

# Assignment – VPC

## Production Network

### 1) VPC Creation

Created VPC 'prod-vpc' with CIDR 10.0.0.0/16

VPC > Your VPCs > vpc-07ce801262200963f

vpc-07ce801262200963f / **prod-vpc** Actions ▼

**Details** Info

VPC ID vpc-07ce801262200963f	State Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-0f59dd4ee3f4ed213	Main route table rtb-07fe415e261df2d5f	Main network ACL acl-059f4b22c28c169df
Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool -	IPv6 CIDR -
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID 992382547561	

### 2) Subnet Creation

Created 5 subnets web, app1, app2, dbcache and db

CIDRs for web, app1, app2, dbcache and db are 10.0.0.0/24, 10.0.1.0/24, 10.0.2.0/24, 10.0.3.0/24 and 10.0.4.0/24 resp.

**Subnets (5)** Info Last updated 37 minutes ago Actions ▼ Create subnet

Find resources by attribute or tag

<input type="checkbox"/>	Name	Subnet ID	State	VPC	IPv4 CIDR
<input type="checkbox"/>	web	subnet-004d3fa79dfe22033	Available	vpc-07ce801262200963f   prod...	10.0.0.0/24
<input type="checkbox"/>	app1	subnet-0cd10735770e46265	Available	vpc-07ce801262200963f   prod...	10.0.1.0/24
<input type="checkbox"/>	app2	subnet-0400d7375d1676d4d	Available	vpc-07ce801262200963f   prod...	10.0.2.0/24
<input type="checkbox"/>	dbcache	subnet-099c2f388251d5185	Available	vpc-07ce801262200963f   prod...	10.0.3.0/24
<input type="checkbox"/>	db	subnet-0d4740408054201e6	Available	vpc-07ce801262200963f   prod...	10.0.4.0/24

### 3) Route Table creation

Because there must be 1 public and 4 private subnets, 2 route tables will have to be created. Internet gateway will get added to one of the route tables with which web (public) subnet will also get associated.

**Route tables (2)** Info Last updated less than a minute ago Actions ▼ Create route table

Find resources by attribute or tag

<input type="checkbox"/>	Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
<input type="checkbox"/>	rt-public	rtb-07fe415e261df2d5f	-	-	Yes	vpc-07ce801262200963f
<input type="checkbox"/>	rt-private	rtb-0d6f2471e42f3961f	-	-	No	vpc-07ce801262200963f

#### 4) Associate subnets with the route table

To associate subnets with route table, select the subnet and click the EDIT SUBNET ASSOCIATIONS on the SUBNET ASSOCIATION tab.

Routes **Subnet associations** Edge associations Route propagation Tags

Explicit subnet associations (0) [Edit subnet associations](#)

< 1 > ⚙

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
No subnet associations			
You do not have any subnet associations.			

Add web subnet to rt-public subnet:

VPC > Route tables > **rtb-07fe415e261df2d5f** > Edit subnet associations

### Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/5)

< 1 > ⚙

	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input type="checkbox"/>	db	<a href="#">subnet-0d4740408054201e6</a>	10.0.4.0/24	-	<a href="#">rtb-0d6f2471e42f3961f / rt-private</a>
<input type="checkbox"/>	app1	<a href="#">subnet-0cd10735770e46265</a>	10.0.1.0/24	-	<a href="#">rtb-0d6f2471e42f3961f / rt-private</a>
<input type="checkbox"/>	dbcache	<a href="#">subnet-099c2f388251d5185</a>	10.0.3.0/24	-	<a href="#">rtb-0d6f2471e42f3961f / rt-private</a>
<input checked="" type="checkbox"/>	web	<a href="#">subnet-004d3fa79dfe22033</a>	10.0.0.0/24	-	<a href="#">rtb-07fe415e261df2d5f / rt-public</a>
<input type="checkbox"/>	app2	<a href="#">subnet-0400d7375d1676d4d</a>	10.0.2.0/24	-	<a href="#">rtb-0d6f2471e42f3961f / rt-private</a>

Selected subnets

[subnet-004d3fa79dfe22033 / web](#) ✕

Cancel [Save associations](#)

Add app1, app2, dbcache and db subnets to rt-private subnet:

VPC > Route tables > **rtb-0d6f2471e42f3961f** > Edit subnet associations

### Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (4/5)

< 1 > ⚙

	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/>	db	<a href="#">subnet-0d4740408054201e6</a>	10.0.4.0/24	-	<a href="#">Main (rtb-07fe415e261df2d5f / rt-pub.</a>
<input checked="" type="checkbox"/>	app1	<a href="#">subnet-0cd10735770e46265</a>	10.0.1.0/24	-	<a href="#">Main (rtb-07fe415e261df2d5f / rt-pub.</a>
<input checked="" type="checkbox"/>	dbcache	<a href="#">subnet-099c2f388251d5185</a>	10.0.3.0/24	-	<a href="#">Main (rtb-07fe415e261df2d5f / rt-pub.</a>
<input type="checkbox"/>	web	<a href="#">subnet-004d3fa79dfe22033</a>	10.0.0.0/24	-	<a href="#">rtb-07fe415e261df2d5f / rt-public</a>
<input checked="" type="checkbox"/>	app2	<a href="#">subnet-0400d7375d1676d4d</a>	10.0.2.0/24	-	<a href="#">Main (rtb-07fe415e261df2d5f / rt-pub.</a>

Selected subnets

[subnet-0d4740408054201e6 / db](#) ✕ [subnet-0cd10735770e46265 / app1](#) ✕ [subnet-099c2f388251d5185 / dbcache](#) ✕ [subnet-0400d7375d1676d4d / app2](#) ✕

Cancel [Save associations](#)

## 5) Make web subnet public

To make a subnet public, it has to be routed to an internet gateway

As we have already added the subnet to the rt-public route table, we will now create an Internet Gateway and add to the same route table to make the web subnet public.

Internet Gateway creation:

igw-04a9e180a813c897a / prod-igw

Actions

Details Info

Internet gateway ID	State	VPC ID	Owner
igw-04a9e180a813c897a	Detached	-	992382547561

Attach the Internet Gateway to VPC:

Internet gateways (1/1) Info

Search

Name	Internet gateway ID	State	VPC ID	Owner
prod-igw	igw-04a9e180a813c897a	Detached	-	992382547561

Actions

- View details
- Attach to VPC
- Detach from VPC
- Manage tags
- Delete internet gateway

Create internet gateway

Select option to attach VPC

VPC > Internet gateways > Attach to VPC (igw-04a9e180a813c897a)

## Attach to VPC (igw-04a9e180a813c897a) Info

**VPC**

Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs

Attach the internet gateway to this VPC.

Q vpc-07ce801262200963f X

AWS Command Line Interface command

Cancel Attach internet gateway

Select a VPC

Internet gateways (1) Info

Search

Name	Internet gateway ID	State	VPC ID	Owner
prod-igw	igw-04a9e180a813c897a	Attached	vpc-07ce801262200963f   prod-vpc	992382547561

VPC attached

Add Internet Gateway to the Route Table:

Select the Route Table & go to the ROUTES tab. Select EDIT ROUTES.

Find resources by attribute or tag						
< 1 > ⚙						
<input checked="" type="checkbox"/>	Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
<input checked="" type="checkbox"/>	rt-public	rtb-07fe415e261df2d5f	-	-	Yes	vpc-07ce8012622005

rtb-07fe415e261df2d5f / rt-public

Details Routes Subnet associations Edge associations Route propagation Tags

Routes (1)

Both Edit routes

Filter routes

< 1 > ⚙

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

In Edit routes screen, click ADD ROUTE button to add the Internet Gateway

VPC > Route tables > rtb-07fe415e261df2d5f > Edit routes

### Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
<input type="text" value="local"/>			

Add route

Cancel Preview Save changes

Once a new field to include a new route is added, select the destination CIDR, Target service to route and its ID

In this case, destination CIDR is '0.0.0.0/0' so that the server can communicate to any destination IP. Target routing service is Internet Gateway and the ID is the ID of internet gateway we have created.

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
<input type="text" value="local"/>			
0.0.0.0/0	Internet Gateway	-	No
<input type="text" value="igw-04a9e180a813c897a"/>			

Add route

Cancel Preview Save changes

After the internet gateway and subnets are added, the association can be verified in the respective tabs under the route table

Routes	Subnet associations	Edge associations	Route propagation	Tags
<b>Routes (2)</b>				
<input type="text" value="Filter routes"/>				
Destination	Target	Status	Propagated	
0.0.0.0/0	<a href="#">igw-04a9e180a813c897a</a>	Active	No	
10.0.0.0/16	local	Active	No	

Routes	Subnet associations	Edge associations	Route propagation	Tags
<b>Explicit subnet associations (1)</b>				
<input type="text" value="Find subnet association"/>				
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	
web	<a href="#">subnet-004d3fa79dfe22033</a>	10.0.0.0/24	-	

## 6) Add NAT gateway for private subnets

NAT gateway helps private subnets connect to the services outside VPC while keeping their IP addresses private. All inbound communications are blocked in NAT gateway.

NAT gateway creation:

Go to NAT gateway and click CREATE NAT GATEWAY

### Create NAT gateway [Info](#)

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.

#### NAT gateway settings

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

The name can be up to 256 characters long.

**Subnet**  
Select a subnet in which to create the NAT gateway.

subnet-004d3fa79dfe22033 (web) ▾

**Connectivity type**  
Select a connectivity type for the NAT gateway.

☒ Public  
☐ Private

**Elastic IP allocation ID [Info](#)**  
Assign an Elastic IP address to the NAT gateway.

eipalloc-0dcf7fcc62ba95ccd ▾ Allocate Elastic IP

▶ **Additional settings** [Info](#)

Select the VPC under which the NAT gateway must be created.

Select the public subnet because a NAT gateway will always be inside a public subnet.

Click ALLOCATE ELASTIC IP and finally click create NAT gateway

Add NAT gateway to route table:

Now the NAT gateway has to be added to route table that is associated with private subnets

VPC > Route tables > rtb-0d6f2471e42f3961f

rtb-0d6f2471e42f3961f / rt-private

Actions

DetailsInfo

Route table ID

rtb-0d6f2471e42f3961f

VPC

vpc-07ce801262200963f | prod-vpc

Main

No

Owner ID

992382547561

Explicit subnet associations

4 subnets

Edge associations

-

Routes

Subnet associations

Edge associations

Route propagation

Tags

Routes (1)

BothEdit routes

Filter routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

VPC > Route tables > rtb-0d6f2471e42f3961f > Edit routes

Edit routes

Destination

10.0.0.0/16

Target

local

Status

Active

Propagated

No

0.0.0.0/0

local

NAT Gateway

-

No

Remove

nat-01630c6c629d5098d

Add route

Cancel

Preview

Save changes

After the NAT gateway is added to the route table, we can find the link in the tab as show below

Routes

Subnet associations

Edge associations

Route propagation

Tags

Routes (2)

BothEdit routes

Filter routes

Destination	Target	Status	Propagated
0.0.0.0/0	nat-01630c6c629d5098d	Active	No
10.0.0.0/16	local	Active	No

## 7) EC2 instances creation:

Create a key-pair:

The 'Create key pair' dialog shows the following configuration:

- Key pair name:** vpcassignment
- Key pair type:** RSA (selected), ED25519
- Private key file format:** .pem (selected), .ppk
- Warning:** When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance.
- Buttons:** Cancel, Create key pair

Web server instance creation:

Select the key-pair, VPC and subnet (web). As this server must communicate with services outside VPC, a public IP will be assigned. We will simultaneously create a new security group for all EC2 instances.

The 'Network settings' section shows the following configuration:

- Key pair name:** vpc-assignment
- VPC:** vpc-07ce801262200963f (prod-vpc)
- Subnet:** subnet-004d3fa79dfe22033 (web)
- Auto-assign public IP:** Enable
- Firewall (security groups):** Create security group (selected), Select existing security group
- Security group name:** prod-web-sg

As the other 4 servers will be private, public IP won't be created for them keeping the rest of the steps same. The final EC2 instances will look like below

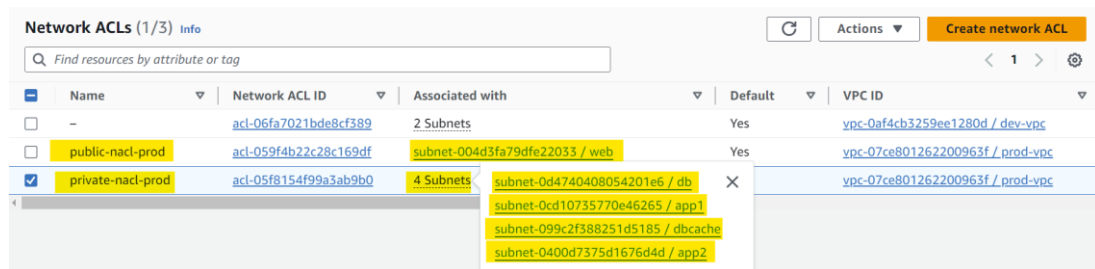
Instances (6) Info							
Find Instance by attribute or tag (case-sensitive)							
All states							
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	
<input type="checkbox"/>	prod-app1-instance	i-0c6a9bfd48f349360	Running	t2.micro	2/2 checks passed	View alarms	
<input type="checkbox"/>	prod-app2-instance	i-019a32bf3faa42ad	Running	t2.micro	2/2 checks passed	View alarms	
<input type="checkbox"/>	prod-db-instance	i-09aeb277f6040e9fe	Running	t2.micro	2/2 checks passed	View alarms	
<input type="checkbox"/>	prod-dbcache-instance	i-02731f62efd379853	Running	t2.micro	2/2 checks passed	View alarms	
<input type="checkbox"/>	prod-web-instance	i-0c9529877866fa6fe	Running	t2.micro	-	View alarms	

## 8) Configure NACL and security groups for security

NACL creation:

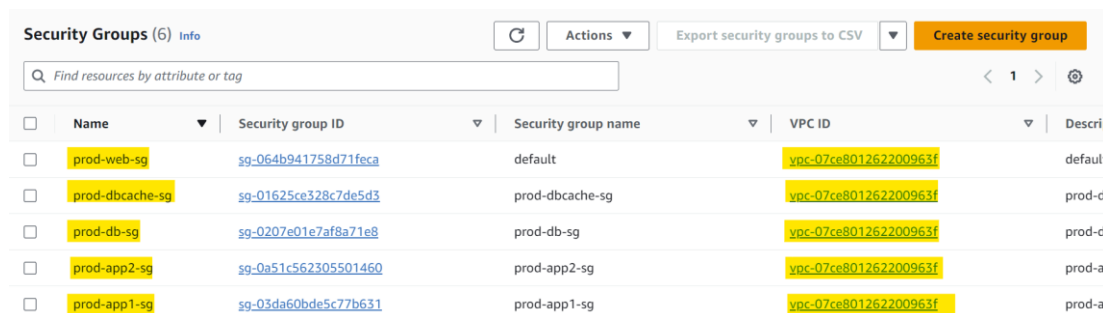
We will need 2 NACLs for private and public subnets.

A default NACL is automatically created for every VPC. We can use the same NACL out of the two Renamed default NACL for including public subnet and created a NACL for the private subnets



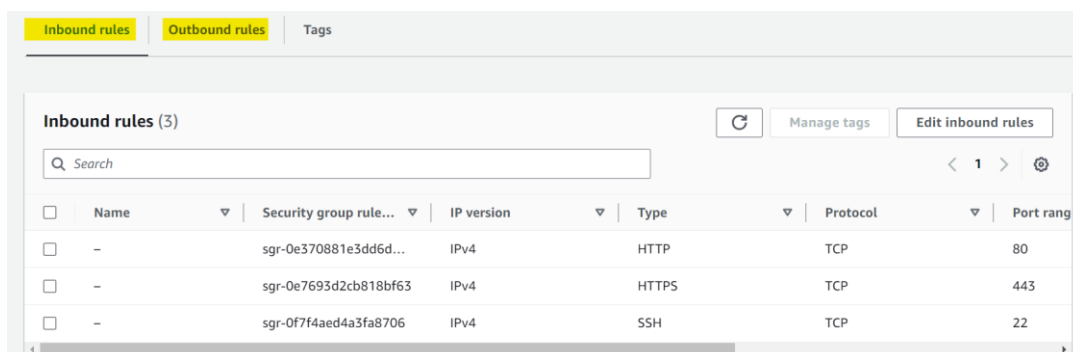
Name	Network ACL ID	Associated with	Default	VPC ID
-	acl-06fa7021bde8cf389	2 Subnets	Yes	vpc-0af4cb3259ee1280d / dev-vpc
public-nacl-prod	acl-059f4b22c28c169df	subnet-004d3fa79dfe22033 / web	Yes	vpc-07ce801262200963f / prod-vpc
private-nacl-prod	acl-05f8154f99a3ab9b0	4 Subnets subnet-0d4740408054201e6 / db subnet-0cd10735770e46265 / app1 subnet-099c2f388251d5185 / dbcache subnet-0400d7375d1676d4d / app2	X	vpc-07ce801262200963f / prod-vpc

Security Group creation:



Name	Security group ID	Security group name	VPC ID
prod-web-sg	sg-064b941758d71feca	default	vpc-07ce801262200963f
prod-dbcache-sg	sg-01625ce328c7de5d3	prod-dbcache-sg	vpc-07ce801262200963f
prod-db-sg	sg-0207e01e7af8a71e8	prod-db-sg	vpc-07ce801262200963f
prod-app2-sg	sg-0a51c562305501460	prod-app2-sg	vpc-07ce801262200963f
prod-app1-sg	sg-03da60bde5c77b631	prod-app1-sg	vpc-07ce801262200963f

Declare inbound and outbound rules of NACL and Security Groups:



Name	Security group rule...	IP version	Type	Protocol	Port rang
-	sgr-0e370881e3dd6d...	IPv4	HTTP	TCP	80
-	sgr-0e7693d2cb818bf63	IPv4	HTTPS	TCP	443
-	sgr-0f7f4aed4a3fa8706	IPv4	SSH	TCP	22



## Development Network

### 1) VPC Creation

The screenshot shows the AWS VPC console for a VPC named 'dev-vpc' (ID: vpc-0af4cb3259ee1280d). The VPC is in an 'Available' state. Key details include:

- VPC ID:** vpc-0af4cb3259ee1280d
- State:** Available
- DNS hostnames:** Disabled
- DNS resolution:** Enabled
- Tenancy:** Default
- DHCP option set:** dopt-0f59dd4ee3f4ed213
- Main route table:** rtb-0d9bce25cbbf50cc0 / rt-public-dev
- Main network ACL:** acl-06fa7021bde8cf389
- Default VPC:** No
- IPv4 CIDR:** 107.12.0.0/16
- IPv6 pool:** -
- IPv6 CIDR:** -
- Network Address Usage metrics:** Disabled
- Route 53 Resolver DNS Firewall rule groups:** -
- Owner ID:** 992382547561

### 2) Subnet Creation

The screenshot shows the AWS Subnets console with a list of subnets. The subnets are:

Name	Subnet ID	State	VPC	IPv4 CIDR
db-dev	subnet-0874576f3538f52f6	Available	vpc-0af4cb3259ee1280d / dev-...	107.12.1.0/24
web-dev	subnet-02d2a4001d5346842	Available	vpc-0af4cb3259ee1280d / dev-...	107.12.0.0/24

### 3) Route Table creation

Because there must be 1 public and 1 private subnet, 2 route tables will have to be created. Internet gateway will get added to one of the route tables with which web (public) subnet will also get associated.

The screenshot shows the AWS Route Tables console with a list of route tables. The route tables are:

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
rt-public-dev	rtb-0d9bce25cbbf50cc0	subnet-02d2a4001d5346...	-	Yes	vpc-0af4cb3259ee128...
rt-private-dev	rtb-0d8d3fde1758584a0	subnet-0874576f3538f5...	-	No	vpc-0af4cb3259ee128...

### 4) Associate subnets with the route table

Steps to associate subnets with route table is same as mentioned production setup.

Below is a screenshot after the association is done

The screenshot shows the AWS Route Tables console after subnet associations. The route tables are:

Name	Route table ID	Explicit subnet associations	Edge associations	Main	V
rt-public-dev	rtb-0d9bce25cbbf50cc0	subnet-02d2a4001d5346842 / web-dev	-	Yes	vi
rt-private-dev	rtb-0d8d3fde1758584a0	subnet-0874576f3538f52f6 / db-dev	-	No	vi
rt-private	rtb-0d6f2471e42f3961f	4 subnets	-	No	vi

## 5) Make web subnet public

To make a subnet public, it has to be routed to an internet gateway

As we have already added the subnet to the rt-public-dev route table, we will now create an Internet Gateway and add to the same route table to make the web subnet public.

Internet Gateway creation:

VPC > Internet gateways > igw-0975195833ead7d17

### igw-0975195833ead7d17 / dev-igw

Actions

**Details** Info

Internet gateway ID igw-0975195833ead7d17	State Attached	VPC ID vpc-0af4cb3259ee1280d   dev-vpc	Owner 992382547561
--	-------------------	---	-----------------------

**Tags** Manage tags

Search tags

Key	Value
Name	dev-igw

Attach the Internet Gateway to VPC:

Internet gateways (2) Info

Search

Actions Create internet gateway

Name	Internet gateway ID	State	VPC ID	Owner
dev-igw	igw-0975195833ead7d17	Attached	vpc-0af4cb3259ee1280d   dev-vpc	992382547561

VPC attached

Add Internet Gateway to the Route Table:

VPC > Route tables > rtb-0d9bce25cbbf50cc0

### rtb-0d9bce25cbbf50cc0 / rt-public-dev

Actions

**Details** Info

Route table ID rtb-0d9bce25cbbf50cc0	Main Yes	Explicit subnet associations subnet-02d2a4001d5346842 / web-dev	Edge associations -
VPC vpc-0af4cb3259ee1280d   dev-vpc	Owner ID 992382547561		

**Routes** Subnet associations Edge associations Route propagation Tags

**Routes (2)** Both Edit routes

Filter routes

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0975195833ead7d17	Active	No
107.12.0.0/16	local	Active	No

Add subnet to route table:

VPC > Route tables > rtb-0d9bce25cbbf50cc0

rtb-0d9bce25cbbf50cc0 / **rt-public-dev** Actions ▾

**Details** Info

Route table ID rtb-0d9bce25cbbf50cc0	Main Yes	Explicit subnet associations subnet-02d2a4001d5346842 / web-dev	Edge associations -
VPC vpc-0af4cb3259ee1280d   dev-vpc	Owner ID 992382547561		

Routes Subnet associations Edge associations Route propagation Tags

**Explicit subnet associations (1)** Edit subnet associations

Find subnet association

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
web-dev	subnet-02d2a4001d5346842	107.12.0.0/24	-

## 6) Add NAT gateway for private subnets

NAT gateway helps private subnets connect to the services outside VPC while keeping their IP addresses private. All inbound communications are blocked in NAT gateway.

NAT gateway creation:

VPC > NAT gateways > nat-062aeeb88b939e2c6

nat-062aeeb88b939e2c6 / **dev-nat-gateway** Actions ▾

**Details**

NAT gateway ID nat-062aeeb88b939e2c6	Connectivity type Public	State Pending	State message -
NAT gateway ARN arn:aws:ec2:eu-west-1:992382547561:natgateway/nat-062aeeb88b939e2c6	Primary public IPv4 address -	Primary private IPv4 address 107.12.0.86	Primary network interface ID eni-0d1ee57af23584f1a
VPC vpc-0af4cb3259ee1280d   dev-vpc	Subnet subnet-02d2a4001d5346842 / web-dev	Created Monday 22 July 2024 at 01:15:26 GMT+5:30	Deleted -

Add NAT gateway to route table:

Now the NAT gateway has to be added to route table that is associated with private subnets

VPC > Route tables > rtb-0d8d3fde1758584a0

rtb-0d8d3fde1758584a0 / **rt-private-dev** Actions ▾

**Details** Info

Route table ID rtb-0d8d3fde1758584a0	Main No	Explicit subnet associations subnet-0874576f3538f52f6 / db-dev	Edge associations -
VPC vpc-0af4cb3259ee1280d   dev-vpc	Owner ID 992382547561		

Routes Subnet associations Edge associations Route propagation Tags

**Routes (2)** Both Edit routes

Filter routes

Destination	Target	Status	Propagated
0.0.0.0/0	nat-062aeeb88b939e2c6	Active	No
107.12.0.0/16	local	Active	No

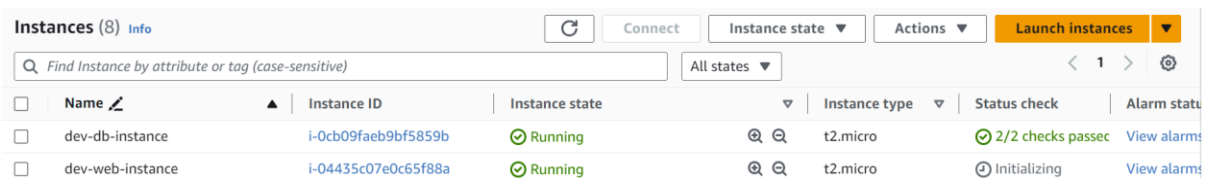
## 7) EC2 instances creation:

We can use the same key-pair for development environment.

Like production, web instance in development will also have a public IP and db instance won't as it is private.

The instance creation steps will be the same.

After creation the instances will like below



The screenshot shows the 'Instances (8)' page in the AWS Management Console. It includes a search bar, a table of instances, and various action buttons. The table lists two instances: 'dev-db-instance' and 'dev-web-instance', both in a 'Running' state.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
dev-db-instance	i-0cb09faeb9bf5859b	Running	t2.micro	2/2 checks passed	View alarms
dev-web-instance	i-04435c07e0c65f88a	Running	t2.micro	Initializing	View alarms

## 8) Configure NACL and security groups for security

NACL creation:

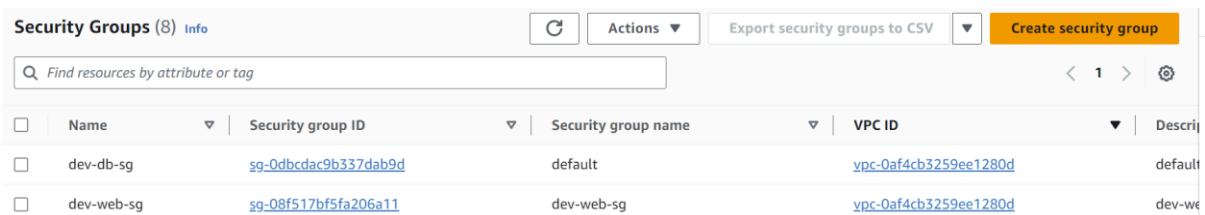
Like production, here too we will need 2 NACLs for private and public subnets.



The screenshot shows the 'Network ACLs (4)' page in the AWS Management Console. It includes a search bar, a table of network ACLs, and a 'Create network ACL' button. The table lists two network ACLs: 'public-nacl-dev' and 'private-nacl-dev'.

Name	Network ACL ID	Associated with	Default	VPC ID
public-nacl-dev	acl-06fa7021bde8cf389	subnet-02d2a4001d5346842 / web-dev	Yes	vpc-0af4cb3259ee1280d / dev-vpc
private-nacl-dev	acl-0ceb0f56826a41012	subnet-0874576f3538f52f6 / db-dev	No	vpc-0af4cb3259ee1280d / dev-vpc

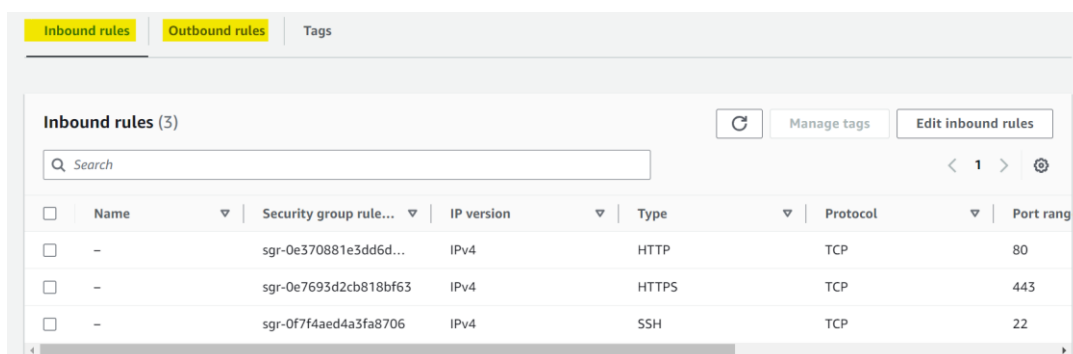
Security Group creation:



The screenshot shows the 'Security Groups (8)' page in the AWS Management Console. It includes a search bar, a table of security groups, and a 'Create security group' button. The table lists two security groups: 'dev-db-sg' and 'dev-web-sg'.

Name	Security group ID	Security group name	VPC ID	Description
dev-db-sg	sg-0dbcdac9b337dab9d	default	vpc-0af4cb3259ee1280d	default
dev-web-sg	sg-08f517bf5fa206a11	dev-web-sg	vpc-0af4cb3259ee1280d	dev-we

Declare inbound and outbound rules of NACL and Security Groups:



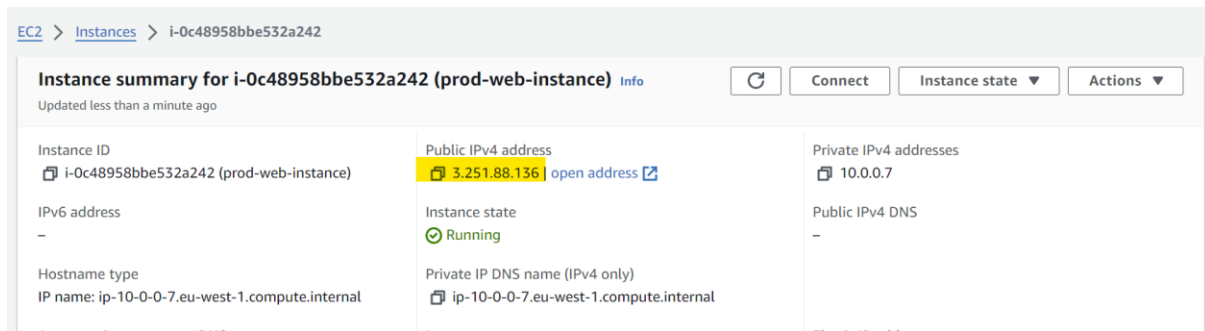
The screenshot shows the 'Inbound rules (3)' page in the AWS Management Console. It includes a search bar, a table of inbound rules, and buttons for 'Manage tags' and 'Edit inbound rules'. The table lists three inbound rules for a security group.

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-0e370881e3dd6d...	IPv4	HTTP	TCP	80
-	sgr-0e7693d2cb818bf63	IPv4	HTTPS	TCP	443
-	sgr-0f7f4aed4a3fa8706	IPv4	SSH	TCP	22

## Testing connection between private and public networks (with PuTTY)

Production:

The public IP of web instance is **3.251.88.136**, and we will configure the same in PuTTY.

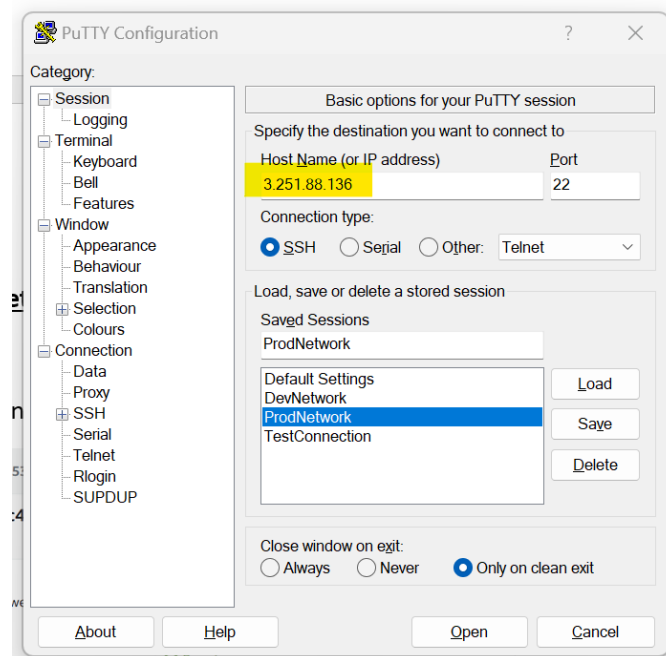


EC2 > Instances > i-0c48958bbe532a242

**Instance summary for i-0c48958bbe532a242 (prod-web-instance)** Info

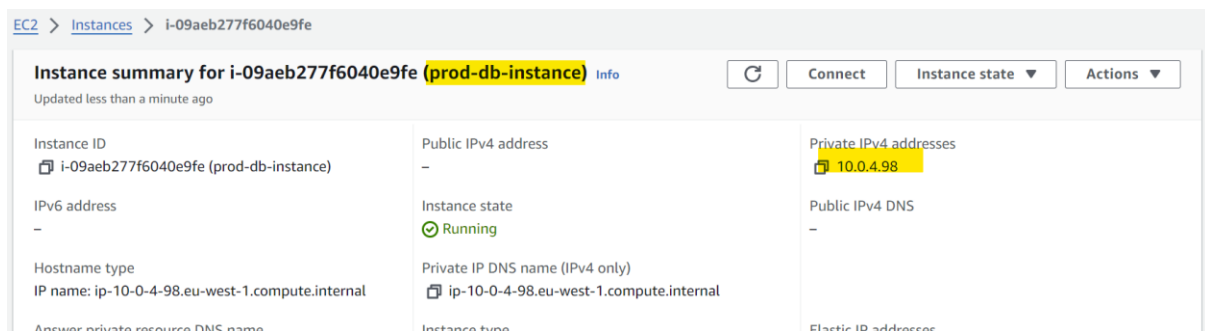
Updated less than a minute ago

Instance ID i-0c48958bbe532a242 (prod-web-instance)	Public IPv4 address <b>3.251.88.136</b> <a href="#">open address</a>	Private IPv4 addresses 10.0.0.7
IPv6 address -	Instance state <b>Running</b>	Public IPv4 DNS -
Hostname type IP name: ip-10-0-0-7.eu-west-1.compute.internal	Private IP DNS name (IPv4 only) ip-10-0-0-7.eu-west-1.compute.internal	



Now, we will ping any of the private servers (app1, app2, db, dbcache) from the configured web instance.

Private IP of db instance is **10.0.4.98**, which we will ping to see if successful connection gets established



EC2 > Instances > i-09aeb277f6040e9fe

**Instance summary for i-09aeb277f6040e9fe (prod-db-instance)** Info

Updated less than a minute ago

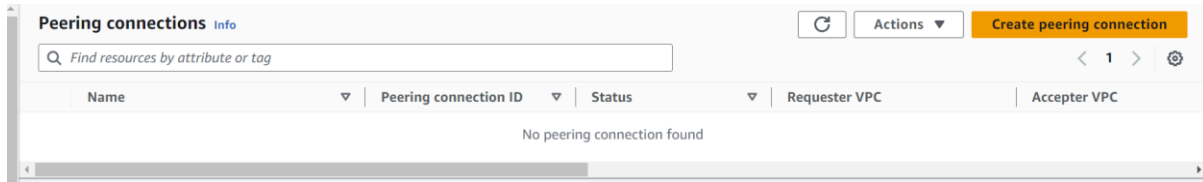
Instance ID i-09aeb277f6040e9fe (prod-db-instance)	Public IPv4 address -	Private IPv4 addresses <b>10.0.4.98</b>
IPv6 address -	Instance state <b>Running</b>	Public IPv4 DNS -
Hostname type IP name: ip-10-0-4-98.eu-west-1.compute.internal	Private IP DNS name (IPv4 only) ip-10-0-4-98.eu-west-1.compute.internal	

Same steps can be followed for testing connection of *development* network.

```
ec2-user@ip-10-0-0-7:~  
Using username "ec2-user".  
Authenticating with public key "imported-openssh-key"  
  
#_##### Amazon Linux 2023  
~\#####\  
~~\#####\  
~~\####|  
~~\##/  
~~V~'-'>  
~~~~  
~~~~.  
~~~~/_m/'-/->  
Last login: Mon Jul 22 08:30:24 2024 from 111.125.240.53  
[ec2-user@ip-10-0-0-7 ~]$ ping 10.0.4.98  
PING 10.0.4.98 (10.0.4.98) 56(84) bytes of data:  
64 bytes from 10.0.4.98: icmp_seq=1 ttl=127 time=0.842 ms  
64 bytes from 10.0.4.98: icmp_seq=2 ttl=127 time=0.672 ms  
64 bytes from 10.0.4.98: icmp_seq=3 ttl=127 time=0.494 ms  
64 bytes from 10.0.4.98: icmp_seq=4 ttl=127 time=0.462 ms  
64 bytes from 10.0.4.98: icmp_seq=5 ttl=127 time=0.490 ms  
64 bytes from 10.0.4.98: icmp_seq=6 ttl=127 time=0.536 ms  
64 bytes from 10.0.4.98: icmp_seq=7 ttl=127 time=0.492 ms  
64 bytes from 10.0.4.98: icmp_seq=8 ttl=127 time=0.538 ms  
64 bytes from 10.0.4.98: icmp_seq=9 ttl=127 time=0.514 ms  
^C  
--- 10.0.4.98 ping statistics ---  
9 packets transmitted, 9 received, 0% packet loss, time 8357ms  
rtt min/avg/max/mdev = 0.462/0.560/0.842/0.114 ms  
[ec2-user@ip-10-0-0-7 ~]$
```

## Peering production and development networks

To create a peering connection, in VPC dashboard, select 'Peering Connections' and click Create peering connection.



Enter peering name and select requester VPC and acceptor VPC

**Peering connection settings**

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

**Select a local VPC to peer with**

VPC ID (Requester)

VPC CIDRs for vpc-07ce801262200963f (prod-vpc)

CIDR	Status	Status reason
10.0.0.0/16	✔ Associated	-

**Select another VPC to peer with**

Account  
☒ My account  
☐ Another account

Region  
☒ This Region (eu-west-1)  
☐ Another Region

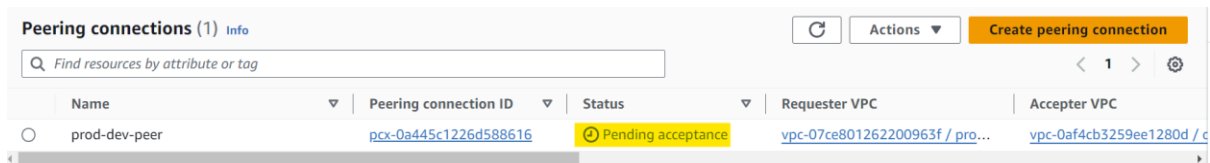
VPC ID (Acceptor)

VPC CIDRs for vpc-0af4cb3259ee1280d (dev-vpc)

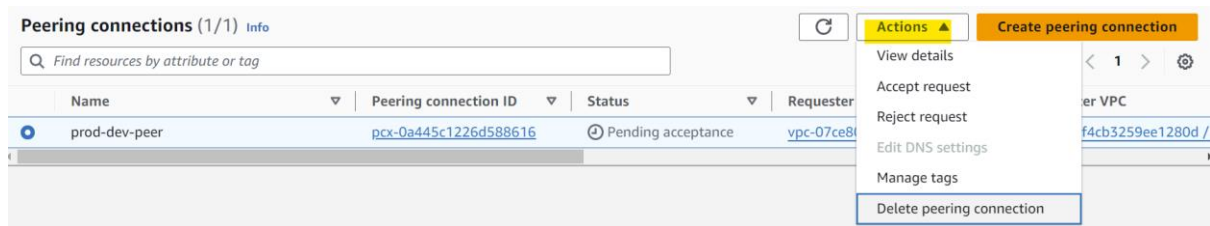
CIDR	Status	Status reason
107.12.0.0/16	✔ Associated	-

Once done, click 'Create peering connection' button at the bottom

Once peering created, the status is Pending Acceptance until user accepts the connection request

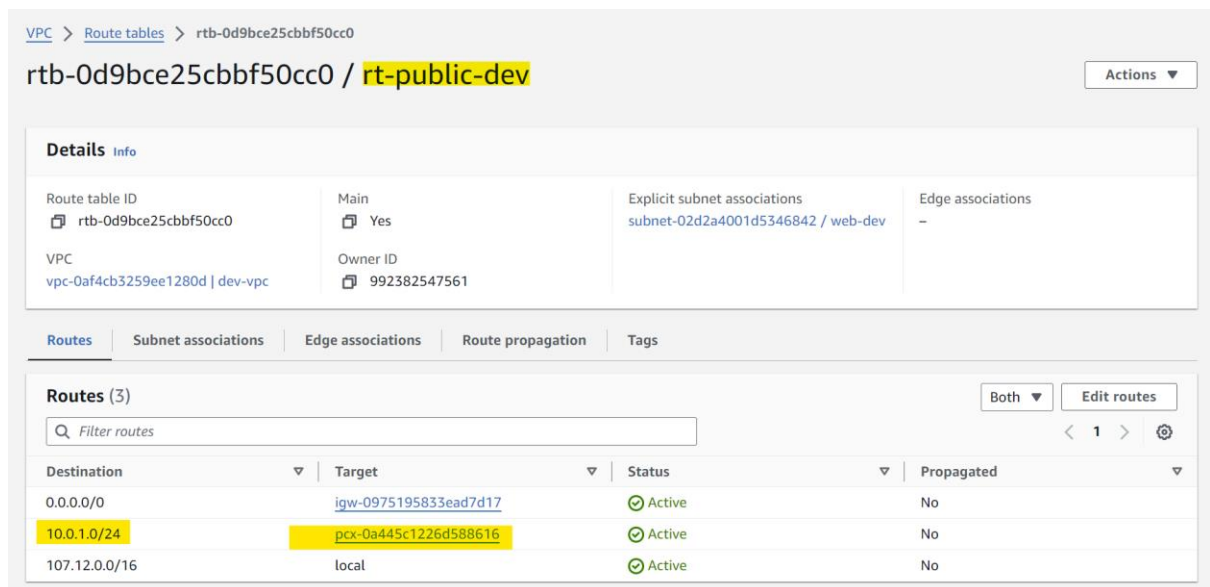


Choose the peering connection, click actions and select Accept Request. Follow the screenshot



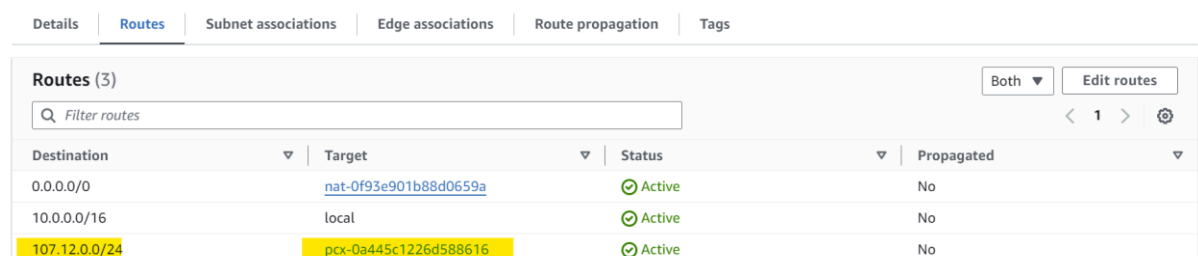
## Test peering connections

To test the established peered connection we will configure the add the peering connection in public and private route tables in both production and development networks and destination CIDR of the required subnets.



Add peering connection in development public route

## rtb-0d6f2471e42f3961f / rt-private



Add peering connection in production private route



## Testing the connection in PuTTY

Launch PuTTY with web instance of development environment and ping app1 from production network to see if the peering was successful.

Subnets (2) Info

Last updated 22 minutes ago

Actions

Create subnet

subnet-02da4001d5346842

×

subnet-0cd10735770e46265

×

Clear filters

< 1 > ⚙

<input type="checkbox"/>	Name	Subnet ID	State	VPC	IPv4 CIDR
<input type="checkbox"/>	web-dev	<a href="#">subnet-02da4001d5346842</a>	Available	<a href="#">vpc-0af4cb3259ee1280d</a>   dev...	107.12.0.0/24
<input type="checkbox"/>	app1	<a href="#">subnet-0cd10735770e46265</a>	Available	<a href="#">vpc-07ce801262200963f</a>   prod...	10.0.1.0/24

107.12.0.0/24 is CIDR of web subnet in development network and 10.0.1.0/24 is CIDR of app1 in production network. The same can be found in the screenshot below.

```
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"

#_
~\##### Amazon Linux 2023
~~\#####
~~\###|
~~\#/ https://aws.amazon.com/linux/amazon-linux-2023
~~V~'-'>
~~~
~~~
~~~
~~~
Last login: Mon Jul 22 09:01:36 2024 from 111.125.240.53
[ec2-user@ip-107-12-0-82 ~]$ ping 10.0.1.231
PING 10.0.1.231 (10.0.1.231) 56(84) bytes of data.
64 bytes from 10.0.1.231: icmp_seq=1 ttl=127 time=0.488 ms
64 bytes from 10.0.1.231: icmp_seq=2 ttl=127 time=0.731 ms
64 bytes from 10.0.1.231: icmp_seq=3 ttl=127 time=0.618 ms
64 bytes from 10.0.1.231: icmp_seq=4 ttl=127 time=0.532 ms
64 bytes from 10.0.1.231: icmp_seq=5 ttl=127 time=0.476 ms
64 bytes from 10.0.1.231: icmp_seq=6 ttl=127 time=0.588 ms
^C
--- 10.0.1.231 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5223ms
rtt min/avg/max/mdev = 0.476/0.572/0.731/0.087 ms
[ec2-user@ip-107-12-0-82 ~]$
```

As the ping is receiving a response, we can conclude that peering between production and development network was established successfully.