

Configure Private App-to-DB Connectivity Across Two VPCs (Using NAT + VPC Peering)

By: Nikhilesh Sakhare

Prerequisites

- AWS account access to VPC and EC2 services.
- Region used: **Asia Pacific (Mumbai)**.
- Key pair available: **sample-2-ind.pem**.

Step 1 — Create VPC-1 (MyVPC-1)

1. Open VPC → Your VPCs → Create VPC and select VPC only.

The screenshot shows the AWS VPC Dashboard. On the left, there's a sidebar with options like VPC dashboard, Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only Internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, Route servers), and Security. The main area has sections for Resources by Region (VPCs, Subnets, Route Tables, Internet Gateways) and NAT Gateways, all showing counts for the Mumbai region. To the right are Service Health, Settings (with Block Public Access and Zones), and Additional Information (with VPC Documentation, All VPC Resources, Forums, and Report an Issue). At the bottom, there's a search bar, CloudShell, Feedback, and Console Mobile App.

This screenshot shows the 'Your VPCs' section of the VPC dashboard. It lists one VPC named 'VPC encryption controls - new' with ID 'vpc-0387f1efb5c5f5fe'. The table includes columns for Name, VPC ID, State, Encryption controls, Encryption control status, Block Public Access, and IP version. The state is 'Available' and the block public access is 'Off'. There are buttons for Actions and Create VPC. Below the table, it says 'Select a VPC above'.

2. Set Name: MyVPC-1 and IPv4 CIDR: 10.0.0.0/16.

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 - [optional](#)
Creates a tag with a key of 'Name' and a value that you specify.
MyVPC-1

IPv4 CIDR block [Info](#)
 IPv4 CIDR manual input IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.0.0.0/16

IPv6 CIDR block [Info](#)
 No IPv6 CIDR block IPAM-allocated IPv6 CIDR block

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

3. Create the VPC and verify VPC ID is created.

You successfully created vpc-059374692d8327522 / MyVPC-1

vpc-059374692d8327522 / MyVPC-1

Actions

Details [Info](#)

VPC ID vpc-059374692d8327522	State Available	Block Public Access Off	DNS hostnames Disabled
DNS resolution Enabled	Tenancy default	DHCP option set dopt-0609942e1829dfcf0	Main route table rtb-0c7aa5b2e1f658016
Main network ACL acl-01954f67fbfb0adc	Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool -
IPv6 CIDR (Network border group) -	Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID 673586x3
Encryption control ID -	Encryption control mode -	Resource map Info	

Resource map | CIDRs | Flow logs | Tags | Integrations

Resource map [Info](#) Show all details

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 2 — Create public and private subnets in VPC-1

1. Go to VPC → Subnets → Create subnet and choose VPC MyVPC-1.

The screenshot shows the AWS VPC Subnets page. On the left, there's a navigation sidebar with options like VPC dashboard, AWS Global View, Virtual private cloud (Your VPCs, Subnets), Route tables, Internet gateways, Egress-only Internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, and Route servers. Below that is a Security section. The main area is titled "Subnets (3) Info" and contains a table with three rows of subnet information:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
-	subnet-0feb3c3994f12cf2b	Available	vpc-0387f1efb5c5fcfe	Off	172.31.32.0/2
-	subnet-04db1f5f83b1caebe	Available	vpc-0387f1efb5c5fcfe	Off	172.31.0.0/20
-	subnet-0375e18025a1586f2	Available	vpc-0387f1efb5c5fcfe	Off	172.31.16.0/2

At the bottom, there's a "Select a subnet" dropdown and a "Create subnet" button.

The screenshot shows the "Create subnet" page. At the top, it says "VPC ID" and "Create subnets in this VPC." A dropdown menu shows "vpc-059374692d8327522 (MyVPC-1)". Below that is a "Associated VPC CIDRs" section with "IPv4 CIDRs" set to "10.0.0.0/16".

The main area is titled "Subnet settings" and says "Specify the CIDR blocks and Availability Zone for the subnet." It shows "Subnet 1 of 1" and a "Subnet name" field containing "my-subnet-01".

At the bottom, there are links for CloudShell, Feedback, and Console Mobile App, along with copyright and privacy information.

2. Create public-subnet with CIDR 10.0.0.0/21.

The screenshot shows the AWS VPC Subnets creation interface. The subnet name is set to "public-subnet". The availability zone is "No preference". The IPv4 CIDR block is "10.0.0.0/16". A tag "Name" is added with the value "public-subnet".

3. Create private-subnet with CIDR 10.0.8.0/21.

The screenshot shows the AWS VPC Subnets creation interface for a private subnet. The subnet name is "private-subnet". The availability zone is "No preference". The IPv4 CIDR block is "10.0.0.0/16". A tag "Name" is added with the value "private-subnet".

4. Verify both subnets exist in the subnet list.

Step 3 — Enable public IP auto-assign for the public subnet

1. Select **public-subnet** → Actions → Edit subnet settings.

The screenshot shows the AWS VPC Subnets page. A green banner at the top indicates "You have successfully created 2 subnets: subnet-0eb4a2efc6e24c790, subnet-06b79212e2fa330b0". The "Subnets (1/5) Info" table lists five subnets:

Name	Subnet ID	State	VPC
-	subnet-0eb4a2efc6e24c790	Available	vpc-0387f1efb5c...
-	subnet-04db1f5fb3b1cae8a	Available	vpc-0387f1efb5c...
-	subnet-0375e18025a1586f2	Available	vpc-0387f1efb5c...
<input checked="" type="checkbox"/> public-subnet	subnet-0eb4a2efc6e24c790	Available	vpc-059374692dc...
<input type="checkbox"/> private-subnet	subnet-06b79212e2fa330b0	Available	vpc-059374692dc...

A context menu is open over the public-subnet row, with "Edit subnet settings" highlighted. Other options in the menu include View details, Create flow log, Edit IPv6 CIDRs, Edit network ACL association, Edit route table association, Edit CIDR reservations, Share subnet, Manage tags, and Delete subnet.

2. Enable Auto-assign public IPv4 address and save.

The screenshot shows the "Edit subnet settings" page for the subnet-0eb4a2efc6e24c790. The "Subnet" section shows the Subnet ID as subnet-0eb4a2efc6e24c790 and the Name as public-subnet.

The "Auto-assign IP settings" section contains the following configuration:

- Enable auto-assign public IPv4 address Info
- Enable auto-assign customer-owned IPv4 address Info
Option disabled because no customer owned pools found.

The "Resource-based name (RBN) settings" section contains the following configuration:

- Enable resource name DNS A record on launch Info
- Enable resource name DNS AAAA record on launch Info

The "Hostname type" section shows "Resource name" selected.

Step 4 — Create and attach Internet Gateway (my-igw) to VPC-1

1. Go to VPC → Internet gateways → Create internet gateway.

The screenshot shows the AWS VPC Internet Gateways page. On the left, there's a navigation sidebar with 'Virtual private cloud' expanded, showing 'Internet gateways' selected. The main area displays a table titled 'Internet gateways (1)'. The table has columns for Name, Internet gateway ID, State, VPC ID, and Owner. One row is listed: Name 'igw-0beea323345d7cb27', Internet gateway ID 'igw-0beea323345d7cb27', State 'Attached', VPC ID 'vpc-0387f1efb5c5f5fe', and Owner '673586849368'. A search bar at the top says 'Find internet gateways by attribute or tag'. A blue button at the top right says 'Create internet gateway'.

2. Name it **my-igw** and create.

The screenshot shows the 'Create internet gateway' wizard. The first step, 'Internet gateway settings', is displayed. It has a 'Name tag' input field containing 'my-igw'. Below it is a 'Tags - optional' section with a table showing one tag: Key 'Name' and Value 'my-igw'. A button 'Add new tag' is available. At the bottom right of the wizard is a blue 'Create internet gateway' button.

3. Attach it to MyVPC-1 (vpc-059374692d8327522).

The screenshot shows the AWS VPC Internet Gateways page. A green banner at the top indicates that an internet gateway named 'my-igw' has been created. Below this, the 'igw-06226e1fe6c49b445 / my-igw' card displays its details: Internet gateway ID (igw-06226e1fe6c49b445), State (Detached), VPC ID (-), and Owner (673586711111). It also shows a single tag named 'Name' with the value 'my-igw'. On the left sidebar, under 'Virtual private cloud', the 'Internet gateways' section is selected. At the bottom, there is a search bar and links for CloudShell, Feedback, and Console Mobile App.

The screenshot then transitions to the 'Attach to VPC' dialog. It shows the 'Available VPCs' section with a search bar containing 'vpc-059374692d8327522 - MyVPC-1'. A yellow 'Attach internet gateway' button is visible at the bottom right. The URL in the browser's address bar is [https://console.aws.amazon.com/vpc/home?region=ap-south-1#/internet-gateways/igw-06226e1fe6c49b445/attach](#).

Step 5 — Configure public route table (0.0.0.0/0 → IGW) and associate public-subnet

1. Go to **VPC → Route tables**, select the route table for VPC-1 (example shown: rtb-0c7aa5b2e1f658016).

The screenshot shows the AWS VPC Route Tables page. On the left sidebar, under 'Virtual private cloud', 'Route tables' is selected. In the main area, the title 'Route tables (1/2)' is displayed above a table. The table has columns: Name, Route table ID, Explicit subnet associations, Edge associations, Main, and VPC. Two rows are listed: 'rtb-0e00b5acbbff38ce' and 'rtb-0c7aa5b2e1f658016'. The second row is selected. Below the table, a detailed view for 'rtb-0c7aa5b2e1f658016' is shown with tabs for Details, Routes, Subnet associations, Edge associations, Route propagation, and Tags. The 'Routes' tab is selected, showing a table with columns: Destination, Target, Status, Propagated, and Route Origin. One route is listed: '10.0.0.0/16' with 'local' as the target and 'Active' status. A 'Create Route Table' button is located at the bottom right of this section.

2. Open **Routes → Edit routes** and add: **0.0.0.0/0 → Internet Gateway (igw-06226e1fe6c49b445)**, then save.

The screenshot shows the 'Edit routes' dialog box. At the top, the path 'VPC > Route tables > rtb-0c7aa5b2e1f658016 > Edit routes' is visible. The main area is titled 'Edit routes' and contains a table with columns: Destination, Target, Status, Propagated, and Route Origin. Two routes are listed:

- Destination: 10.0.0.0/16, Target: local, Status: Active, Propagated: No, Route Origin: CreateRouteTable.
- Destination: 0.0.0.0/0, Target: Internet Gateway, Status: - (pending creation), Propagated: No, Route Origin: CreateRoute.

A 'Remove' button is next to the second route. At the bottom, there are 'Add route', 'Cancel', 'Preview', and 'Save changes' buttons. The 'Save changes' button is highlighted with a yellow background.

3. Open Subnet associations → Edit subnet associations, select public-subnet, and save.

rtb-0c7aa5b2e1f658016

Details

Route table ID: rtb-0c7aa5b2e1f658016
Main: Yes
Owner ID: 67358684110

Subnet associations

No subnet associations

Subnets without explicit associations (0)

Edit subnet associations

Available subnets (1/2)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
public-subnet	subnet-0eb4a2efc6e24c790	10.0.0.0/21	-	Main (rtb-0c7aa5b2e1f658016)
private-subnet	subnet-06b79212e2fa330b0	10.0.8.0/21	-	Main (rtb-0c7aa5b2e1f658016)

Selected subnets

subnet-0eb4a2efc6e24c790 / public-subnet X

Save associations

Step 6 — Launch Proxy Server EC2 in the public subnet

1. Go to EC2 → Launch instance and set Name: Proxy Server.

The screenshot shows the AWS EC2 Instances page. On the left, a sidebar lists options like Dashboard, AWS Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager, Images, AMIs, and AMI Catalog. The main content area has a title "Amazon Elastic Compute Cloud (EC2)" with the subtitle "Create, manage, and monitor virtual servers in the cloud." It features a "Launch a virtual server" section with "Launch instance" and "View dashboard" buttons, and links for "Get started walkthroughs" and "Get started tutorial". Below this is a "Benefits and features" section with a sub-section titled "EC2 offers ultimate scalability and control". A note says "Fully resizable compute capacity to support virtually any workload. This service is best if you want:" followed by a bullet point about the highest level of control. To the right is an "Additional actions" section with "View running instances" and "Migrate a server" links. At the bottom, a blue banner says "It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices" with "Take a walkthrough" and "Do not show me this message again" buttons. The main form for launching an instance is shown, with fields for Name (set to "Proxy Server"), Application and OS Images (Amazon Machine Image) (set to "Amazon Linux 2023 AMI 2023.9.2..."), and other configuration options like Firewall (security group), Storage (volumes), and Network (subnet). The "Launch instance" button is at the bottom right. The page footer includes standard AWS links and copyright information.

2. Network settings:

- VPC: **MyVPC-1**
- Subnet: **public-subnet**
- Auto-assign public IP: **Enable**

The screenshot shows the 'Launch an instance' wizard in the AWS Management Console. The 'Network settings' section is expanded, displaying the selected VPC (MyVPC-1), subnet (public-subnet), and auto-assign public IP (Enabled). Under 'Firewall (security groups)', the 'Create security group' option is selected. The 'Summary' section on the right shows 1 instance being launched with an Amazon Linux 2023 AMI and t3.micro instance type. The 'Launch instance' button is visible at the bottom right.

3. Configure security group to allow SSH (22) (and HTTP shown enabled in screenshots).

The screenshot shows the 'Launch an instance' wizard with the 'Firewall (security groups)' section expanded. A new security group named 'launch-wizard-1' is being created. It contains three rules: 'Allow SSH traffic from Anywhere (0.0.0.0/0)', 'Allow HTTPS traffic from the internet (unchecked)', and 'Allow HTTP traffic from the internet (unchecked)'. A note below the rules states: 'Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' The 'Summary' section on the right shows 1 instance being launched with an Amazon Linux 2023 AMI and t3.micro instance type. The 'Launch instance' button is visible at the bottom right.

4. Launch and verify Proxy Server is running (example shown: i-0ef75892d2584824d).

The screenshot shows the AWS EC2 Instances "Launch an instance" page. At the top, there is a green success banner stating "Successfully initiated launch of instance (i-0ef75892d2584824d)". Below the banner, the "Launch log" section lists several steps with status icons: "Initializing requests" (Succeeded), "Creating security groups" (Succeeded), "Creating security group rules" (Succeeded), and "Launch initiation" (Succeeded). In the "Next Steps" section, there are four cards: "Create billing usage alerts" (To manage costs and avoid surprise bills, set up email notifications for billing usage thresholds.), "Connect to your instance" (Once your instance is running, log into it from your local computer.), "Connect an RDS database" (Configure the connection between an EC2 instance and a database to allow traffic flow between them.), and "Create EBS snapshot policy" (Create a policy that automates the creation, retention, and deletion of EBS snapshots.). The bottom of the page includes standard AWS navigation links like CloudShell, Feedback, and Console mobile app, along with copyright information and privacy terms.

Step 7 — Create NAT Gateway (my-nat) in the public subnet

1. Go to VPC → NAT gateways → Create NAT gateway.

The screenshot shows the AWS VPC "NAT gateways" page. On the left, the "Virtual private cloud" sidebar is expanded, showing options like Your VPCs, Subnets, Route tables, Internet gateways, Egress-only Internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, and NAT gateways (which is currently selected). The main area displays a table titled "NAT gateways" with columns for Name, NAT gateway ID, Connectivity..., State, State message, Availability ..., and Route table ID. A search bar at the top of the table says "Find NAT gateways by attribute or tag". Below the table, a message states "No NAT gateways found". At the bottom of the page, there is a modal window titled "Select a NAT gateway" which is currently empty. The bottom of the screen includes standard AWS navigation links like CloudShell, Feedback, and Console Mobile App, along with copyright information and privacy terms.

2. Set:

- Name: **my-nat**
- Subnet: **public-subnet**
- Connectivity: **Public**

The screenshot shows the 'Create NAT gateway' page in the AWS VPC console. The 'Name' field is set to 'my-nat'. The 'Availability mode' section shows 'Zonal' selected, which provides granular control within a specific availability zone. The 'Subnet' dropdown is set to 'subnet-0eb4a2efc6e24c790 (public-subnet)'. The 'Connectivity type' section shows 'Public' selected. At the bottom, there are links for CloudShell, Feedback, and Console Mobile App, along with copyright information for 2025 and links for Privacy, Terms, and Cookie preferences.

3. Allocate an Elastic IP (example shown: **13.204.219.175**) and create NAT gateway.

The screenshot shows the 'Create NAT gateway' page after an elastic IP has been allocated. A green banner at the top indicates that the IP address '13.204.219.175 (eipalloc-05c9463205e40c0cf)' has been allocated. Below this, the connectivity type is set to 'Public'. The 'Elastic IP allocation ID' dropdown is set to 'eipalloc-05c9463205e40c0cf'. The 'Allocate Elastic IP' button is visible. The 'Additional settings' section is collapsed. The 'Tags' section shows a single tag 'Name: my-nat' added. At the bottom, there are links for CloudShell, Feedback, and Console Mobile App, along with copyright information for 2025 and links for Privacy, Terms, and Cookie preferences.

4. Verify NAT gateway is created (example shown: nat-01b86df8c73d2c245 / my-nat).

The screenshot shows the AWS VPC dashboard with the path: VPC > NAT gateways > nat-01b86df8c73d2c245. A green success message at the top states: "NAT gateway nat-01b86df8c73d2c245 | my-nat was created successfully." The main card displays the following details:

NAT gateway ID	nat-01b86df8c73d2c245	Connectivity type	Public	State	Pending	State message	Info
NAT gateway ARN	arn:aws:ec2:ap-south-1:673586848208:natgateway/nat-01b86df8c73d2c245	Primary public IPv4 address	-	Primary private IPv4 address	10.0.0.237	Primary network interface ID	eni-01b9bd0a73b2892b0
VPC	vpc-059374692d8327522 / MyVPC-1	Subnet	subnet-0eb4a2efc6e24c790 / public-subnet	Created	Sunday, 14 December 2025 at 11:11:43 GMT+5:30	Deleted	-

Below the main card, there are tabs for "Secondary IPv4 addresses", "Monitoring", and "Tags". The "Secondary IPv4 addresses" tab shows a table with columns: Private IPv4 address, Network interface ID, and Status.

Step 8 — Launch Application Server EC2 in the private subnet

1. Go to EC2 → Launch instance and set Name: Application Server.

The screenshot shows the AWS EC2 Instances launch wizard with the path: EC2 > Instances > Launch an instance. A blue banner at the top says: "It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices". Buttons for "Take a walkthrough" and "Do not show me this message again." are present.

The main form is titled "Launch an instance" and includes the following fields:

- Name and tags**: Name is set to "Application Server".
- Application and OS Images (Amazon Machine Image)**: Search bar placeholder: "Search our full catalog including 1000s of application and OS images".
- Summary** section:
 - Number of instances: 1
 - Software Image (AMI): Amazon Linux 2023 AMI 2023.9.2... [read more](#)
 - Virtual server type (instance type): t3.micro
 - Firewall (security group): New security group
 - Storage (volumes): 1 volume(s) - 8 GiB

At the bottom, there are "Cancel" and "Launch instance" buttons.

2. Network settings:

- VPC: **MyVPC-1**
- Subnet: **private-subnet**
- Auto-assign public IP: **Disable**

The screenshot shows the 'Launch an instance' wizard in the AWS Management Console. In the 'Network settings' section, the VPC is set to 'MyVPC-1' and the subnet is 'private-subnet'. The 'Auto-assign public IP' option is set to 'Disable'. On the right side, the 'Summary' panel shows 1 instance being launched with the AMI 'Amazon Linux 2023 AMI 2023.9.2...'. Other details include the virtual server type 't3.micro', a security group 'launch-wizard-1', and storage of 1 volume(s) - 8 GiB.

3. Select existing security group **launch-wizard-1 (sg-0cdcb2091978b33e1)**.

The screenshot shows the 'Launch an instance' wizard. In the 'Firewall (security groups)' section, the 'Select existing security group' option is selected, and the security group 'launch-wizard-1 (sg-0cdcb2091978b33e1)' is chosen. The 'Common security groups' dropdown also lists 'launch-wizard-1'. The 'Summary' panel on the right remains the same as in the previous step, showing 1 instance being launched with the specified configuration.

4. Verify Application Server is running and has private IP 10.0.14.230.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like Dashboard, AWS Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, and Capacity Manager. The main area displays a table titled 'Instances (1/2) Info' with two rows. The first row is for a 'Proxy Server' with the ID i-0ef75892d2584824d, which is 'Running' and has a status check of '3/3 checks passed'. The second row is for an 'Application Se...' with the ID i-09a754b97868a83ac, which is also 'Running' but is currently 'Initializing'. Below the table, a specific instance, 'i-09a754b97868a83ac (Application Server)', is selected. Its details are shown in a modal: it has a Public IPv4 address (empty) and a Private IPv4 address of 10.0.14.230. The modal also includes tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags.

Step 9 — Create a private subnet route table (nat-rt) and route private traffic to NAT

1. Go to VPC → Route tables → Create route table.
2. Name it **nat-rt** and select VPC **MyVPC-1**.

The screenshot shows the 'Create route table' wizard. In the first step, 'Route table settings', the user has entered the name 'nat-rt' and selected the VPC 'MyVPC-1'. In the second step, 'Tags', a single tag is added with the key 'Name' and value 'nat-rt'. There is a note indicating that up to 49 more tags can be added. At the bottom right, there are 'Cancel' and 'Create route table' buttons.

3. Verify route table created successfully (example shown: rtb-0a4757d6fb8ec718d / nat-rt).

The screenshot shows the AWS VPC Route Tables page. A green success message at the top states: "Route table rtb-0a4757d6fb8ec718d | nat-rt was created successfully." Below this, the route table details are displayed:

- Route table ID:** rtb-0a4757d6fb8ec718d
- Main:** No
- Owner ID:** vpc-059374692d8327522 | MyVPC-1
- Explicit subnet associations:** -
- Edge associations:** -

The "Routes" tab is selected, showing one route entry:

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	Active	No	Create Route Table

4. Associate private-subnet with nat-rt (explicit association should show subnet-06b79212e2fa330b0 / private-subnet).

The screenshot shows the AWS VPC Route Tables page. A green success message at the top states: "Updated routes for rtb-0a4757d6fb8ec718d / nat-rt successfully". Below this, the route table details are displayed:

- Route table ID:** rtb-0a4757d6fb8ec718d
- Main:** No
- Owner ID:** vpc-059374692d8327522 | MyVPC-1
- Explicit subnet associations:** -
- Edge associations:** -

The "Subnet associations" tab is selected, showing the "Explicit subnet associations (0)" section. A button labeled "Edit subnet associations" is visible.

The "Edit subnet associations" page shows the available subnets:

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
public-subnet	subnet-0eb4a2efc6e24c790	10.0.0.0/21	-	rtb-0c7aa5b2e1f658016
<input checked="" type="checkbox"/> private-subnet	subnet-06b79212e2fa330b0	10.0.8.0/21	-	Main (rtb-0c7aa5b2e1f658016)

The "Selected subnets" section contains the entry: "subnet-06b79212e2fa330b0 / private-subnet". Buttons for "Cancel" and "Save associations" are at the bottom.

5. In **nat-rt** → **Routes** → **Edit routes**, add:

- **0.0.0.0/0** → **NAT Gateway (nat-01b86df8c73d2c245)**, then save.

The screenshot shows the AWS VPC Route Tables interface. A new route is being added to the 'rtb-0a4757d6fb8ec718d' route table. The destination is '0.0.0.0/0' and the target is 'NAT Gateway'. The status is 'Active' and propagation is set to 'No'. The route origin is 'CreateRouteTable'. There is a 'Remove' button and an 'Add route' button. At the bottom right are 'Cancel', 'Preview', and 'Save changes' buttons. The top navigation bar shows 'Account ID: 6735-868...', 'Region: Asia Pacific (Mumbai)', and various AWS services like IAM, S3, EC2, Simple Notification Service, VPC, Aurora and RDS, Route 53, and Certificate Manager.

Step 10 — Validate private subnet outbound connectivity

1. Use SSH from the Proxy environment to connect to the Application Server private IP **10.0.14.230** using sample-2-ind.pem.

```
RostedNik@RostedNik MINGW64 ~/Downloads/AWS
$ scp -i sample-2-ind.pem sample-2-ind.pem ec2-user@3.110.131.57:/home/ec2-user
The authenticity of host '3.110.131.57 (3.110.131.57)' can't be established.
ED25519 key fingerprint is SHA256:xD5uzbE9fIypk4gGdwawPYO4pTEXR4h1JxRvmgrQ3hA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.110.131.57' (ED25519) to the list of known hosts.
sample-2-ind.pem                                         100% 1678    167.9KB/s   00:00
```

The screenshot shows the AWS CloudShell terminal. It displays the command 'ssh -i "sample-2-ind.pem" ec2-user@10.0.14.230' and its execution. The output shows the user is connected to an Amazon Linux 2023 instance with the URL 'https://aws.amazon.com/linux/amazon-linux-2023'. The terminal prompt is '[ec2-user@ip-10-0-5-34 ~]\$'. At the bottom, it shows the session details: 'i-0ef75892d2584824d (Proxy Server)', 'PublicIPs: 3.110.131.57 PrivateIPs: 10.0.5.34', and the AWS footer with 'CloudShell Feedback Console Mobile App' and copyright information.

2. From the Application Server session, run ping 8.8.8.8 and verify responses are received.

The screenshot shows the AWS CloudShell interface. At the top, there's a navigation bar with tabs like IAM, S3, EC2, Simple Notification Service, VPC, Aurora and RDS, Route 53, and Certificate Manager. The account ID is listed as 6735-8684-0741. Below the navigation bar is a terminal window displaying the following text:

```
[ec2-user@ip-10-0-5-34 ~]$ ssh -i "sample-2-ind.pem" ec2-user@10.0.14.230
,
~\ #####
~~ \####\
~~ \###|
~~ \|/ V~'-'>
~~ ./
~/m/'

[ec2-user@ip-10-0-14-230 ~]$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=7 ttl=114 time=1.34 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=114 time=1.25 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=114 time=1.22 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=114 time=1.32 ms
```

Below the terminal, there's a message box titled "i-0ef75892d2584824d (Proxy Server)" with the text "Public IPs: 3.110.131.57 Private IPs: 10.0.5.34". At the bottom of the screen, there are links for "CloudShell", "Feedback", and "Console Mobile App". The footer of the page includes copyright information: "© 2025, Amazon Web Services, Inc. or its affiliates.", "Privacy", "Terms", and "Cookie preferences".

Step 11 — Create VPC-2 (MyVPC-2) and its private subnet

1. Go to **VPC** → **Your VPCs** → **Create VPC** and create:

- Name: **MyVPC-2**
- IPv4 CIDR: **11.0.0.0/16**

The screenshot shows the "Create VPC" page in the AWS VPC service. The top navigation bar includes tabs for IAM, S3, EC2, Simple Notification Service, VPC, Aurora and RDS, Route 53, and Certificate Manager. The account ID is 6735-8684-0741. The region is set to Asia Pacific (Mumbai). The URL shows the path: 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.

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 - optional
Creates a tag with a key of 'Name' and a value that you specify.
MyVPC-2

IPv4 CIDR block Info
 IPv4 CIDR manual input IPAM-allocated IPv4 CIDR block
11.0.0.0/16

CIDR block size must be between /16 and /28.

IPv6 CIDR block Info
 No IPv6 CIDR block IPAM-allocated IPv6 CIDR block

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

2. Go to VPC → Subnets → Create subnet and create:

- Name: **private-subnet-2**
- CIDR: **11.0.0.0/21**

The screenshot shows the 'Create subnet' wizard in the AWS VPC console. The steps completed are:

- Subnet name:** private-subnet-2
- Availability Zone:** No preference
- IPv4 VPC CIDR block:** 11.0.0.0/16
- IPv4 subnet CIDR block:** 11.0.0.0/21 (selected from a dropdown menu)
- Tags - optional:** A single tag 'Name: private-subnet-2' is added.

3. Verify subnet is created (example shown: subnet-02b3afb42fae45b4e).

The screenshot shows the 'Subnets' dashboard in the AWS VPC console. A success message at the top indicates 'You have successfully created 1 subnet: subnet-02b3afb42fae45b4e'. The main table lists the following subnets:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
-	subnet-0feb3c3994f12cf2b	Available	vpc-0387f1efb5c5f5fce	Off	172.31.32.0/2
-	subnet-04db1f5f83b1caebe	Available	vpc-0387f1efb5c5f5fce	Off	172.31.0.0/20
-	subnet-0375e18025a1586f2	Available	vpc-0387f1efb5c5f5fce	Off	172.31.16.0/2
public-subnet	subnet-0eb4a2efc6e24c790	Available	vpc-059374692d8327522 My...	Off	10.0.0.0/21
private-subnet	subnet-06b79212e2fa330b0	Available	vpc-059374692d8327522 My...	Off	10.0.8.0/21
private-subnet-2	subnet-02b3afb42fae45b4e	Available	vpc-03f91d83688eaec8a My...	Off	11.0.0.0/21

Step 12 — Launch Database Server EC2 in VPC-2 private subnet

1. Go to EC2 → Launch instance and set Name: Database Server.

The screenshot shows the 'Launch an instance' wizard in the AWS Management Console. The 'Name and tags' step is active, where the instance name is set to 'Database Server'. The 'Software Image (AMI)' section shows 'Amazon Linux 2023 AMI 2023.9.2...' selected. The 'Virtual server type (instance type)' is set to 't3.micro'. Under 'Storage (volumes)', there is one volume of 8 GiB. The 'Summary' box indicates 1 instance. The 'Launch Instance' button is prominently displayed at the bottom right.

2. Network settings:

- VPC: **MyVPC-2 (vpc-03f91d83688eaec8a)**
- Subnet: **private-subnet-2**
- Auto-assign public IP: **Disable**

The screenshot shows the 'Launch an instance' wizard with the 'Network settings' step selected. The 'Key pair (login)' section shows 'sample-2-ind' selected. In the 'Network settings' section, the 'Subnet' dropdown is set to 'private-subnet-2'. The 'Auto-assign public IP' dropdown is set to 'Disable'. The 'Summary' box indicates 1 instance. The 'Launch Instance' button is at the bottom right.

3. Create a new security group **launch-wizard-2** (SSH inbound shown in screenshot).

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

Security group name - required
launch-wizard-2

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-./[!#:@]+=&|!\$*

Description - required | Info
launch-wizard-2 created 2025-12-14T05:46:17.902Z

Inbound Security Group Rules

- Security group rule 1 (TCP, 22, 0.0.0.0/0)**

Type Info	Protocol Info	Port range Info
ssh	TCP	22

Source type | Info
Anywhere

Source | Info
Add CIDR, prefix list or security group
e.g. SSH for admin desktop
- Security group rule 2 (TCP, 80, 0.0.0.0/0)**

Type Info	Protocol Info	Port range Info
HTTP	TCP	80

Source type | Info
Anywhere

Source | Info
Add CIDR, prefix list or security group
e.g. SSH for admin desktop

Summary

Number of instances | Info
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.9.2...read more
ami-00ca570c1b6d79f36

Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Cancel **Launch instance** **Preview code**

4. Verify Database Server is running and has private IP **11.0.1.238** (example shown).

Instances (1/3) | Info
Last updated less than a minute ago

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
Proxy Server	i-0ef75892d2584824d	Running	t3.micro	3/3 checks passed	View alarms +	ap-south-1a	-
Application Se...	i-09a754b97868a83ac	Running	t3.micro	3/3 checks passed	View alarms +	ap-south-1a	-
Database Server	i-094478730818ee793	Running	t3.micro	Initializing	View alarms +	ap-south-1c	-

i-094478730818ee793 (Database Server)

Details **Status and alarms** **Monitoring** **Security** **Networking** **Storage** **Tags**

Instance summary | Info

Instance ID i-094478730818ee793	Public IPv4 address -	Private IPv4 addresses 11.0.1.238
---	---------------------------------	---

Step 13 — Create and accept VPC Peering (my-**pc**)

1. Go to **VPC** → **Peering connections** → **Create peering connection**.

2. Set:

- Name: **my-**pc****
- Requester: **MyVPC-1 (vpc-059374692d8327522)**
- Acceptor: **MyVPC-2 (vpc-03f91d83688eaec8a)**

The screenshot shows the 'Create peering connection' page in the AWS VPC console. The 'Peering connection settings' section is filled out with the following details:

- Name**: my-**pc**
- VPC ID (Requester)**: vpc-059374692d8327522 (MyVPC-1)
- VPC CIDRs for vpc-059374692d8327522 (MyVPC-1)**: 10.0.0.0/16 (Status: Associated)
- Select another VPC to peer with**:
 - Account**: My account
 - Region**: This Region (ap-south-1)
 - VPC ID (Acceptor)**: vpc-03f91d83688eaec8a (MyVPC-2)
 - VPC CIDRs for vpc-03f91d83688eaec8a (MyVPC-2)**: 11.0.0.0/16 (Status: Associated)
- Tags**: A tag named 'Name' with value 'my-**pc**' is added.

At the bottom right, there are 'Cancel' and 'Create peering connection' buttons.

3. Open the created peering connection (example shown: pcx-02c4425d1ef728949) and select **Actions** → **Accept request**.

The screenshot shows the AWS VPC Peering connections page. A specific peering connection, "pcx-02c4425d1ef728949 / my-pc", is selected. The status is "Pending acceptance". On the right, a context menu is open with options: "Accept request", "Reject request", "Edit DNS settings", "Manage tags", and "Delete peering connection". Below the main content, there are tabs for "DNS", "Route tables", and "Tags", with "DNS" currently selected. A "DNS settings" section is visible.

A modal dialog box titled "Accept VPC peering connection request" is displayed. It contains fields for Requester VPC (vpc-059374692d8327522 / MyVPC-1), Acceptor VPC (vpc-03f91d83688eaec8a / MyVPC-2), Requester CIDRs (10.0.0.0/16), Acceptor CIDRs (empty), Requester Region (Mumbai (ap-south-1)), and Acceptor Region (Mumbai (ap-south-1)). At the bottom are "Cancel" and "Accept request" buttons.

4. Acceptance is required for the peering connection to become active.

The screenshot shows the AWS VPC Peering connections page again. The peering connection status has changed to "Established". A green message box states: "Your VPC peering connection (pcx-02c4425d1ef728949 | my-pc) 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." There is a "Modify my route tables now" button. The "Actions" dropdown is visible at the top right.

Step 14 - finalize end-to-end App → DB access

- To complete private App → DB communication, add route table entries on both sides so:
 - VPC-1 routes **11.0.0.0/16** → pcx-...
 - VPC-2 routes **10.0.0.0/16** → pcx-...

The screenshots illustrate the process of configuring route tables in two VPCs to enable communication between the Application and Database subnets.

Screenshot 1: Route Tables (1/4) - Details View

This screenshot shows the list of route tables in the VPC. It includes columns for Route table ID, Explicit subnet associations, Edge associations, Main, and VPC. The route table **rtb-05b838d7c5754b9b3** is selected.

Route table ID	Explicit subnet associations	Edge associations	Main	VPC
rtb-0e00b5acbbfff38ce	-	-	Yes	vpc-0387f1efb5c5fcfe
rtb-05b838d7c5754b9b3	-	-	Yes	vpc-03f91d83689eaecc8a MyVPC-2
rtb-0c7aa52e1f658016	subnet-0eb4a2efc6e24c7...	-	Yes	vpc-059374692d8327522 MyVPC-1
rtb-0a4757d6fb8ec718d	subnet-06b79212e2fa33...	-	No	vpc-059374692d8327522 MyVPC-1

Screenshot 2: Edit routes - Route Table Configuration

This screenshot shows the configuration of routes for the selected route table **rtb-05b838d7c5754b9b3**. A route entry for **11.0.0.0/16** is defined with a target of **local**.

Destination	Target	Status	Propagated	Route Origin
11.0.0.0/16	local	Active	No	CreateRouteTable

Screenshot 3: Edit subnet associations - Route Table Configuration

This screenshot shows the configuration of subnet associations for the selected route table **rtb-05b838d7c5754b9b3**. It lists the available subnets and the selected subnet **private-subnet-2**.

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
private-subnet-2	subnet-02b3afb42fae45b4e	11.0.0.0/21	-	Main (rtb-05b838d7c5754b9b3)

- After routes are added, validate connectivity from **Application Server** to **Database Server** private IP (11.0.1.238) using the required DB port or a network test.

aws | Search [Alt+S] | IAM S3 EC2 Simple Notification Service VPC Aurora and RDS Route 53 Certificate Manager | Account ID: 6735-8684-... | Asia Pacific (Mumbai) | Logout

```
64 bytes from 8.8.8.8: icmp_seq=10 ttl=114 time=1.32 ms
64 bytes from 8.8.8.8: icmp_seq=11 ttl=114 time=1.31 ms
64 bytes from 8.8.8.8: icmp_seq=12 ttl=114 time=1.25 ms
^C
-- 8.8.8.8 ping statistics --
12 packets transmitted, 6 received, 50% packet loss, time 11243ms
rtt min/avg/max/mdev = 1.217/1.281/1.339/0.045 ms
[ec2-user@ip-10-0-14-230 ~]$ exit
logout
Connection to 10.0.14.230 closed.
[ec2-user@ip-10-0-5-34 ~]$ scp -i sample-2-ind.pem sample-2-ind.pem ec2-user@10.0.14.230:/home/ec2-user
sample-2-ind.pem
[ec2-user@ip-10-0-5-34 ~]$ ssh -i "sample-2-ind.pem" ec2-user@10.0.14.230
100% 1678 4.5MB/s 00:00
' #'
~\ #####
~~ \#####
~~ \###/
~~ \#/ https://aws.amazon.com/linux/amazon-linux-2023
~~ \~-' '->
~~ .-
~~ /-
~/m/'

Last login: Sun Dec 14 05:41:03 2025 from 10.0.5.34
[ec2-user@ip-10-0-14-230 ~]$ ls
sample-2-ind.pem
[ec2-user@ip-10-0-14-230 ~]$ i-0ef75892d2584824d (Proxy Server)
PublicIPs: 3.110.131.57 PrivateIPs: 10.0.5.34
CloudShell Feedback Console Mobile App | © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences | Account ID: 6735-8684-... | Asia Pacific (Mumbai) | Logout
```

aws | Search [Alt+S] | IAM S3 EC2 Simple Notification Service VPC Aurora and RDS Route 53 Certificate Manager | Account ID: 6735-8684-... | Asia Pacific (Mumbai) | Logout

EC2 > Instances

Instances (1/3) Info								
Last updated 6 minutes ago								
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
<input type="checkbox"/>	Proxy Server	i-0ef75892d2584824d	Running	t3.micro	3/3 checks passed	View alarms +	ap-south-1a	-
<input type="checkbox"/>	Application Se...	i-09a754b97868a83ac	Running	t3.micro	3/3 checks passed	View alarms +	ap-south-1a	-
<input checked="" type="checkbox"/>	Database Server	i-094478730818ee793	Running	t3.micro	Initializing	View alarms +	ap-south-1c	-

i-094478730818ee793 (Database Server)

[Details](#) [Status and alarms](#) [Monitoring](#) [Security](#) [Networking](#) [Storage](#) [Tags](#)

Instance summary [Info](#)

Instance ID	i-094478730818ee793	Public IPv4 address
		11.0.1.238

Private IPv4 address copied

CloudShell Feedback Console mobile app | © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences | Account ID: 6735-8684-... | Asia Pacific (Mumbai) | Logout

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

Last login: Sun Dec 14 05:41:03 2025 from 10.0.5.34
[ec2-user@ip-10-0-14-230 ~]$ ls
sample-2-ind.pem
[ec2-user@ip-10-0-14-230 ~]$ ssh -i "sample-2-ind.pem" ec2-user@11.0.1.238
The authenticity of host '11.0.1.238 (11.0.1.238)' can't be established.
ED25519 key fingerprint is SHA256:UC0H4Y8jekg8a6TPchHzAv5y4cZ/zvFvYOCzGhzLsd0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '11.0.1.238' (ED25519) to the list of known hosts.
' #'
~\ #####
~~ \#####
~~ \###/
~~ \#/ https://aws.amazon.com/linux/amazon-linux-2023
~~ \~-' '->
~~ .-
~~ /-
~/m'

[ec2-user@ip-11-0-1-238 ~]$ i-0ef75892d2584824d (Proxy Server)
PublicIPs: 3.110.131.57 PrivateIPs: 10.0.5.34
CloudShell Feedback Console Mobile App | © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences | Account ID: 6735-8684-... | Asia Pacific (Mumbai) | Logout
```