

Use case

Secure VPC Architecture

Use case Description

Design a multi-tier VPC with private/public subnets and NAT Gateways.

Approach :

1. Create a VPC
2. Create 2 Public Subnets + 2 Private Subnets (in two AZs)
3. Create and attach an Internet Gateway
4. Create Public Route Table - Route to Internet Gateway
5. Create Private Route Table - Route to NAT Gateway
6. Create a NAT Gateway in Public Subnet + Allocate EIP
7. Launch EC2 in Public Subnet
8. Launch EC2 in Private Subnet
9. Configure Security Groups
10. Test: SSH into Public EC2, then SSH into Private EC2 via Public one

Create VPC

The screenshot displays the AWS Management Console interface for a VPC named 'SecureVPC' (VPC ID: vpc-0585872f8271112f6). The console is in the 'Asia Pacific (Mumbai)' region, owned by 'aws-dimshikarous (7704-2478-7712)'. The left sidebar shows the 'VPC dashboard' with a search bar and a filter by VPC. The main content area is divided into several sections:

- Details:** Shows VPC ID, State (Available), DNS resolution (Enabled), Main network ACL (acl-05d0a551c7983c70f), IPv6 CIDR (Network border group), Tenancy (default), Default VPC (No), Network Address Usage metrics (Disabled), Block Public Access (Off), DHCP option set (dhcp-0c252ba8a238db05), IP v4 CIDR (10.0.0.0/16), Route 53 Resolver DNS Firewall rule groups (None), DNS hostnames (Disabled), Main route table (rtb-05168822994a18c09), IPv6 pool (None), and Owner ID (770424767712).
- Resource map:** Provides a visual overview of the VPC resources and their connections.
 - VPC:** Shows 'SecureVPC' as the VPC.
 - Subnets (4):** Lists four subnets: 'PublicSubnet1', 'PrivateSubnet2', 'PrivateSubnet1', and 'PublicSubnet2', grouped under two availability zones: 'ap-south-1a' and 'ap-south-1b'.
 - Route tables (3):** Shows three route tables: 'Public-RT', 'Private-RT', and 'rtb-05168822994a18c09'.
 - Network connections (2):** Shows two connections: 'SecureVPC-IGW' and 'SecureVPC-NAT'.

Create Subnets

The screenshot shows the AWS VPC console's Subnets page. The left sidebar contains navigation links for VPC dashboard, EC2 Global View, and Virtual private cloud. The main content area shows a table of subnets. The subnets listed are:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR	IPv6 CIDR association ID	Available IPv4 addresses
-	subnet-0c74fcd414a955635	Available	vpc-0b408ac844bffa015	Off	172.31.16.0/20	-	-	4091
-	subnet-04d2d7500060a2f59	Available	vpc-0b408ac844bffa015	Off	172.31.32.0/20	-	-	4091
PublicSubnet1	subnet-0c8fc33fc23263f8	Available	vpc-0585872f8271112f6 Secu...	Off	10.0.1.0/24	-	-	249
-	subnet-0b8e0751da03a969	Available	vpc-0b408ac844bffa015	Off	172.31.0.0/20	-	-	4088
MainSubnet	subnet-0ed81de80a6755a8	Available	vpc-0209a795591d0791d Maj...	Off	10.0.1.0/24	-	-	250
PublicSubnet2	subnet-030f0522847c33340	Available	vpc-0585872f8271112f6 Secu...	Off	10.0.2.0/24	-	-	251
PrivateSubnet2	subnet-0ab2a5d87c2d6d349	Available	vpc-0585872f8271112f6 Secu...	Off	10.0.4.0/24	-	-	251
PrivateSubnet1	subnet-0c6da763ed187d115	Available	vpc-0585872f8271112f6 Secu...	Off	10.0.3.0/24	-	-	250

Create Internet Gateway

The screenshot shows the AWS VPC console's Internet gateways page. The left sidebar contains navigation links for VPC dashboard, EC2 Global View, and Virtual private cloud. The main content area shows the details for the Internet gateway igw-05dd29f91b5d8b791. The details include:

- Internet gateway ID: igw-05dd29f91b5d8b791
- State: Attached
- VPC ID: vpc-0585872f8271112f6 | SecureVPC
- Owner: 770424767712

Attach IGW to the public subnet

The screenshot shows the AWS VPC console's Edit route table association page. The left sidebar contains navigation links for VPC dashboard, EC2 Global View, and Virtual private cloud. The main content area shows the details for the route table association. The details include:

- Subnet route table settings: Subnet ID: subnet-0c8fc33fc23263f8, Route table ID: rtb-0aed42d7dfc2d1ad: (Public-RT)
- Routes (2):

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-05dd29f91b5d8b791

Create elastic IP

The screenshot shows the AWS EC2 console's Elastic IP addresses page. The left sidebar contains navigation links for EC2 dashboard, Events, Instances, Images, Elastic Block Store, and Network & Security. The main content area shows the details for the Elastic IP address 3.6.84.171. The details include:

- Summary:
 - Allocated IPv4 address: 3.6.84.171
 - Association ID: eipassoc-063d7ce056eb6387b
 - Network interface ID: eni-0a395414ecf23b4c
 - Address pool: Amazon
- Type: Public IP
- Scope: VPC
- Network interface owner account ID: 770424767712
- Network border group: ap-south-1
- Allocation ID: eipalloc-0ea3e7d8ca8ea2bf
- Associated instance ID: -
- Public DNS: -
- Reverse DNS record: -
- Private IP address: 10.0.1.163
- NAT Gateway ID: nat-09b77a2086e11a9df | SecureVPC-NAT1

Create NAT Gateway

The screenshot shows the AWS Management Console for a NAT Gateway. The left sidebar contains navigation links for VPC, Subnets, Route tables, Internet gateways, Egress-only internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, Security, and Network ACLs. The main content area displays the details for the NAT Gateway 'nat-09b72d3b96e11a0af / SecureVPC-NAT'. The 'Details' section shows the NAT gateway ID, NAT gateway ARN, VPC, Connectivity type (Public), Primary public IPv4 address (5.6.94.171), Subnet (subnet-0c3dc33fc23263f8 / PublicSubnet1), State (Available), Primary private IPv4 address (10.0.1.163), Created time (Tuesday, July 15, 2025 at 18:52:33 GMT+5:30), and State message. Below the details, there is a section for 'Secondary IPv4 addresses' with a search bar and a table with columns: Private IPv4 address, Allocation ID, Association ID, Public IPv4 address, Network interface ID, and Status. A message states: 'Secondary IPv4 addresses are not available for this nat gateway.'

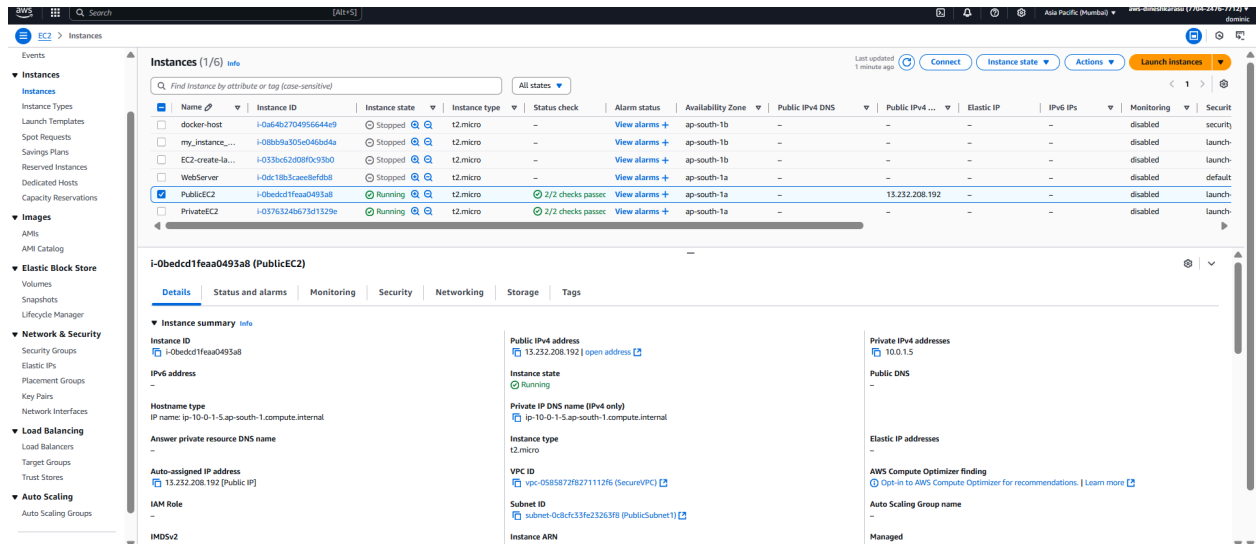
Attach NAT GW to the private subnet

The screenshot shows the 'Edit route table association' page in the AWS Management Console. The 'Subnet route table settings' section shows the Subnet ID (subnet-0c3da763ed187d115) and the selected Route table ID (rtb-0206b3b75817ea475 (Private-RT)). Below this, the 'Routes (2)' section shows a table with columns: Destination, Target, and a link to view the route. The table contains two entries: one for destination 10.0.0.0/16 with target 'local', and another for destination 0.0.0.0/0 with target 'nat-09b72d3b96e11a0af'. At the bottom right, there are 'Cancel' and 'Save' buttons.

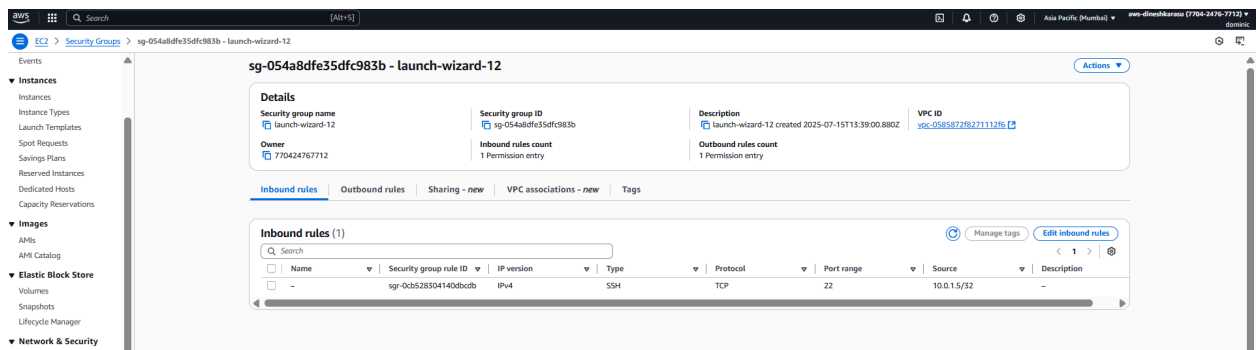
Create a private instance in the VPC

The screenshot shows the AWS Management Console for EC2 Instances. The left sidebar contains navigation links for Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, Load Balancers, Target Groups, Trust Stores, Auto Scaling, and Auto Scaling Groups. The main content area displays the 'Instances (1/6)' page with a table of instances. The instance 'PrivateEC2' (i-0376324b673d1329e) is highlighted. Below the table, the 'Details' section for 'i-0376324b673d1329e (PrivateEC2)' is shown. The 'Instance summary' section includes the Instance ID, IP name, Hostname type, Answer private resource DNS name, Auto-assigned IP address, IAM Role, and IMDSv2. The 'Private IPv4 address' section shows the address 10.0.3.201. The 'Public DNS' section shows the address 10.0.3.201. The 'Elastic IP addresses' section shows the address 15.232.208.192. The 'AWS Compute Optimizer Finding' section shows the finding ID and a link to learn more. The 'Auto Scaling Group name' section shows the name 'Managed-fake'.

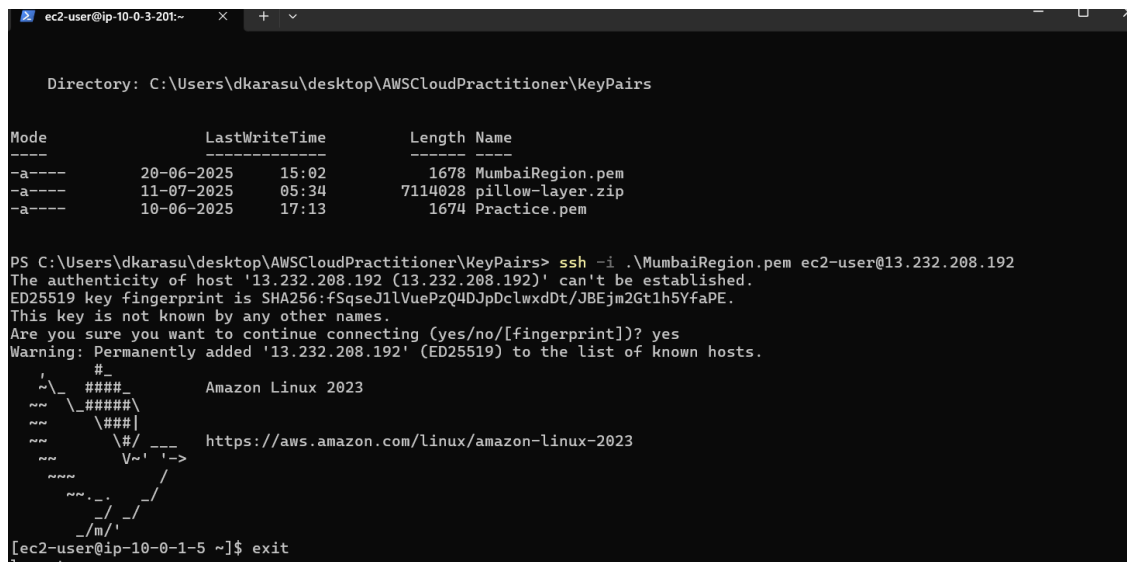
Create a public EC2 instance in the VPC



Modify security group inbound rules of Private EC2 instance to allow only the newly created Public instance PrivateIP4 address



Test SSH into the public instance




Secure copy the key pair file to the EC2 instance

```
ec2-user@ip-10-0-0-3-201:~$ exit  
logout  
Connection to 13.232.208.192 closed.  
PS C:\Users\dkarasu\Desktop\AWScloudPractitioner\KeyPairs> scp -i MumbaiRegion.pem MumbaiRegion.pem ec2-user@13.232.208.192:/home/ec2-user/  
MumbaiRegion.pem 100% 1678    63.0KB/s   00:00  
PS C:\Users\dkarasu\Desktop\AWScloudPractitioner\KeyPairs> ssh -i MumbaiRegion.pem ec2-user@13.232.208.192  
  
#  
#####  
_ _ _ _ _ Amazon Linux 2023  
_ _ _ _ _ #####  
_ _ _ _ _ \####  
_ _ _ _ _ \|  
_ _ _ _ _ \#/ https://aws.amazon.com/linux/amazon-linux-2023  
_ _ _ _ _ V ~ ^ ! ->  
  
_ _ _ _ _  
_ _ _ _ _  
_ _ _ _ _  
_ _ _ _ _  
_ _ _ _ _  
_ _ _ _ _  
_ _ _ _ _
```

Last login: Tue Jul 15 13:48:54 2025 from 103.183.203.20
[ec2-user@ip-10-0-1-5 ~]\$ ls
MumbaiRegion.pem

SSH into the private instance from the already logged in public instance

```
[ec2-user@ip-10-0-1-5 ~]$ ssh -i MumbaiRegion.pem ec2-user@10.0.3.201
The authenticity of host '10.0.3.201 (10.0.3.201)' can't be established.
ED25519 key fingerprint is SHA256:6Tohoe4kRc04cWiv835qdekqvUeUrIcMzwLudBSXoY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.3.201' (ED25519) to the list of known hosts.
```



```
#
Amazon Linux 2023

https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-10-0-3-201 ~]$ curl https://google.com
<HTML><HEAD><meta http-equiv="content-type" content="text/html;charset=utf-8">
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="https://www.google.com/">here</A>.
</BODY></HTML>
[ec2-user@ip-10-0-3-201 ~]$ sudo yum update --y
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved
```

159 kB/s | 17 kB 00:00

Try Logging into the Private instance from the CLI, we get timeout error

```
PS C:\Users\dkarasu\desktop\AWScloudPractitioner\KeyPairs> ssh -i MumbaiRegion.pem ec2-user@10.0.1.5
ssh: connect to host 10.0.1.5 port 22: Connection timed out
PS C:\Users\dkarasu\desktop\AWScloudPractitioner\KeyPairs> |
```