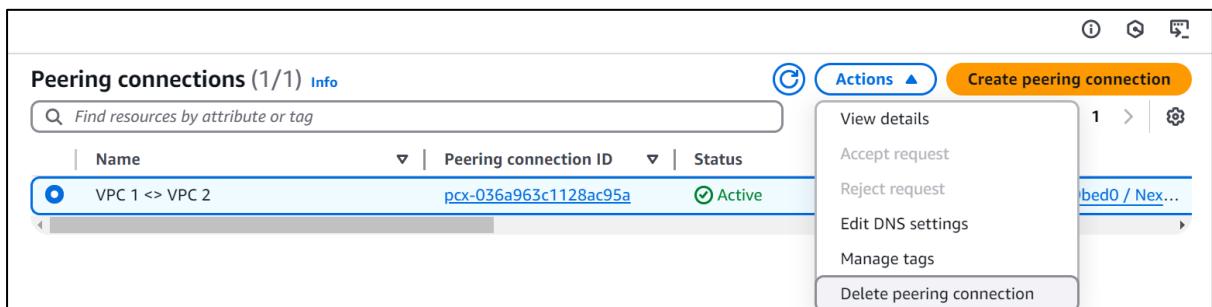




- Select **Release**.

Delete VPC Peering Connections

- Head back to your **VPC** console.
- Select **Peering connections** from your left hand navigation panel.
- Select the VPC 1 <> VPC 2 peering connection.
- Select **Actions**, then **Delete peering connection**.



- Select the checkbox to **Delete related route table entries**.
- Type **delete** in the text box and click **Delete**.

Routes targeting this peering connection

Route table ID	Destination
rtb-04ee724bbcf3ee793	10.1.0.0/16
rtb-00f6e413ea8c75d72	10.2.0.0/16

Route table entries
 Select whether or not to delete the entries that target this peering connection from the route tables.

Delete related route table entries
 Do not delete route table entries

To confirm deletion, type **delete** in the field:

delete

[Cancel](#) [Delete](#)

Delete your VPCs

- Select **Your VPCs** from your left hand navigation panel.
- Select **NextWork-1-vpc**, then **Actions**, and **Delete VPC**.
- Type **delete** in the text box and click **Delete**.

This VPC will be deleted permanently and cannot be recovered later:

Name	VPC ID	State
NextWork-1-vpc	vpc-01e6725856249bed0	Available

Will also be deleted
 The following 3 resources will also be deleted permanently and cannot be recovered later:

Name	Resource ID	State
NextWork-1-igw	igw-04d6646d1af356e1b	Available
NextWork-1-rtb-public	rtb-00f6e413ea8c75d72	-
NextWork-1-subnet-public1-ap-south-1a	subnet-06bfed490a4382234	Available

To confirm deletion, type **delete** in the field:

delete

[Cancel](#) [Delete](#)

- Note: if you get stopped from deleting your VPC because **network interfaces** are still attached to your VPC - delete all the attached network interfaces first!
- Select **NextWork-2-vpc**, then **Actions**, and **Delete VPC**.
- Type **delete** in the text box and click **Delete**.

Other network components should be automatically deleted with your VPC, but it's always a good idea to check anyway:

1. Subnets 2.Route tables 3. Internet gateways 4. Network ACLs 5.Security groups

Don't forget to **refresh** each page before checking if the resources are still in your account!