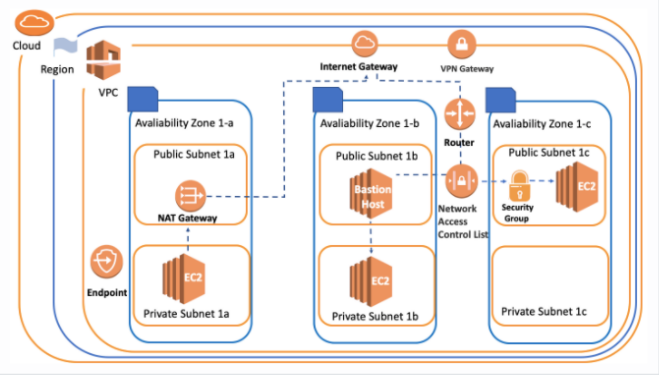
Creating VPC

VPC Scenario



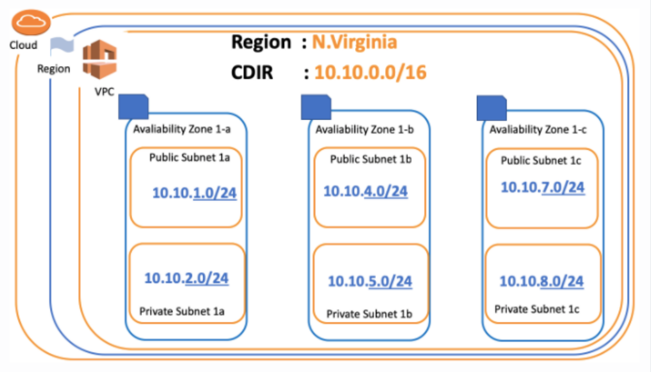
VPC Scenario

In our VPC, as you see in the picture above, we will have 6 subnets totally. 3 of them are Public Subnets that can be accessed from the Internet and the others are private subnets, that are not accessible from the internet.

So we will add 1 public and one private subnet to each of the 3 AZs located in the N. Virginia Region.

In the following sections, we'll add VPC's various components and see the functionality of them step by step.

VPC CIDR



VPC CIDR

First of all, we will determine our VPCs CIDR.

We'll use 10.10.0.0/16 CIDR. This will assign us more than 65,000 IP addresses that we can use under this VPC.

We need to determine this well according to our needs from the beginning. Because AWS does not allow you to expand the IP range you assign to VPC. You can only add additional second blocks.

In our VPC, we have 3 Public Subnets and 3 Private Subnets. For each subnet, we have determined networks that will be under the network of 10.10.0.0/16.

Public Subnets:

- 10.10.1.0/24
- 10.10.4.0/24
- 10.10.7.0/24

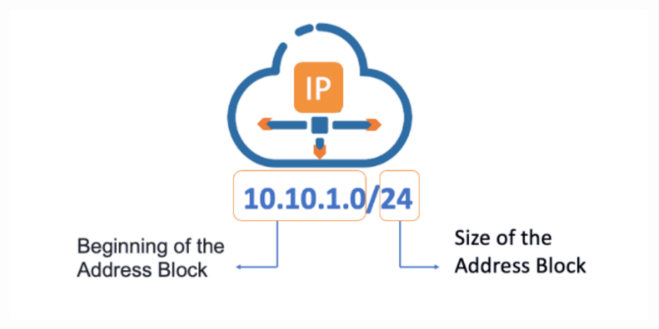
Private Subnets:

- 10.10.2.0/24
- 10.10.5.0/24
- 10.10.8.0/24 in our private subnets.

💡Tips:

- 10.10.3.0/24 and 10.10.6.0/24 are reserved as forward looking need.

IP Definition



IP Definition

The first part of 10.10.1.0/24 (**10.10.1.0**) on left: indicates where the IP address block we want to use in our own virtual network started.

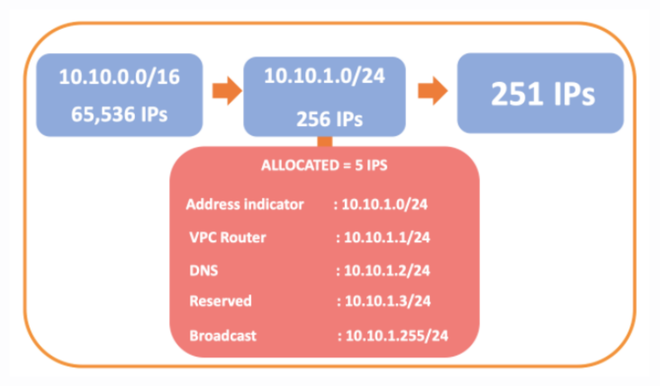
The second part of 10.10.1.0/24 (/ **24**)on right: this indicates the size of the address block.

As the **Size Block** (/ number) decreases, the number of IP located in CDIR Block increases.

For example, while there are 256 IPs in 10.10.1.0/**24** IP block, there is only 1 IP in 10.10.1.0/**32** IP block.

> You can get further information and calculation about IP block from this link mxtoolbox.com

Unavailable IP Addresses For Use



ALLOCATED = 5 IPS

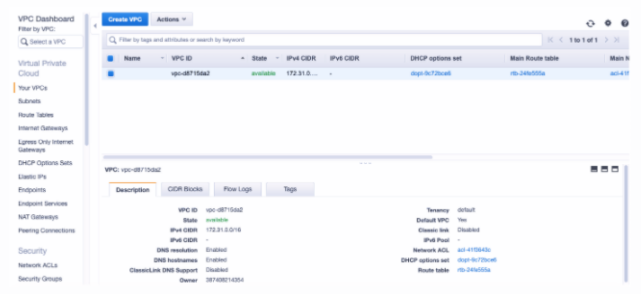
Address indicator	: 10.10.1.0/24
VPC Router	: 10.10.1.1/24
DNS	: 10.10.1.2/24
Reserved	: 10.10.1.3/24
Broadcast	: 10.10.1.255/24

Unavailable IP Addresses For Use

As you see in the picture above, there are 5 IP addresses that we cannot use from this block and therefore we can use a total of 251 IP addresses.

So it means we can create 251 different devices in this subnet.

Creating VPC



Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Main Route table	Main N
	vpc-d8715da2	available	172.31.0....	-	dopt-9c72bce6	rtb-24fe555a	acl-41f

VPC: vpc-d8715da2

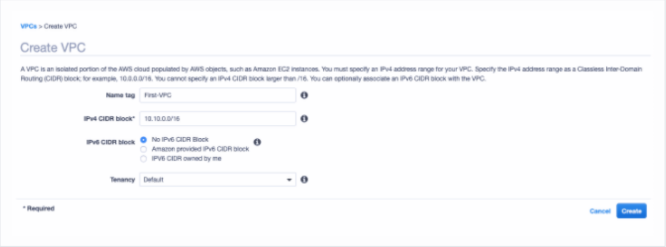
VPC ID	vpc-d8715da2	Tenancy	default
State	available	Default VPC	Yes
IPv4 CIDR	172.31.0.0/16	Classic link	Disabled
IPv6 CIDR	-	IPv6 Pool	-
DNS resolution	Enabled	Network ACL	acl-41f3643c
DNS hostnames	Enabled	DHCP options set	dopt-9c72bce6
ClassicLink DNS Support	Disabled	Route table	rtb-24fe555a
Owner	387408214354		

Your VPC

Now let's connect to the console and start the process of creating VPC step by step.

If we press the **Your VPC** tab on the left-hand menu, we 'll find our current VPCs as you see in the picture above.

Then click **Create VPC** tab to start and you'll see the page seen below.



A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances. You must specify an IPv4 address range for your VPC. Specify the IPv4 address range as a Classless Inter-Domain Routing (CIDR) block; for example, 10.0.0.0/16. You cannot specify an IPv4 CIDR block larger than /16. You can optionally associate an IPv6 CIDR block with the VPC.

Name tag: First-VPC

IPv4 CIDR block*: 10.10.0.0/16

IPv6 CIDR block: ● No IPv6 CIDR Block ○ Amazon provided IPv6 CIDR block ○ IPV6 CIDR owned by me

Tenancy: Default

Create VPC Page

Name Tag:

You can write any name for your VPC. Let's say First-VPC

IPv4 CIDR Block:

Now, we need to determine the CIDR block. Let's write the 10.10.0.0/16 block. It allows more than 65000 IP addresses.

IPv6 CIDR Block:

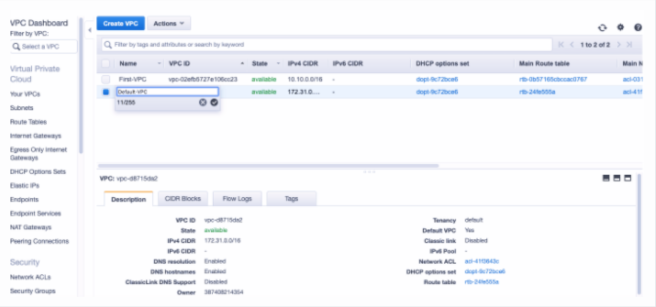
AWS asks if we want to use IPv6 block as an option, let's not use it for now and click **No IPv6 CIDR Block**.

Tenancy:

In the tenancy section, there are 2 options: Default and Dedicated, If we do not have an extra security policy, we can use it in a shared environment by choosing the default.

If we choose dedicated, a VPC is created on special devices. Of course, in this case, an additional fee policy is applied. Let's continue leaving Default.

Let's create our first VPC by clicking **Create**. Congratulations!!!!!



Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Main Route table	Main N
First-VPC	vpc-02efb6727e106cc23	available	10.10.0.0/16	-	dopt-9c72bce6	rtb-0b57165cbccac0767	acl-031
Default-VPC 11/255		available	172.31.0....	-	dopt-9c72bce6	rtb-24fe555a	acl-41f

VPC: vpc-d8715da2

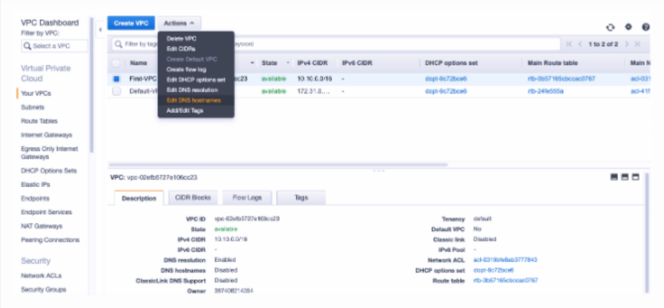
VPC ID: vpc-d8715da2
State: available
IPv4 CIDR: 172.31.0.0/16
IPv6 CIDR: -
DNS resolution: Enabled
DNS hostnames: Enabled
ClassicLink DNS Support: Disabled
Owner: 387408214354
Tenancy: default
Default VPC: Yes
Classic link: Disabled
IPv6 Pool: -
Network ACL: acl-41f3643c
DHCP options set: dopt-9c72bce6
Route table: rtb-24fe555a

VPC

When we look at our VPCs, we see that there are 2 VPC. Let's name our default VPC as Default so that we don't confuse it when it appears.

As we can see, we can create VPC so easily in one step. But of course, It's not finish yet. We also need to create all the components under VPC one by one and this is the complicated part of the work.

Enabling DNS Hostname



Name		State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Main Route table	Main N
First-VPC	c23	available	10.10.0.0/16	-	dopt-9c72bce6	rtb-0b57165cbccac0767	acl-031
Default-V		available	172.31.0....	-	dopt-9c72bce6	rtb-24fe555a	acl-41f

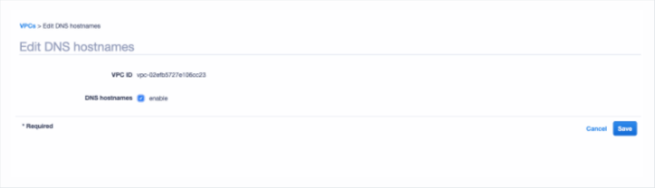
VPC: vpc-02efb6727e106cc23

VPC ID: vpc-02efb5727e106cc23
State: available
IPv4 CIDR: 10.10.0.0/16
IPv6 CIDR: -
DNS resolution: Enabled
DNS hostnames: Disabled
ClassicLink DNS Support: Disabled
Owner: 387408214354
Tenancy: default
Default VPC: No
Classic link: Disabled
IPv6 Pool: -
Network ACL: acl-0319bfe8ab3777843
DHCP options set: dopt-9c72bce6
Route table: rtb-0b57165cbccac0767

VPC Action Menu

The first thing we need to do after creating a new VPC is to activate DNS Hostname.

After choosing our VPC, we select the Edit DNS hostnames option from the Action Menu.



VPC ID: vpc-02efb5727e106cc23

DNS hostnames: ☑ enable

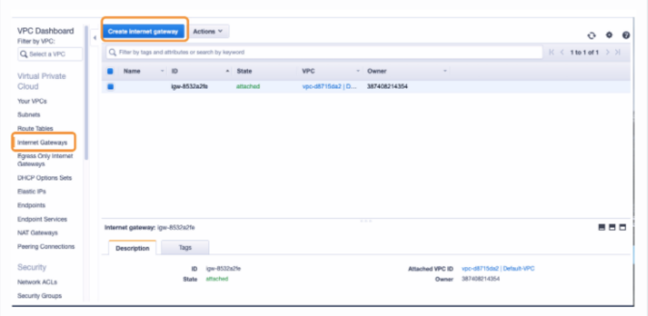
Enabling DNS Hostname

Here we select the relevant box to enable DNS hostname as you see in the picture above.

If we do not activate it, resources created or assigned under this VPC will not have a DNS hostname. Thus, these machines will not be able to communicate over these DNS names.

This is why it is very important to activate the DNS Host Name when a new VPC is created.

Creating Internet Gateway



Name	ID	State	VPC	Owner
	igw-8532a2fe	attached	vpc-d8715da2 | D...	387408214354

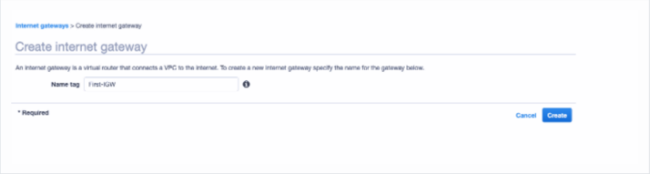
Internet gateway: igw-8532a2fe

ID: igw-8532a2fe
State: attached
Attached VPC ID: vpc-d8715da2 | Default-VPC
Owner: 387408214354

Internet Gateway

We have completed creating a new VPC. It's time to connect this virtual data center to the internet through the Internet Gateway. So you can depict it as ADSL or Fiber line drawing like your home.

- Let's click the **Internet Gateway** tab from left-hand menu as you see in the picture above.
- Then click **Create Internet Gateway.**



An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

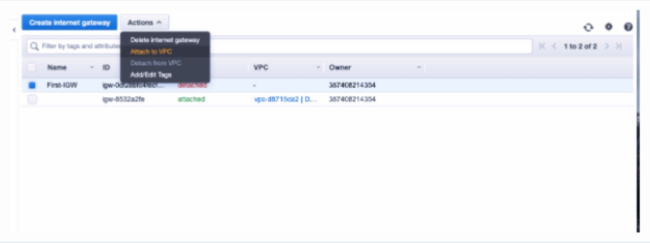
Name tag: First-IGW

Creating Internet Gateway

- Let's say First-IGW as a name tag.
- Then complete the first Internet Gateway creation process by clicking **Create**.

It has been created now, but as you can see on the dashboard that Internet Gateway is detached. It is not connected to any VPCs.

Internet Gateway Association

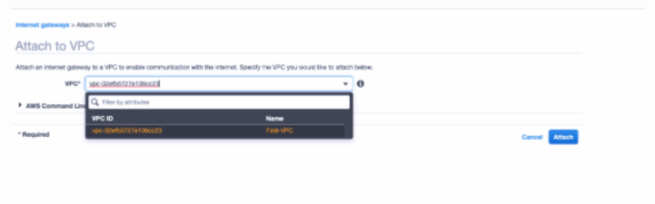


Name	ID		VPC	Owner
First-IGW	igw-0d7a8bf04ecf...	detached	-	387408214354
	igw-8532a2fe	attached	vpc-d8715da2 | D...	387408214354

Internet Gateway Action Menu

Let's attach this Internet Gateway to our VPC and provide the association between our VPC and Intenet Gateway

- After selecting our new Internet Gateway, let's click **Attach** to VPC option from the **Action Menu** as you sen in the picture above.



Internet gateways > Attach to VPC

Attach to VPC

Attach an internet gateway to a VPC to enable communication with the internet. Specify the VPC you would like to attach below.

VPC* vpc-02efb5727e106cc23

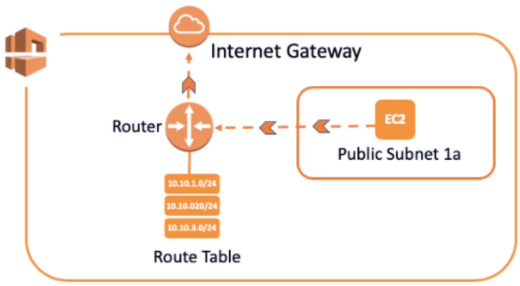
VPC ID	Name
vpc-02efb5727e106cc23	First-VPC

Attaching VPC

- Let's select **First-VPC** from the opened droplet and click **Attach**. It's done.

Thus, we created our internet connection, namely the Internet Gateway, which provides access to the resources in the VPC, as well as the access to the external world of the VPC.

VPC and Route Tables



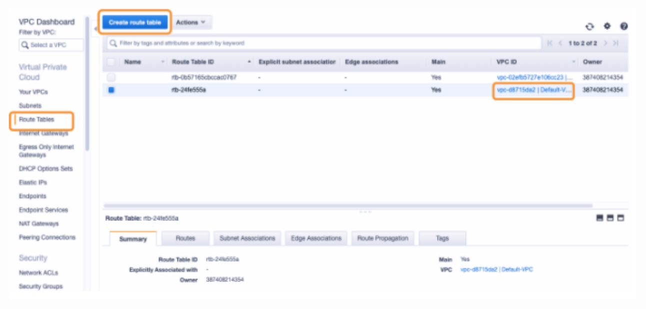
Route Table and Router

Now, we need to ensure that VPC delivers packages to the outside of the VPC via the Internet Gateway, First-IGW.

So we need to edit the packet forwarding rules at this stage. As you can imagine, we will do this via Route Tables.

So let's examine the Route Table.

Route Tables-1



	Name	Route Table ID	Explicit subnet associatior	Edge associations	Main	VPC ID	Owner
□		rtb-0b57165cbccac0767	-	-	Yes	vpc-02efb5727e106cc23 |...	387408214354
■		rtb-24fe555a	-	-	Yes	vpc-d8715da2 | Default-V...	387408214354

Route Table: rtb-24fe555a

Route Table ID rtb-24fe555a
Explicitly Associated with -
Owner 387408214354
Main Yes
VPC vpc-d8715da2 | Default-VPC

Route Tables

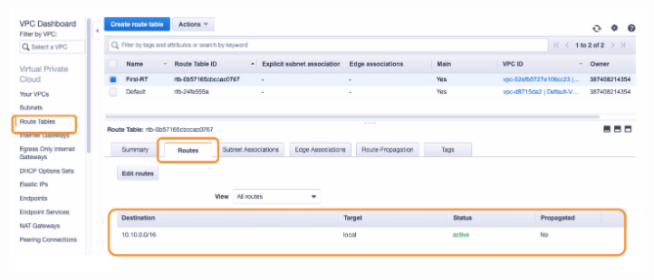
Now, let's go to the Route Tables from the menu bar on the left-hand side of the VPC section in the AWS Console.

When we look at this part, we'll see 2 Route tables as shown in the picture above.

- One of them is the Route Table of default VPC,
- The other is the Route Table of First-VPC that we have just created.

First of all, let's name our new route table as **First-RT** and the default one as **Default** from the **Name** tab just to the right of the VPC boxes.

Route Tables-2



	Name	Route Table ID	Explicit subnet associatior	Edge associations	Main	VPC ID	Owner
■	First-RT	rtb-0b57165cbccac0767	-	-	Yes	vpc-02efb5727e106cc23 |...	387408214354
□	Default	rtb-24fe555a	-	-	Yes	vpc-d8715da2 | Default-V...	387408214354

Route Table: rtb-0b57165cbccac0767

View All routes

Destination	Target	Status	Propagated
10.10.0.0/16	local	active	No

Route Table

While the First-RT is selected, let's click on the **Routes** tab from the below as you see in the picture above.

There is one Default Route here. You'll see two values important: **Destination** and **Target**

Destination:

Destination means, where you want your package to be delivered. It may be your own VPCs CDIR block like 10.10.0.0/16 another VPCs CDIR block or 0.0.0.0/0 CDIR block that represents anywhere outside the VPC.

Target:

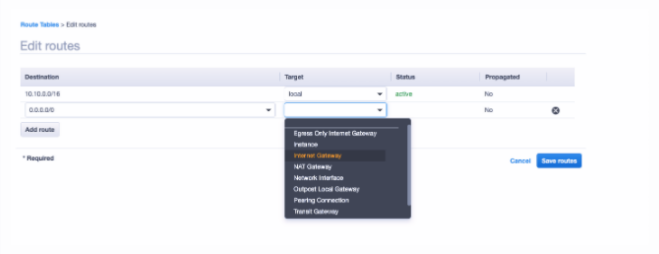
Here AWS asks you, by which component (Target) you'll deliver the package to the determined destination. Target refers to the Internet Gateway, NAT Gateway, Peering Connection, etc. here.

In our First-RT, **Destination** is **10.10.0.0/16** in other words, our current VPC and **Target** is **Local**.

It means to deliver all of the packages on yourself. You can't send packages to the outside.

Route Table-3

But, we want to send our packages to the outside, what do we need to do? Let's add a rule to tackle this issue.



Route Tables > Edit routes

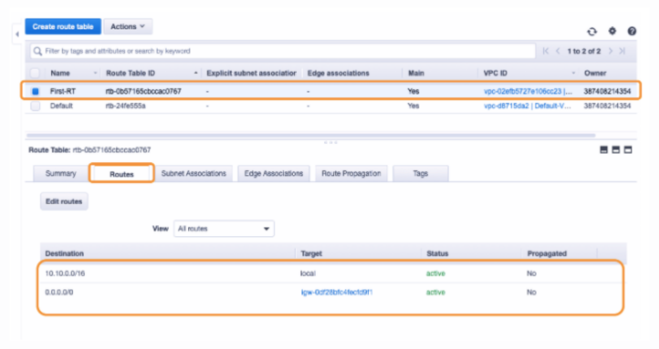
Edit routes

Destination	Target	Status	Propagated
10.10.0.0/16	local	active	No
0.0.0.0/0			No

Add Route

- After clicking the **Add Route** tab in it, we write **0.0.0.0/0** as Destination. It means anywhere outside the VPC.
- As the Target, we select the **Internet Gateway** that we previously created as First-IGW from the drop-down menu.
- Let's save by clicking **Save Routes**.

Route Table-4



	Name	Route Table ID	Explicit subnet associatior	Edge associations	Main	VPC ID	Owner
■	First-RT	rtb-0b57165cbccac0767	-	-	Yes	vpc-02efb5727e106cc23 |...	387408214354
□	Default	rtb-24fe555a	-	-	Yes	vpc-d8715da2 | Default-V...	387408214354

Route Table: rtb-0b57165cbccac0767

View All routes

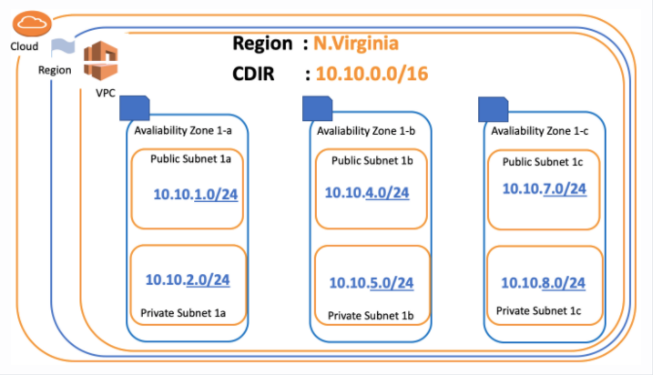
Destination	Target	Status	Propagated
10.10.0.0/16	local	active	No
0.0.0.0/0	igw-0df28bfc4fecfd9f1	active	No

Route Table

So what have we done so far?

- There are now 2 routes under our Route Table:
  - The first route says; If the package is addressed to VPC's IP Block (10.10.0.0/16), don't send this package to the outside of this VPC.
  - The second route says, If the package is addressed anywhere except IP Block of VPC (10.10.0.0/16), send this package to Internet Gateway to be delivered to the outside of the VPC

Creating Subnets-1

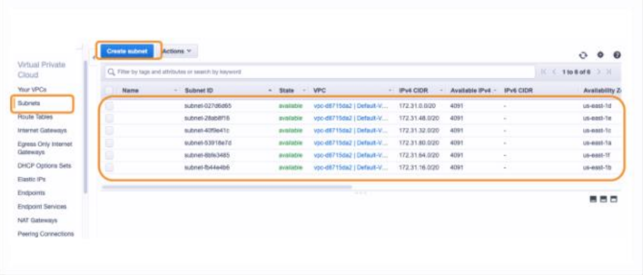


Subnet Diagram

Now we will create all the subnets we mentioned in our scenario. But, first of all, let's remember the diagram.

We'll create 6 subnets in 3 Availability Zones (AZ) totally. So, in each AZ there will be 2 subnets. One will be Public and the other one will be Private.

Let's click the **Subnets** tab from the VPC menu on the left as you see in the picture below.

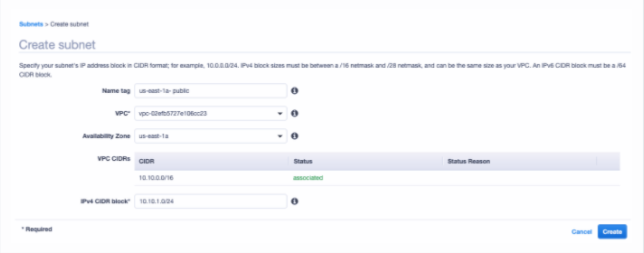


Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Z
	subnet-027d6d65	available	vpc-d8715da2 | Default-V...	172.31.0.0/20	4091	-	us-east-1d
	subnet-28ab8f16	available	vpc-d8715da2 | Default-V...	172.31.48.0/20	4091	-	us-east-1e
	subnet-40f9e41c	available	vpc-d8715da2 | Default-V...	172.31.32.0/20	4091	-	us-east-1c
	subnet-53918e7d	available	vpc-d8715da2 | Default-V...	172.31.80.0/20	4091	-	us-east-1a
	subnet-8bfe3485	available	vpc-d8715da2 | Default-V...	172.31.64.0/20	4091	-	us-east-1f
	subnet-fb44e4b6	available	vpc-d8715da2 | Default-V...	172.31.16.0/20	4091	-	us-east-1b

Subnets

There are 6 default subnets for each AZ in N.Virginia Region. These are subnets of the *Default VPC*. Now we will create our own subnets by clicking **Create Subnet**.

Creating Subnets-2



Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask, and can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag: us-east-1a- public

VPC*: vpc-02efb5727e106cc23

Availability Zone: us-east-1a

VPC CIDRs:

CIDR	Status	Status Reason
10.10.0.0/16	associated	

IPv4 CIDR block*: 10.10.1.0/24

* Required

Creating Subnets

Name Tag:

Let's name our first subnet as **us-east-1a- public**.

VPC:

We choose the newly created **First-VPC**

Availability Zone:

We named the subnet as **us-east-1a- public**. So we select **us-east-1a**. The others will be named as the same methodology.

IPv4 CIDR Block:

Here we write **10.10.1.0/24**. Then click **Create**.

Our subnet is ready. Let's do the same process for the 5 remaining subnets according to the following information, depending on the scenario.

AZ: us-east-1a

Name: us-east-1a-public CIDR: 10.10.1.0/24

Name: us-east-1a-private CIDR:10.10.2.0/24

AZ: us-east-1b

Name: us-east-1b-public CIDR:10.10.4.0/24

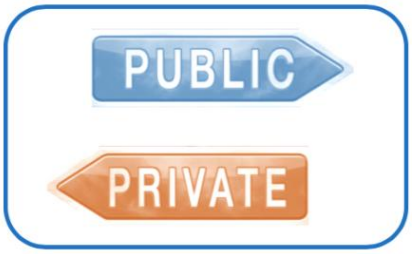
Name: us-east-1b-private CIDR:10.10.5.0/24

AZ: us-east-1c

Name: us-east-1c-public CIDR:10.10.7.0/24

Name: us-east-1c-private CIDR:10.10.8.0/24

Public or Private Subnet-1

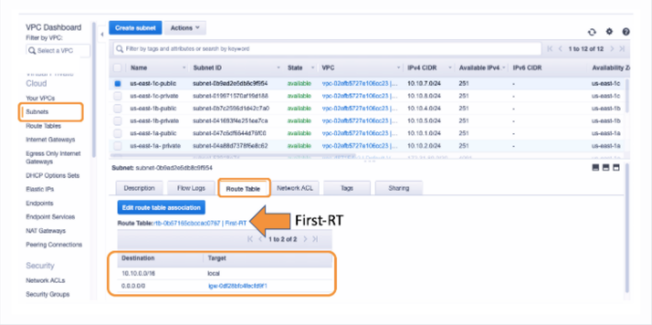


We had 6 subnets in total. We created one public and one private subnet in each AZ.

Currently, these 6 subnets are exactly the same. There is no difference in property between them. We created them all in a similar way. Only their names and CDIRs are different.

So what determines to be public and private? Of course, the answer is **Route Tables**. Let's see how.

Public or Private Subnet-2



Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Z
us-east-1c-public	subnet-0b9ad2e5db8c9f954	available	vpc-02efb5727e106cc23 |...	10.10.7.0/24	251	-	us-east-1c
us-east-1c-private	subnet-019971570af19d188	available	vpc-02efb5727e106cc23 |...	10.10.8.0/24	251	-	us-east-1c
us-east-1b-public	subnet-0b7c2596d1d42c7a0	available	vpc-02efb5727e106cc23 |...	10.10.4.0/24	251	-	us-east-1b
us-east-1b-private	subnet-041693f4e251ea7ca	available	vpc-02efb5727e106cc23 |...	10.10.5.0/24	251	-	us-east-1b
us-east-1a-public	subnet-047c6df6644d76f00	available	vpc-02efb5727e106cc23 |...	10.10.1.0/24	251	-	us-east-1a
us-east-1a- private	subnet-04a88d7378f6e8c62	available	vpc-02efb5727e106cc23 |...	10.10.2.0/24	251	-	us-east-1a

Subnet: subnet-0b9ad2e5db8c9f954

Route Table: rtb-0b57166cbccac0767 | First-RT

Destination	Target
10.10.0.0/16	local
0.0.0.0/0	igw-0df28bfc4fecfd9f1

Route Table of Subnets

Let's click the **Subnets** section from the left-hand menu then click one of our newly created Subnet(us-east-1c-public) and select the **Route Table** tab from the bottom as you see in the picture above.

Here we see that our subnet is assigned to the newly created route table First-RT. When we create subnets, these subnets **automatically associate with that VPC's Route Table**.

Here we have 2 routes in First-RT. As we explain in the last lesson, It means; If the package is related to 10.10.0.0/16 CDIR then consider it as local. If it's related to 0.0.0.0/0, in other words anywhere except local, send it to the Internet Gateway. **So all our subnets have internet connectivity**.

But we do not want our 3 private subnets to be accessible from the internet. This is the necessity to be private.

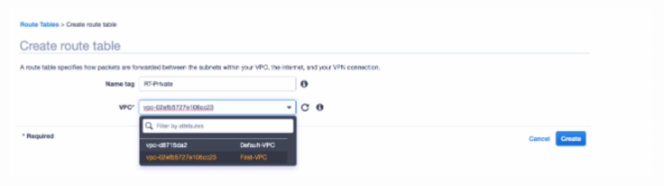
So, we need to create **Private Route Table** hasn't internet connectivity then we'll associate 3 private subnets to Private Route Table.

In other words, we determine the private/public status with the **Route Tables** we create.

Creating Route Table for Private Subnets

Let's create a new route table and connect these subnets to the Route Table and determine the rules accordingly.

- Click **Route Table** tab from the left-hand menu then select the **Create Route Table** tab and you'll see the page shown below.



Route Tables > Create route table

Create route table

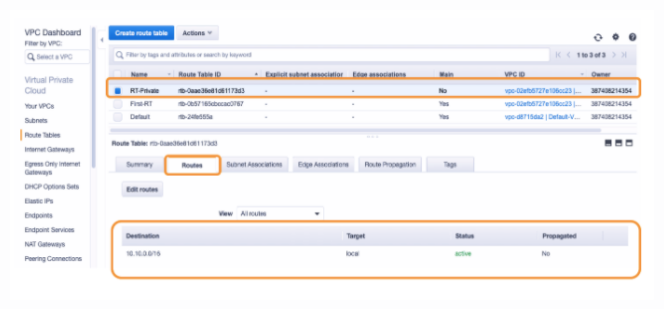
A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Name tag: RT-Private

VPC*: vpc-02efb5727e106cc23

Create New Route Table

- Let's say **RT-Private** as a name.
- Choose our **First-VPC** as VPC and then Click **Create**.



Name	Route Table ID	Explicit subnet associatior	Edge associations	Main	VPC ID	Owner
RT-Private	rtb-0aae36e81d61173d3	-	-	No	vpc-02efb5727e106cc23 |...	387408214354
First-RT	rtb-0b57165cbccac0767	-	-	Yes	vpc-02efb5727e106cc23 |...	387408214354
Default	rtb-24fe555a	-	-	Yes	vpc-d8715da2 | Default-V...	387408214354

Route Table: rtb-0aae36e81d61173d3

View: All routes

Destination	Target	Status	Propagated
10.10.0.0/16	local	active	No

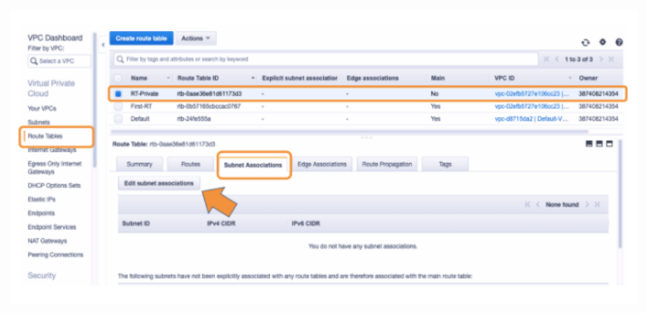
Create New Route Table

When we look at the **Routes** tab of this new route table as you see in the picture above. We'll see only 10.10.0.0/16 Destination for Local Target.

So this route table does not have a route to deliver packages to the outside of the VPC world, as we mention in the last lesson.

Association of Private Subnets to Private Route Table

- First let's click on the **Edit Subnet Associations** tab from the Subnet Association section with the newly created Private Route Table selected as you see in the picture below.



Name	Route Table ID	Explicit subnet associatior	Edge associations	Main	VPC ID	Owner
RT-Private	rtb-0aae36e81d61173d3	-	-	No	vpc-02efb5727e106cc23 |...	387408214354
First-RT	rtb-0b57165cbccac0767	-	-	Yes	vpc-02efb5727e106cc23 |...	387408214354
Default	rtb-24fe555a	-	-	Yes	vpc-d8715da2 | Default-V...	387408214354

Route Table: rtb-0aae36e81d61173d3

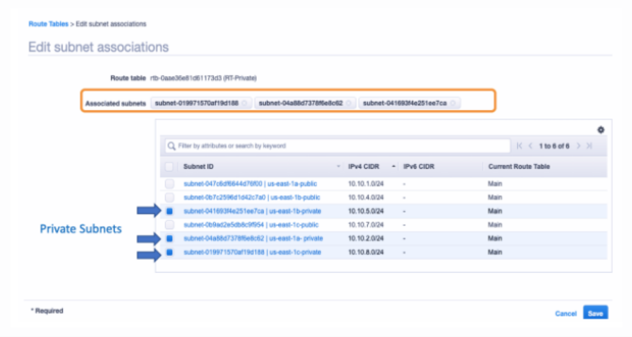
Subnet ID	IPv4 CIDR	IPv6 CIDR

You do not have any subnet associations.

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Edit Subnet Association

- Then choose all **Private Subnets** and associate them by clicking **Save**.



Route Tables > Edit subnet associations

Edit subnet associations

Route table: rtb-0aae36e81d61173d3 (RT-Private)

Associated subnets: subnet-019971570af19d188 subnet-04a88d7378f6e8c62 subnet-041693f4e251ee7ca

	Subnet ID	IPv4 CIDR	IPv6 CIDR	Current Route Table
□	subnet-047c6df6644d76f00 | us-east-1a-public	10.10.1.0/24	-	Main
□	subnet-0b7c2596d1d42c7a0 | us-east-1b-public	10.10.4.0/24	-	Main
☑	subnet-041693f4e251ee7ca | us-east-1b-private	10.10.5.0/24	-	Main
□	subnet-0b9ad2e5db8c9f954 | us-east-1c-public	10.10.7.0/24	-	Main
☑	subnet-04a88d7378f6e8c62 | us-east-1a- private	10.10.2.0/24	-	Main
☑	subnet-019971570af19d188 | us-east-1c-private	10.10.8.0/24	-	Main

Edit Subnets

Thus, we have assigned 3 subnets that we think of as private to the Route Table.

So this route table does not have a route to deliver packages to any outside world.

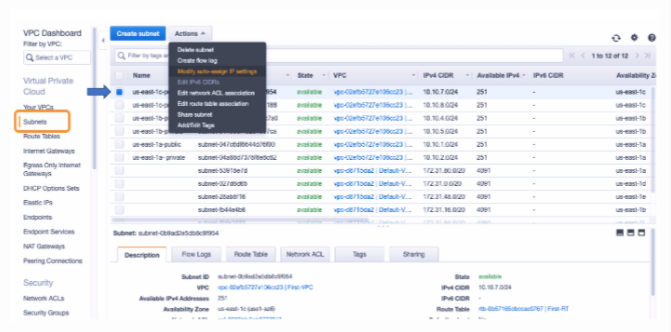
Therefore, subnets connected to this route table will not be able to communicate with the outside world.

As seen here, we make the Private / Public distinction through Route Tables.

Modify Auto-Assign IP Settings for Public Subnets

As for Public Subnets, we want the virtual machines that we will create for Public subnets to be associated with the outside world.

For this, the machines that we will put on these subnets must have Public IP addresses. But, it comes off by default. Let's set them now.



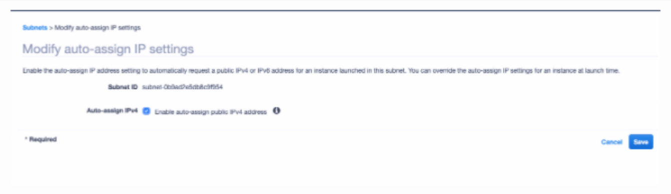
Name		State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Z
us-east-1c-p	954	available	vpc-02efb5727e106cc23 |...	10.10.7.0/24	251	-	us-east-1c
us-east-1c-p	188	available	vpc-02efb5727e106cc23 |...	10.10.8.0/24	251	-	us-east-1c
us-east-1b-p	7a0	available	vpc-02efb5727e106cc23 |...	10.10.4.0/24	251	-	us-east-1b
us-east-1b-p	7ca	available	vpc-02efb5727e106cc23 |...	10.10.5.0/24	251	-	us-east-1b
us-east-1a-public	subnet-047c6df6644d76f00	available	vpc-02efb5727e106cc23 |...	10.10.1.0/24	251	-	us-east-1a
us-east-1a- private	subnet-04a88d7378f6e8c62	available	vpc-02efb5727e106cc23 |...	10.10.2.0/24	251	-	us-east-1a
	subnet-53918e7d	available	vpc-d8715da2 | Default-V...	172.31.80.0/20	4091	-	us-east-1a
	subnet-027d6d65	available	vpc-d8715da2 | Default-V...	172.31.0.0/20	4091	-	us-east-1d
	subnet-28ab8f16	available	vpc-d8715da2 | Default-V...	172.31.48.0/20	4091	-	us-east-1e
	subnet-fb44e4b6	available	vpc-d8715da2 | Default-V...	172.31.16.0/20	4091	-	us-east-1b

Subnet: subnet-0b9ad2e5db8c9f954

Subnet ID: subnet-0b9ad2e5db8c9f954
VPC: vpc-02efb5727e106cc23 | First-VPC
Available IPv4 Addresses: 251
Availability Zone: us-east-1c (use1-az6)
State: available
IPv4 CIDR: 10.10.7.0/24
IPv6 CIDR: -
Route Table: rtb-0b57165cbccac0767 | First-RT

Action Menu

- First, select **Subnets** from the left-hand menu.
- Then click the first Public Subnet and select **Modify Auto-Assign IP Settings** from the Actions menu.
- Then click the checkmark of **Enable Auto-Assign Public IPv4 Address** option as you see in the picture below and save it



Subnets > Modify auto-assign IP settings

Modify auto-assign IP settings

Enable the auto-assign IP address setting to automatically request a public IPv4 or IPv6 address for an instance launched in this subnet. You can override the auto-assign IP settings for an instance at launch time.

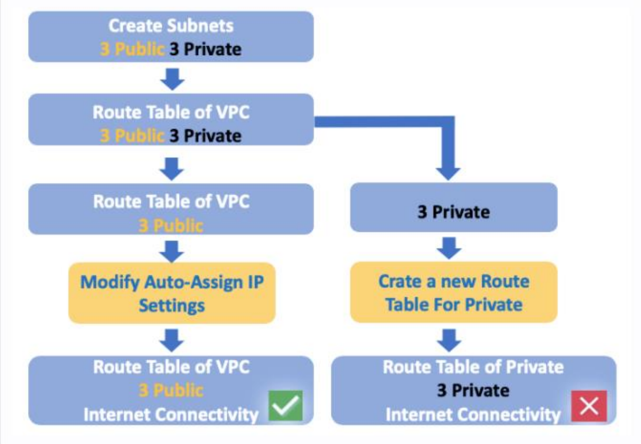
Subnet ID: subnet-0b9ad2e5db8c9f954

Auto-assign IPv4: ☑ Enable auto-assign public IPv4 address

Enable Auto-Assign Public IPv4 Address

After this process, public IP addresses will be given to the machines that we will create in this public subnet and repeat this action for the other 2 public subnets.

Conclusion: PublicSubnet&Private Subnet



Public Subnet&Private Subnet