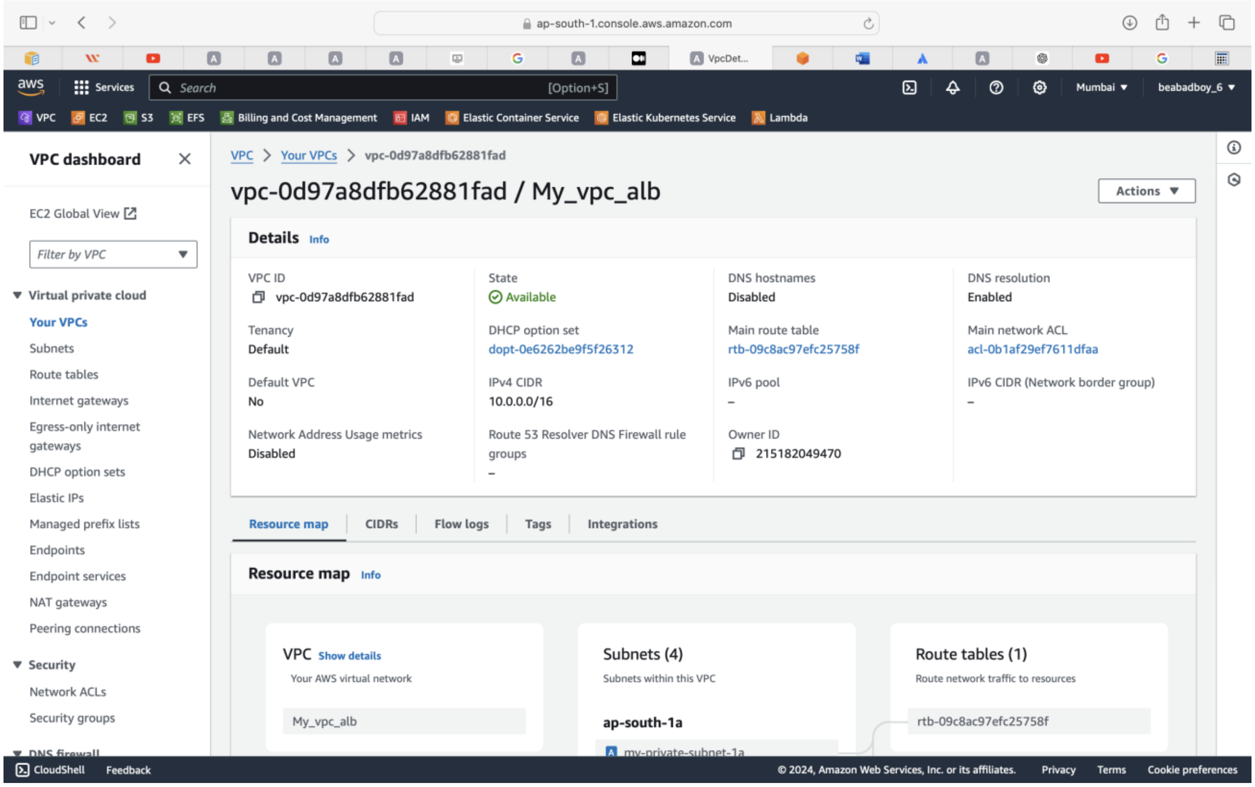
Name – Parth

Position - Intern

Task –1 & Project –1 [Document]

- Create a Vpc (Virtual Private Cloud), and create 2 public subnets and 2 private subnets, connect public subnets with internet gateway and private subnets with Nat gateway. So, that a network can be established, after that create a ec2 instance in public subnet and try to access that instance in cli. Also, create a ec2 instance in private subnet and also try to access that in cli. Note: Private instance can't be accessed directly (without Vpn). The other alternative way is to access with the help of instance present in public subnet.

Step 1: Creating a vpc and involving all the dependecies needed.



vpc-0d97a8dfb62881fad / My_vpc_alb

Details Info

VPC ID vpc-0d97a8dfb62881fad	State Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-0e6262be9f5f26312	Main route table rtb-09c8ac97efc25758f	Main network ACL acl-0b1af29ef7611dfaa
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 215182049470	

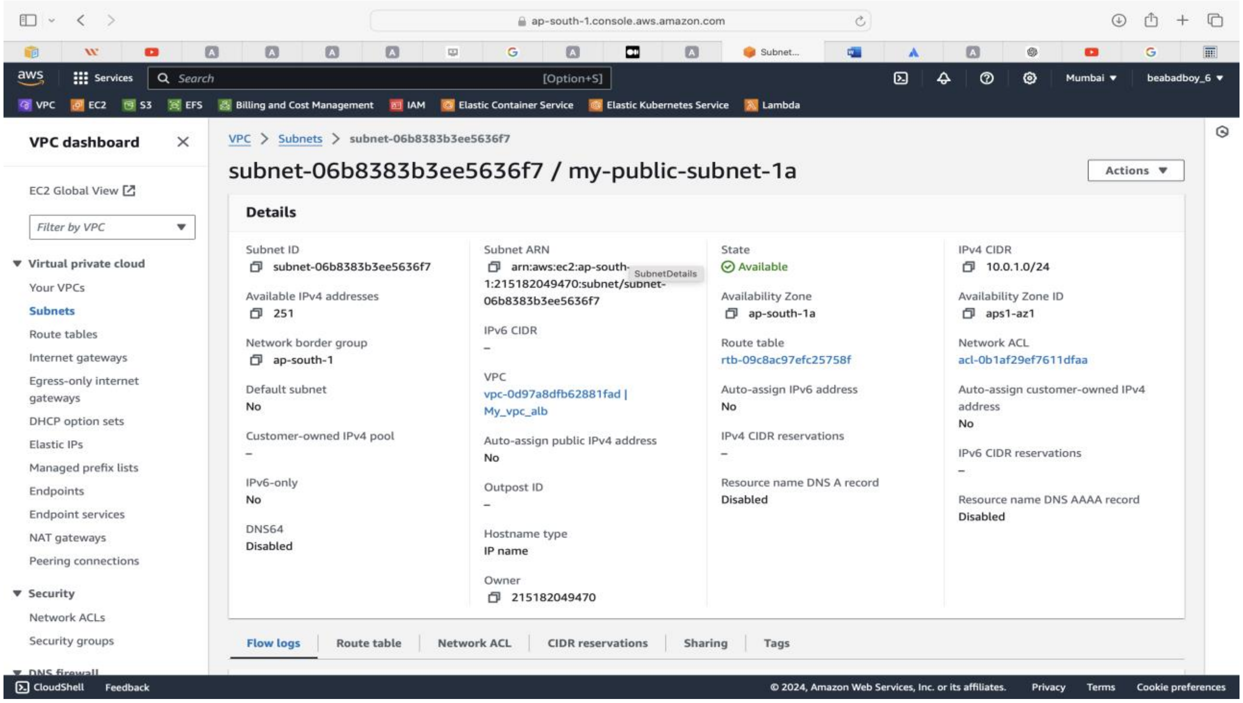
Resource map Info

VPC Show details
Your AWS virtual network
My_vpc_alb

Subnets (4)
Subnets within this VPC
ap-south-1a
my-private-subnet-1a

Route tables (1)
Route network traffic to resources
rtb-09c8ac97efc25758f

Step 2: Now, Creating 2 public subnets and 2 private subnets, with required configurations.



subnet-06b8383b3ee5636f7 / my-public-subnet-1a

Details

Subnet ID
subnet-06b8383b3ee5636f7

Available IPv4 addresses
251

Network border group
ap-south-1

Default subnet
No

Customer-owned IPv4 pool
–

IPv6-only
No

DNS64
Disabled

Subnet ARN
arn:aws:ec2:ap-south-1:215182049470:subnet/subnet-06b8383b3ee5636f7

IPv6 CIDR
–

VPC
vpc-0d97a8dfb62881fad | My_vpc_alb

Auto-assign public IPv4 address
No

Outpost ID
–

Hostname type
IP name

Owner
215182049470

State
Available

Availability Zone
ap-south-1a

Route table
rtb-09c8ac97efc25758f

Auto-assign IPv6 address
No

IPv4 CIDR reservations
–

Resource name DNS A record
Disabled

IPv4 CIDR
10.0.1.0/24

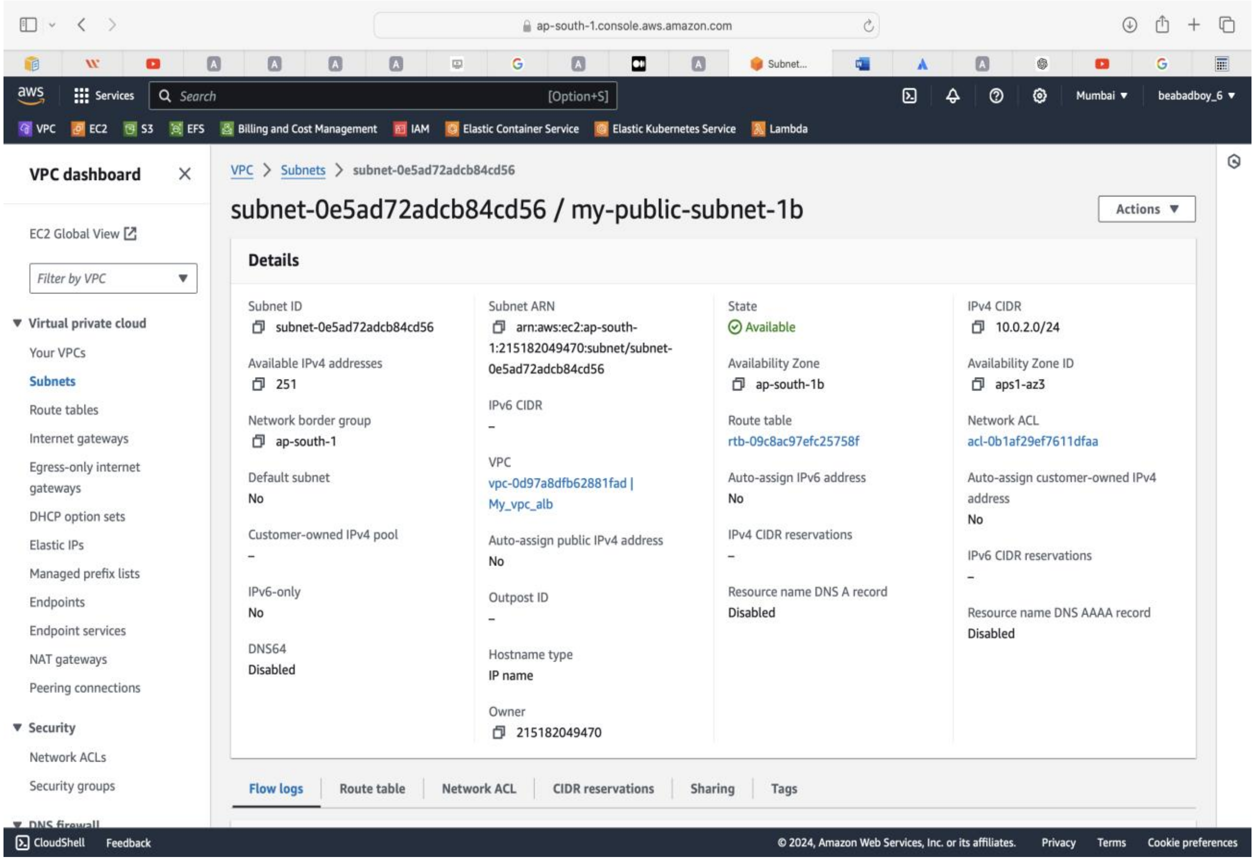
Availability Zone ID
aps1-az1

Network ACL
acl-0b1af29ef7611dfaa

Auto-assign customer-owned IPv4 address
No

IPv6 CIDR reservations
–

Resource name DNS AAAA record
Disabled



subnet-0e5ad72adcb84cd56 / my-public-subnet-1b

Details

Subnet ID
subnet-0e5ad72adcb84cd56

Available IPv4 addresses
251

Network border group
ap-south-1

Default subnet
No

Customer-owned IPv4 pool
–

IPv6-only
No

DNS64
Disabled

Subnet ARN
arn:aws:ec2:ap-south-1:215182049470:subnet/subnet-0e5ad72adcb84cd56

IPv6 CIDR
–

VPC
vpc-0d97a8dfb62881fad | My_vpc_alb

Auto-assign public IPv4 address
No

Outpost ID
–

Hostname type
IP name

Owner
215182049470

State
Available

Availability Zone
ap-south-1b

Route table
rtb-09c8ac97efc25758f

Auto-assign IPv6 address
No

IPv4 CIDR reservations
–

Resource name DNS A record
Disabled

IPv4 CIDR
10.0.2.0/24

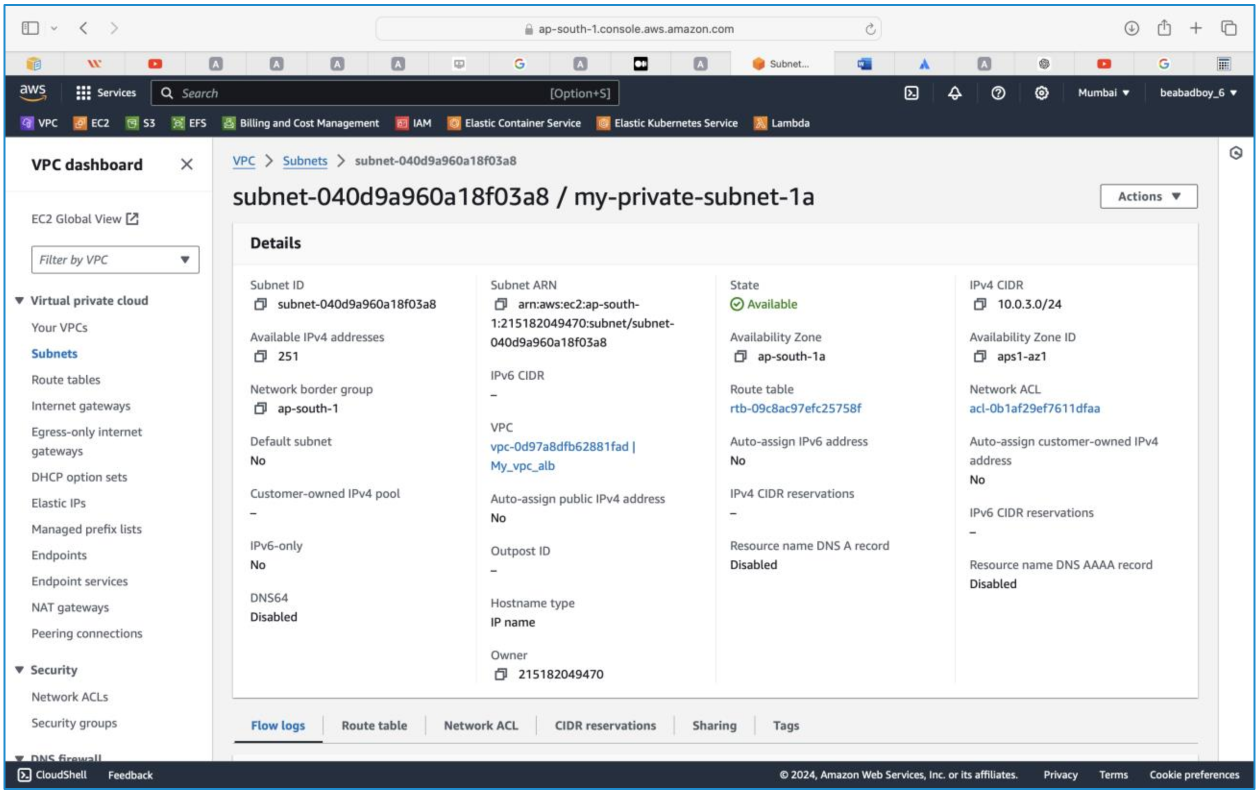
Availability Zone ID
aps1-az3

Network ACL
acl-0b1af29ef7611dfaa

Auto-assign customer-owned IPv4 address
No

IPv6 CIDR reservations
–

Resource name DNS AAAA record
Disabled

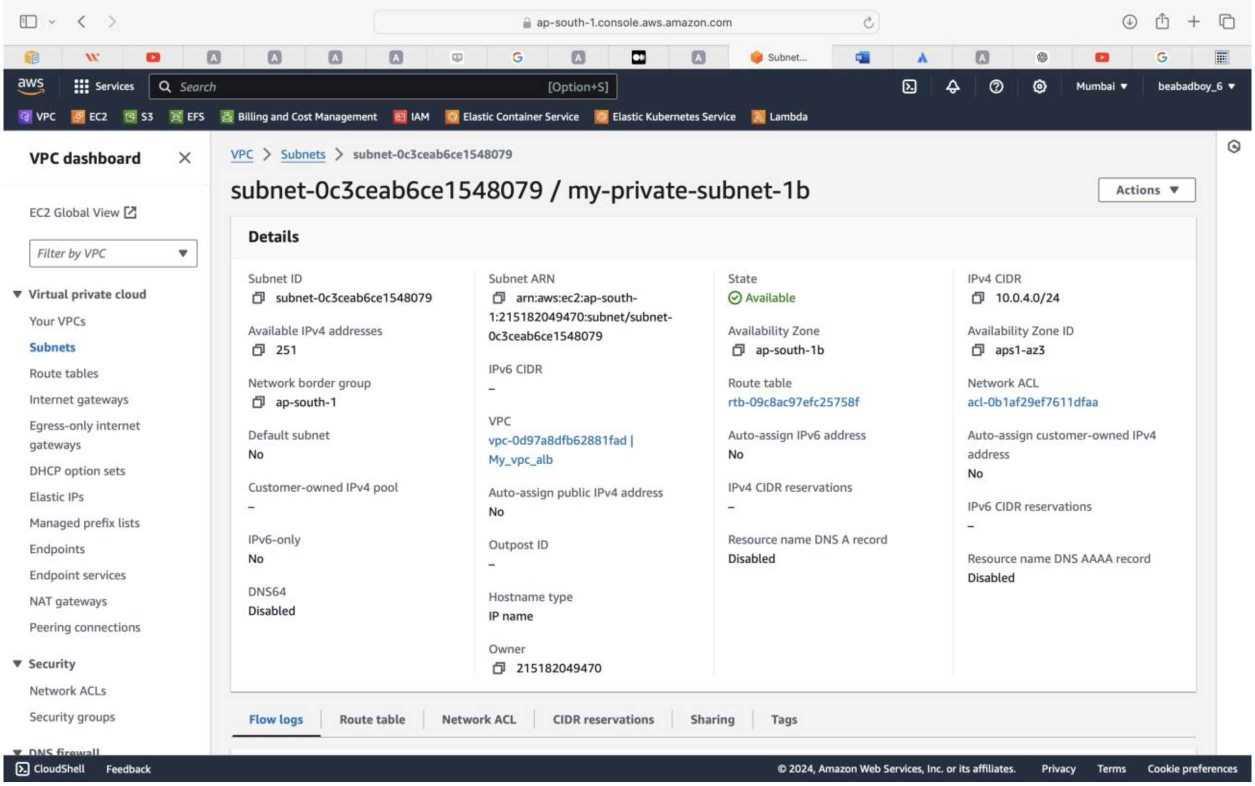


VPC > Subnets > subnet-040d9a960a18f03a8

subnet-040d9a960a18f03a8 / my-private-subnet-1a

Details

Subnet ID subnet-040d9a960a18f03a8	Subnet ARN arn:aws:ec2:ap-south-1:215182049470:subnet/subnet-040d9a960a18f03a8	State Available	IPv4 CIDR 10.0.3.0/24
Available IPv4 addresses 251	IPv6 CIDR –	Availability Zone ap-south-1a	Availability Zone ID aps1-az1
Network border group ap-south-1	VPC vpc-0d97a8dfb62881fad | My_vpc_alb	Route table rtb-09c8ac97efc25758f	Network ACL acl-0b1af29ef7611dfaa
Default subnet No	Auto-assign public IPv4 address No	Auto-assign IPv6 address No	Auto-assign customer-owned IPv4 address No
Customer-owned IPv4 pool –	Outpost ID –	IPv4 CIDR reservations –	IPv6 CIDR reservations –
IPv6-only No	Hostname type IP name	Resource name DNS A record Disabled	Resource name DNS AAAA record Disabled
DNS64 Disabled	Owner 215182049470		



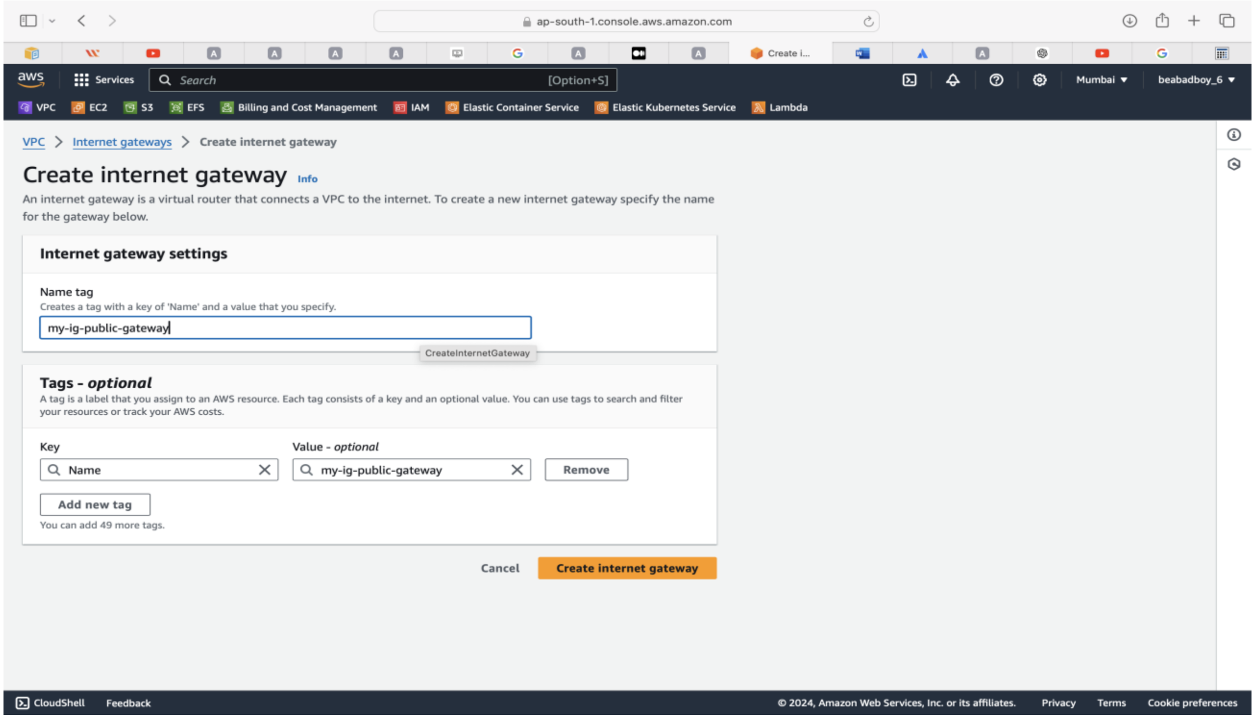
VPC > Subnets > subnet-0c3ceab6ce1548079

subnet-0c3ceab6ce1548079 / my-private-subnet-1b

Details

Subnet ID subnet-0c3ceab6ce1548079	Subnet ARN arn:aws:ec2:ap-south-1:215182049470:subnet/subnet-0c3ceab6ce1548079	State Available	IPv4 CIDR 10.0.4.0/24
Available IPv4 addresses 251	IPv6 CIDR –	Availability Zone ap-south-1b	Availability Zone ID aps1-az3
Network border group ap-south-1	VPC vpc-0d97a8dfb62881fad | My_vpc_alb	Route table rtb-09c8ac97efc25758f	Network ACL acl-0b1af29ef7611dfaa
Default subnet No	Auto-assign public IPv4 address No	Auto-assign IPv6 address No	Auto-assign customer-owned IPv4 address No
Customer-owned IPv4 pool –	Outpost ID –	IPv4 CIDR reservations –	IPv6 CIDR reservations –
IPv6-only No	Hostname type IP name	Resource name DNS A record Disabled	Resource name DNS AAAA record Disabled
DNS64 Disabled	Owner 215182049470		

Step 3: Create an internet gateway and associatie it with public subnets. And, association will take place when you will go and click on subnet association and attach those 2 public subnets with Internet Gateway.



ap-south-1.console.aws.amazon.com
Services
Search [Option+S]
Mumbai
beabadboy_6
VPC EC2 S3 EFS Billing and Cost Management IAM Elastic Container Service Elastic Kubernetes Service Lambda

VPC > Internet gateways > Create internet gateway

Create internet gateway Info

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.

Internet gateway settings

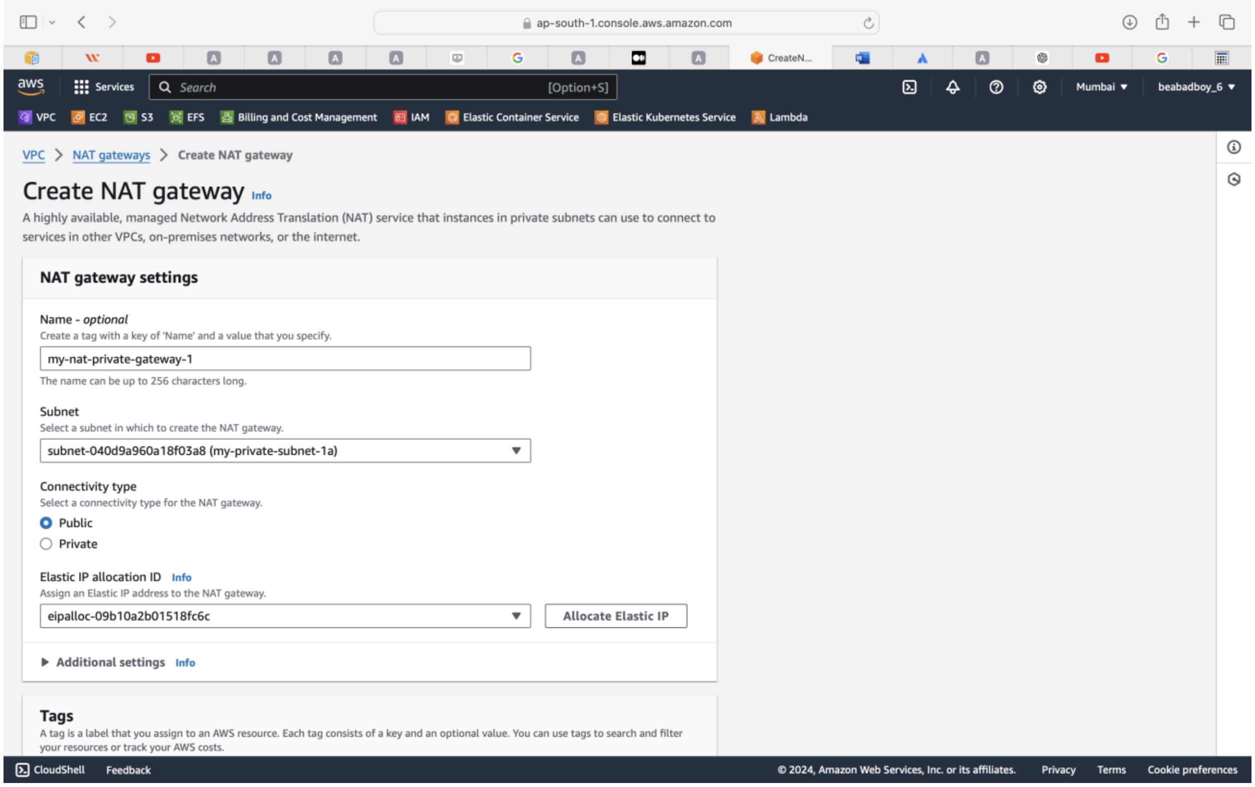
Name tag
Creates a tag with a key of 'Name' and a value that you specify.
my-ig-public-gateway

Tags - *optional*
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*
Name	my-ig-public-gateway

You can add 49 more tags.

Step 4: Create a Nat gateway and associate it with private subnets. And association will take place when you will go and click on subnet association and attach those 2 private subnets with Nat Gateway.



ap-south-1.console.aws.amazon.com
Services
Search [Option+S]
Mumbai
beabadboy_6
VPC EC2 S3 EFS Billing and Cost Management IAM Elastic Container Service Elastic Kubernetes Service Lambda

VPC > NAT gateways > 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.
my-nat-private-gateway-1
The name can be up to 256 characters long.

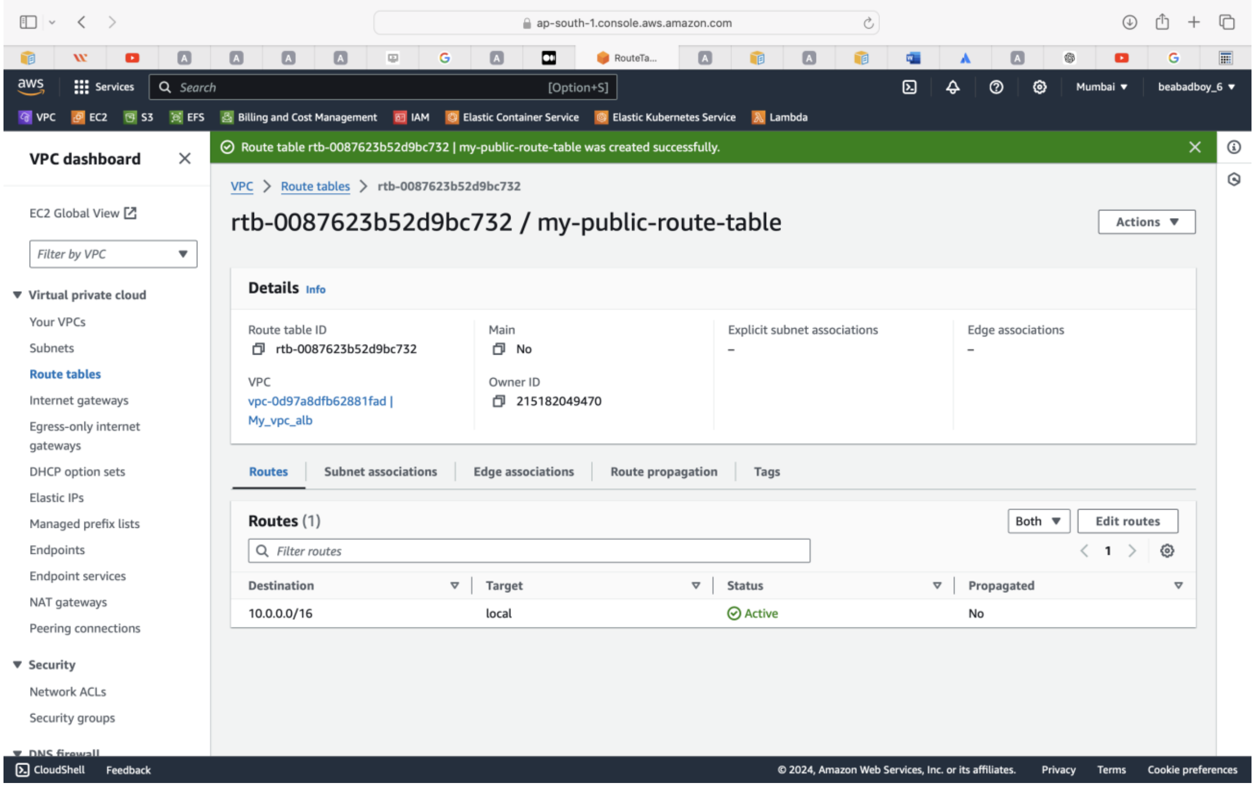
Subnet
Select a subnet in which to create the NAT gateway.
subnet-040d9a960a18f03a8 (my-private-subnet-1a)

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-09b10a2b01518fc6c

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Step 5: Also, create 2 route tables one for the public subnet and one for the private Subnet. For, the public in inbound security give http port 80 or 443 and for private give the inbound security ssh 23.



Route table rtb-0087623b52d9bc732 | my-public-route-table was created successfully.

VPC > Route tables > rtb-0087623b52d9bc732

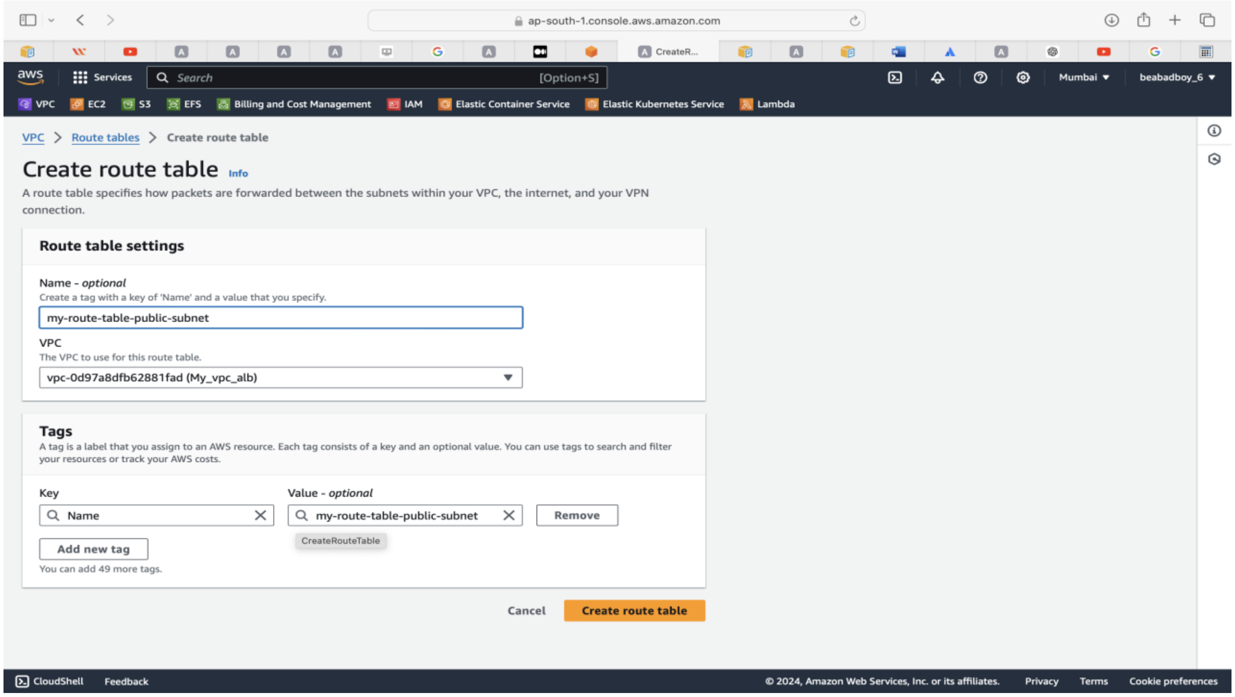
rtb-0087623b52d9bc732 / my-public-route-table

Details Info

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-0087623b52d9bc732	No	–	–
VPC	Owner ID		
vpc-0d97a8dfb62881fad | My_vpc_alb	215182049470		

Routes (1)

Destination	Target	Status	Propagated
10.0.0.0/16	local	✓ Active	No



VPC > Route tables > Create route table

Create route table Info

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

Route table settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.
my-route-table-public-subnet

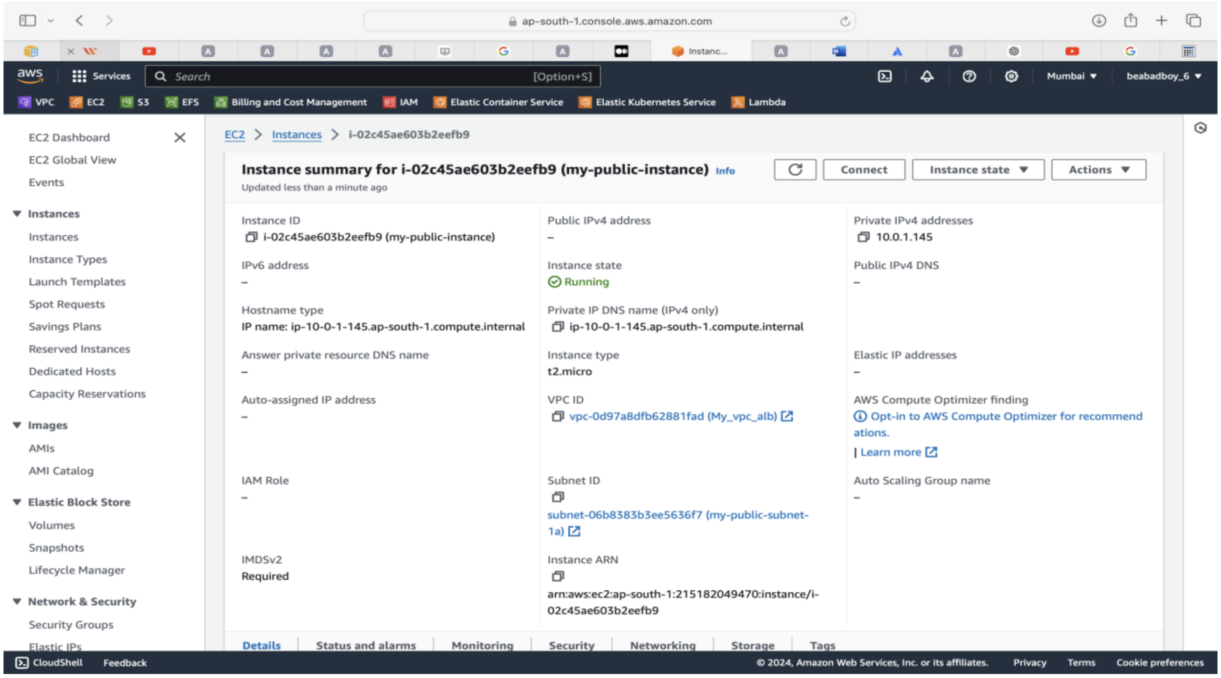
VPC
The VPC to use for this route table.
vpc-0d97a8dfb62881fad (My_vpc_alb)

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*
Name	my-route-table-public-subnet

You can add 49 more tags.

And, similarly for the private subnet. Step 6: Creating an instance in public subnet and naming it as a public instance.



Instance summary for i-02c45ae603b2eefb9 (my-public-instance) Info

Updated less than a minute ago

Instance ID
i-02c45ae603b2eefb9 (my-public-instance)

Public IPv4 address
–

Private IPv4 addresses
10.0.1.145

IPv6 address
–

Instance state
Running

Public IPv4 DNS
–

Hostname type
IP name: ip-10-0-1-145.ap-south-1.compute.internal

Private IP DNS name (IPv4 only)
ip-10-0-1-145.ap-south-1.compute.internal

Answer private resource DNS name
–

Instance type
t2.micro

Elastic IP addresses
–

Auto-assigned IP address
–

VPC ID
vpc-0d97a8dfb62881fad (My_vpc_alb)

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations.
| Learn more

IAM Role
–

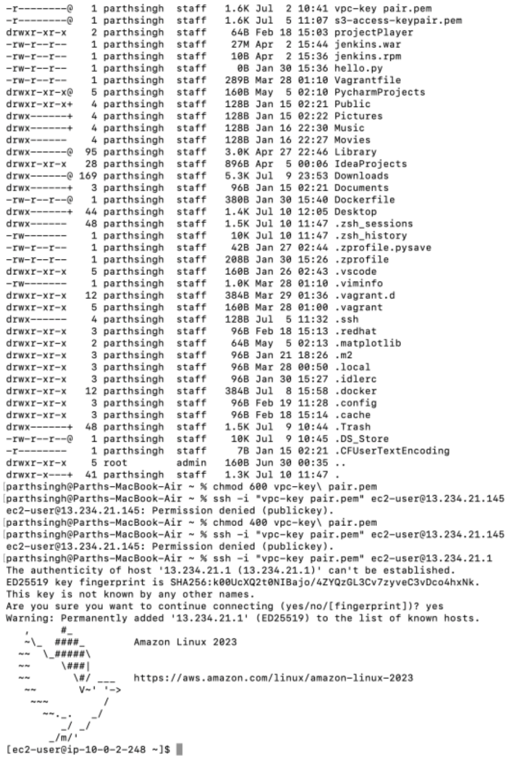
Subnet ID
subnet-06b8383b3ee5636f7 (my-public-subnet-1a)

Auto Scaling Group name
–

IMDSv2
Required

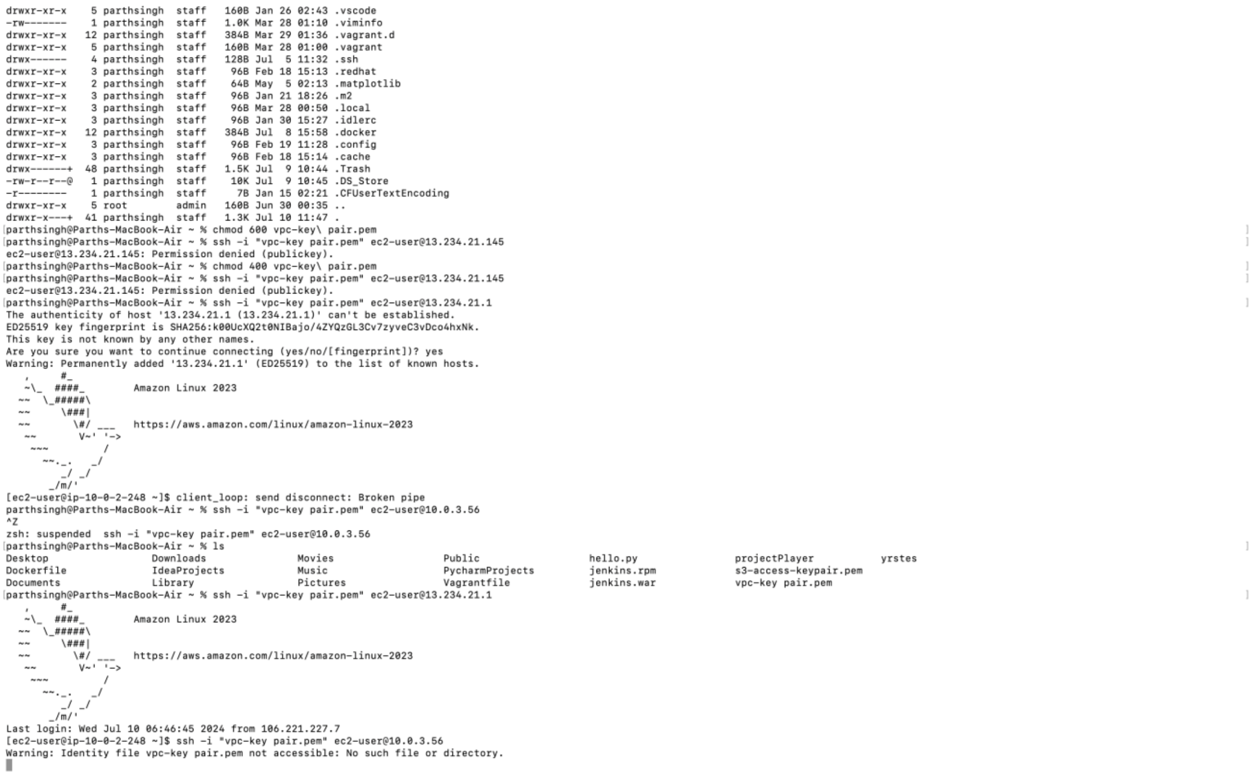
Instance ARN
arn:aws:ec2:ap-south-1:215182049470:instance/i-02c45ae603b2eefb9

Step 7: Now, we will run the instance or will go inside the instance through cli by running some linux commands.



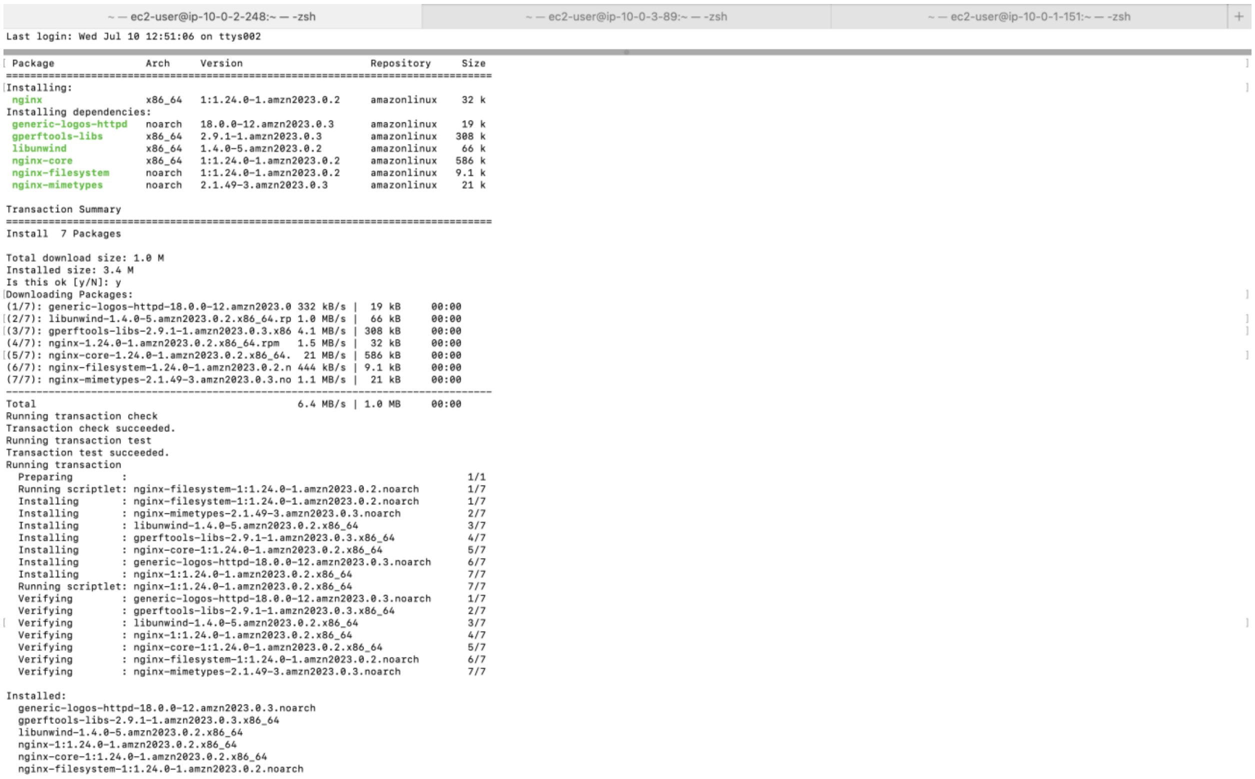
```
-r--------@   1 parthsingh  staff   1.6K Jul  2 10:41 vpc-key pair.pem
-r--------@   1 parthsingh  staff   1.6K Jul  5 11:07 s3-access-keypair.pem
drwxr-xr-x    2 parthsingh  staff    64B Feb 18 15:03 projectPlayer
-rw-r--r--    1 parthsingh  staff    27M Apr  2 15:44 jenkins.war
-rw-r--r--    1 parthsingh  staff    10B Apr  2 15:36 jenkins.rpm
-rw-r--r--    1 parthsingh  staff     0B Jan 30 15:36 hello.py
-rw-r--r--    1 parthsingh  staff   289B Mar 28 01:10 Vagrantfile
drwxr-xr-x@   5 parthsingh  staff   160B May  5 02:10 PycharmProjects
drwxr-xr-x+   4 parthsingh  staff   128B Jan 15 02:21 Public
drwx------+   4 parthsingh  staff   128B Jan 15 02:22 Pictures
drwx------+   4 parthsingh  staff   128B Jan 16 22:30 Music
drwx------    4 parthsingh  staff   128B Jan 16 22:27 Movies
drwx------@  95 parthsingh  staff   3.0K Apr 27 22:46 Library
drwxr-xr-x   28 parthsingh  staff   896B Apr  5 00:06 IdeaProjects
drwx------@ 169 parthsingh  staff   5.3K Jul  9 23:53 Downloads
drwx------+   3 parthsingh  staff    96B Jan 15 02:21 Documents
-rw-r--r--@   1 parthsingh  staff   380B Jan 30 15:40 Dockerfile
drwx------+  44 parthsingh  staff   1.4K Jul 10 12:05 Desktop
drwx------   48 parthsingh  staff   1.5K Jul 10 11:47 .zsh_sessions
-rw-------    1 parthsingh  staff    10K Jul 10 11:47 .zsh_history
-rw-r--r--    1 parthsingh  staff    42B Jan 27 02:44 .zprofile.pysave
-rw-r--r--    1 parthsingh  staff   208B Jan 30 15:26 .zprofile
drwxr-xr-x    5 parthsingh  staff   160B Jan 26 02:43 .vscode
-rw-------    1 parthsingh  staff   1.0K Mar 28 01:10 .viminfo
drwxr-xr-x   12 parthsingh  staff   384B Mar 29 01:36 .vagrant.d
drwxr-xr-x    5 parthsingh  staff   160B Mar 28 01:00 .vagrant
drwx------    4 parthsingh  staff   128B Jul  5 11:32 .ssh
drwxr-xr-x    3 parthsingh  staff    96B Feb 18 15:13 .redhat
drwxr-xr-x    2 parthsingh  staff    64B May  5 02:13 .matplotlib
drwxr-xr-x    3 parthsingh  staff    96B Jan 21 18:26 .m2
drwxr-xr-x    3 parthsingh  staff    96B Mar 28 00:50 .local
drwxr-xr-x    3 parthsingh  staff    96B Jan 30 15:27 .idlerc
drwxr-xr-x   12 parthsingh  staff   384B Jul  8 15:58 .docker
drwxr-xr-x    3 parthsingh  staff    96B Feb 19 11:28 .config
drwxr-xr-x    3 parthsingh  staff    96B Feb 18 15:14 .cache
drwx------+  48 parthsingh  staff   1.5K Jul  9 10:44 .Trash
-rw-r--r--@   1 parthsingh  staff    10K Jul  9 10:45 .DS_Store
-r--------    1 parthsingh  staff     7B Jan 15 02:21 .CFUserTextEncoding
drwxr-xr-x    5 root        admin   160B Jun 30 00:35 ..
drwxr-x---+  41 parthsingh  staff   1.3K Jul 10 11:47 .
parthsingh@Parths-MacBook-Air ~ % chmod 600 vpc-key\ pair.pem
parthsingh@Parths-MacBook-Air ~ % ssh -i "vpc-key pair.pem" ec2-user@13.234.21.145
ec2-user@13.234.21.145: Permission denied (publickey).
parthsingh@Parths-MacBook-Air ~ % chmod 400 vpc-key\ pair.pem
parthsingh@Parths-MacBook-Air ~ % ssh -i "vpc-key pair.pem" ec2-user@13.234.21.145
ec2-user@13.234.21.145: Permission denied (publickey).
parthsingh@Parths-MacBook-Air ~ % ssh -i "vpc-key pair.pem" ec2-user@13.234.21.1
The authenticity of host '13.234.21.1 (13.234.21.1)' can't be established.
ED25519 key fingerprint is SHA256:k00UcXQ2t0NIBajo/4ZYQzGL3Cv7zyveC3vDco4hxNk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.234.21.1' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-2-248 ~]$ 
```

Step 8: Now, we will access or go inside the private instance by taking the help of public instance.



```
drwxr-xr-x    5 parthsingh  staff   160B Jan 26 02:43 .vscode
-rw-------    1 parthsingh  staff   1.0K Mar 28 01:10 .viminfo
drwxr-xr-x   12 parthsingh  staff   384B Mar 29 01:36 .vagrant.d
drwxr-xr-x    5 parthsingh  staff   160B Mar 28 01:00 .vagrant
drwx------    4 parthsingh  staff   128B Jul  5 11:32 .ssh
drwxr-xr-x    3 parthsingh  staff    96B Feb 18 15:13 .redhat
drwxr-xr-x    2 parthsingh  staff    64B May  5 02:13 .matplotlib
drwxr-xr-x    3 parthsingh  staff    96B Jan 21 18:26 .m2
drwxr-xr-x    3 parthsingh  staff    96B Mar 28 00:50 .local
drwxr-xr-x    3 parthsingh  staff    96B Jan 30 15:27 .idlerc
drwxr-xr-x   12 parthsingh  staff   384B Jul  8 15:58 .docker
drwxr-xr-x    3 parthsingh  staff    96B Feb 19 11:28 .config
drwxr-xr-x    3 parthsingh  staff    96B Feb 18 15:14 .cache
drwx------+  48 parthsingh  staff   1.5K Jul  9 10:44 .Trash
-rw-r--r--@   1 parthsingh  staff    10K Jul  9 10:45 .DS_Store
-r--------    1 parthsingh  staff     7B Jan 15 02:21 .CFUserTextEncoding
drwxr-xr-x    5 root        admin   160B Jun 30 00:35 ..
drwxr-x---+  41 parthsingh  staff   1.3K Jul 10 11:47 .
parthsingh@Parths-MacBook-Air ~ % chmod 600 vpc-key\ pair.pem
parthsingh@Parths-MacBook-Air ~ % ssh -i "vpc-key pair.pem" ec2-user@13.234.21.145
ec2-user@13.234.21.145: Permission denied (publickey).
parthsingh@Parths-MacBook-Air ~ % chmod 400 vpc-key\ pair.pem
parthsingh@Parths-MacBook-Air ~ % ssh -i "vpc-key pair.pem" ec2-user@13.234.21.145
ec2-user@13.234.21.145: Permission denied (publickey).
parthsingh@Parths-MacBook-Air ~ % ssh -i "vpc-key pair.pem" ec2-user@13.234.21.1
The authenticity of host '13.234.21.1 (13.234.21.1)' can't be established.
ED25519 key fingerprint is SHA256:k00UcXQ2t0NIBajo/4ZYQzGL3Cv7zyveC3vDco4hxNk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.234.21.1' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-2-248 ~]$ client_loop: send disconnect: Broken pipe
parthsingh@Parths-MacBook-Air ~ % ssh -i "vpc-key pair.pem" ec2-user@10.0.3.56
^Z
zsh: suspended  ssh -i "vpc-key pair.pem" ec2-user@10.0.3.56
parthsingh@Parths-MacBook-Air ~ % ls
Desktop                 Downloads               Movies                  Public                  hello.py                projectPlayer           yrstes
Dockerfile              IdeaProjects            Music                   PycharmProjects         jenkins.rpm             s3-access-keypair.pem
Documents               Library                 Pictures                Vagrantfile             jenkins.war             vpc-key pair.pem
parthsingh@Parths-MacBook-Air ~ % ssh -i "vpc-key pair.pem" ec2-user@13.234.21.1
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Wed Jul 10 06:46:45 2024 from 106.221.227.7
[ec2-user@ip-10-0-2-248 ~]$ ssh -i "vpc-key pair.pem" ec2-user@10.0.3.56
Warning: Identity file vpc-key pair.pem not accessible: No such file or directory.
```

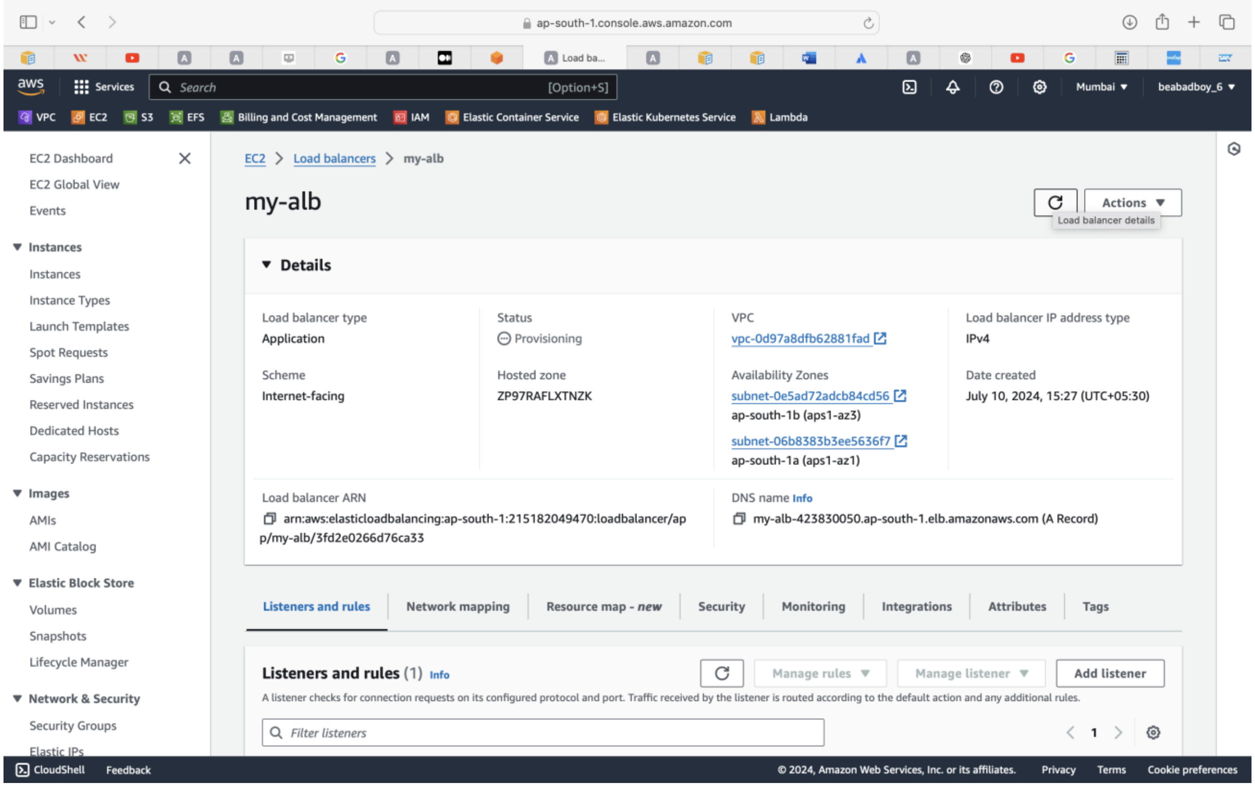
Step 9 : Now, we will install ngnix server on private ec2 instance. Ngnix



~ — ec2-user@ip-10-0-2-248:~ — -zsh
~ — ec2-user@ip-10-0-3-89:~ — -zsh
~ — ec2-user@ip-10-0-1-151:~ — -zsh

```
Last login: Wed Jul 10 12:51:06 on ttys002

 Package                Arch     Version                        Repository     Size
================================================================================
Installing:
 nginx                  x86_64   1:1.24.0-1.amzn2023.0.2        amazonlinux    32 k
Installing dependencies:
 generic-logos-httpd    noarch   18.0.0-12.amzn2023.0.3         amazonlinux    19 k
 gperftools-libs        x86_64   2.9.1-1.amzn2023.0.3           amazonlinux   308 k
 libunwind              x86_64   1.4.0-5.amzn2023.0.2           amazonlinux    66 k
 nginx-core             x86_64   1:1.24.0-1.amzn2023.0.2        amazonlinux   586 k
 nginx-filesystem       noarch   1:1.24.0-1.amzn2023.0.2        amazonlinux   9.1 k
 nginx-mimetypes        noarch   2.1.49-3.amzn2023.0.3          amazonlinux    21 k

Transaction Summary
================================================================================
Install  7 Packages

Total download size: 1.0 M
Installed size: 3.4 M
Is this ok [y/N]: y
Downloading Packages:
(1/7): generic-logos-httpd-18.0.0-12.amzn2023.0 332 kB/s |  19 kB     00:00
(2/7): libunwind-1.4.0-5.amzn2023.0.2.x86_64.rp 1.0 MB/s |  66 kB     00:00
(3/7): gperftools-libs-2.9.1-1.amzn2023.0.3.x86 4.1 MB/s | 308 kB     00:00
(4/7): nginx-1.24.0-1.amzn2023.0.2.x86_64.rpm   1.5 MB/s |  32 kB     00:00
(5/7): nginx-core-1.24.0-1.amzn2023.0.2.x86_64.  21 MB/s | 586 kB     00:00
(6/7): nginx-filesystem-1.24.0-1.amzn2023.0.2.n 444 kB/s | 9.1 kB     00:00
(7/7): nginx-mimetypes-2.1.49-3.amzn2023.0.3.no 1.1 MB/s |  21 kB     00:00
--------------------------------------------------------------------------------
Total                                           6.4 MB/s | 1.0 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Running scriptlet: nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch        1/7
  Installing       : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch        1/7
  Installing       : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch           2/7
  Installing       : libunwind-1.4.0-5.amzn2023.0.2.x86_64                  3/7
  Installing       : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64            4/7
  Installing       : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64              5/7
  Installing       : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch      6/7
  Installing       : nginx-1:1.24.0-1.amzn2023.0.2.x86_64                   7/7
  Running scriptlet: nginx-1:1.24.0-1.amzn2023.0.2.x86_64                   7/7
  Verifying        : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch      1/7
  Verifying        : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64            2/7
  Verifying        : libunwind-1.4.0-5.amzn2023.0.2.x86_64                  3/7
  Verifying        : nginx-1:1.24.0-1.amzn2023.0.2.x86_64                   4/7
  Verifying        : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64              5/7
  Verifying        : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch        6/7
  Verifying        : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch           7/7

Installed:
  generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
  gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64
  libunwind-1.4.0-5.amzn2023.0.2.x86_64
  nginx-1:1.24.0-1.amzn2023.0.2.x86_64
  nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64
  nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch
```

Step 10: Now, we will create alb in public subnet and also create a target group which will shift or point the load of alb towards private instance.



my-alb

▼ Details

Load balancer type
Application

Status
Provisioning

VPC
vpc-0d97a8dfb62881fad

Load balancer IP address type
IPv4

Scheme
Internet-facing

Hosted zone
ZP97RAFLXTNZK

Availability Zones
subnet-0e5ad72adcb84cd56
ap-south-1b (aps1-az3)
subnet-06b8383b3ee5636f7
ap-south-1a (aps1-az1)

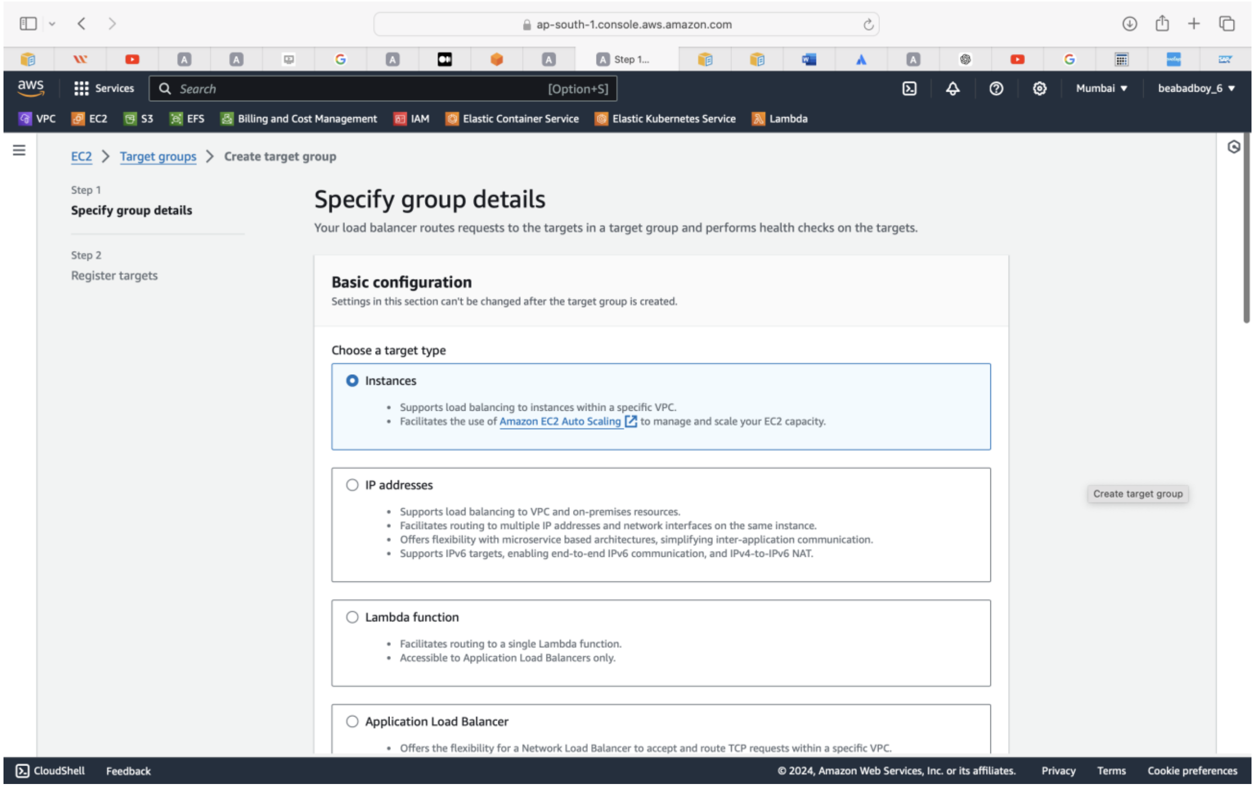
Date created
July 10, 2024, 15:27 (UTC+05:30)

Load balancer ARN
arn:aws:elasticloadbalancing:ap-south-1:215182049470:loadbalancer/app/my-alb/3fd2e0266d76ca33

DNS name Info
my-alb-423830050.ap-south-1.elb.amazonaws.com (A Record)

Listeners and rules (1) Info

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed to the default action and any additional rules.


ap-south-1.console.aws.amazon.com
Services
Search
[Option+S]
Mumbai ▾
beabadboy_6 ▾
VPC
EC2
S3
EFS
Billing and Cost Management
IAM
Elastic Container Service
Elastic Kubernetes Service
Lambda
Create target group
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

EC2 > Target groups > Create target group

Step 1
Specify group details

Step 2
Register targets

Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration
Settings in this section can't be changed after the target group is created.

Choose a target type

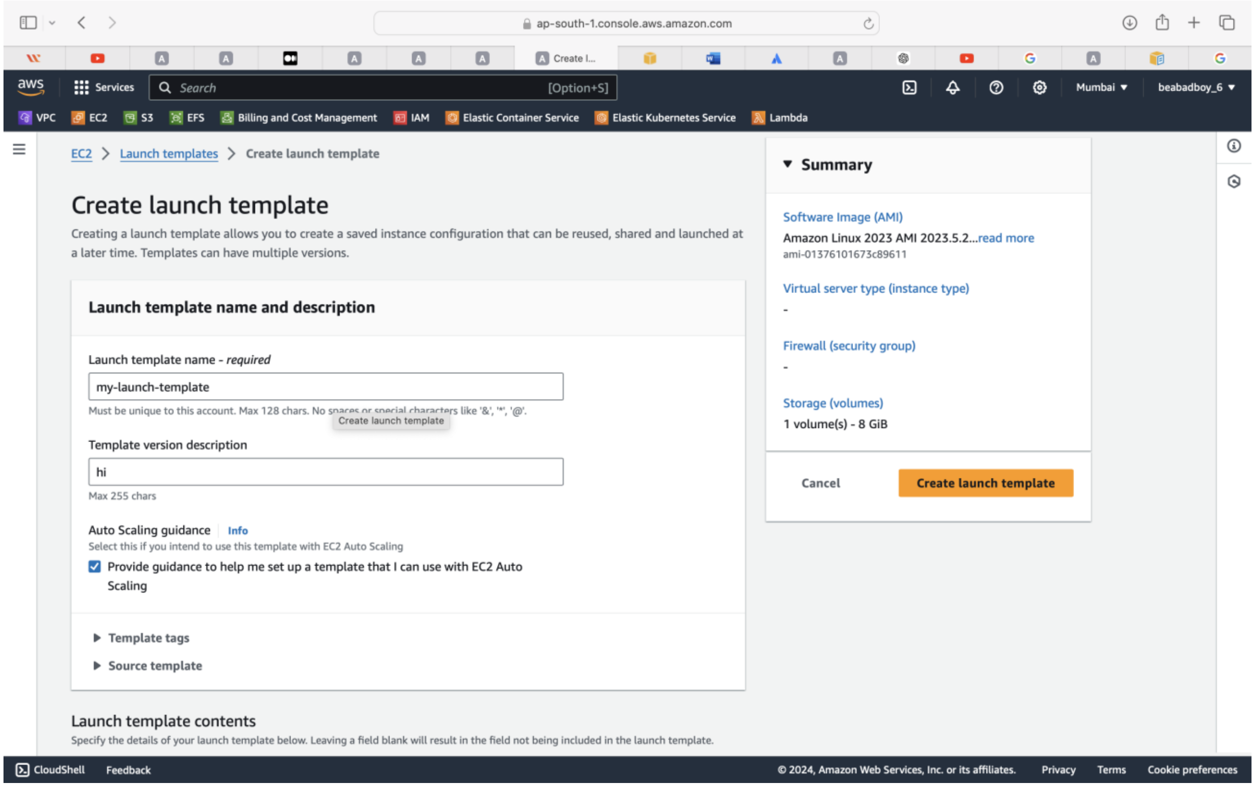
● Instances
- Supports load balancing to instances within a specific VPC.
- Facilitates the use of Amazon EC2 Auto Scaling to manage and scale your EC2 capacity.

○ IP addresses
- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

○ Lambda function
- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

○ Application Load Balancer
- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.

Step 11: Now, we will create auto-scaling groups, and for that first we have to create a launch template(a launch template contains configurational data of an instance).


ap-south-1.console.aws.amazon.com
Services
Search
[Option+S]
Mumbai ▾
beabadboy_6 ▾
VPC
EC2
S3
EFS
Billing and Cost Management
IAM
Elastic Container Service
Elastic Kubernetes Service
Lambda
Create launch template
Cancel
Create launch template
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

EC2 > Launch templates > Create launch template

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - *required*
my-launch-template
Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description
hi
Max 255 chars

Auto Scaling guidance Info
Select this if you intend to use this template with EC2 Auto Scaling
☑ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

▸ Template tags
▸ Source template

Launch template contents
Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

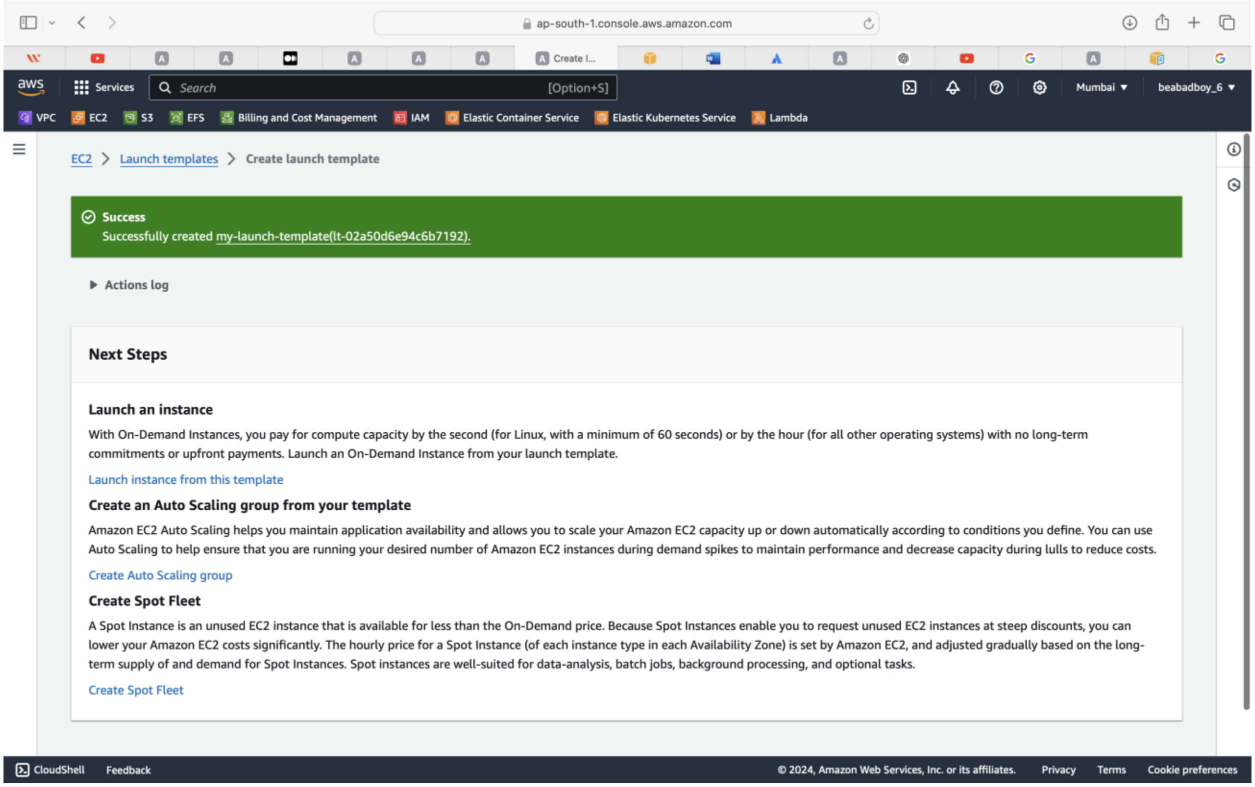
▾ **Summary**

Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...read more
ami-01376101673c89611

Virtual server type (instance type)
-

Firewall (security group)
-

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



ap-south-1.console.aws.amazon.com
Create l...
aws
Services
Search
[Option+S]
Mumbai ▾
beabadboy_6 ▾
VPC
EC2
S3
EFS
Billing and Cost Management
IAM
Elastic Container Service
Elastic Kubernetes Service
Lambda

EC2 > Launch templates > Create launch template

Success
Successfully created my-launch-template(lt-02a50d6e94c6b7192).

▶ Actions log

Next Steps

Launch an instance

With On-Demand Instances, you pay for compute capacity by the second (for Linux, with a minimum of 60 seconds) or by the hour (for all other operating systems) with no long-term commitments or upfront payments. Launch an On-Demand Instance from your launch template.

Launch instance from this template

Create an Auto Scaling group from your template

Amazon EC2 Auto Scaling helps you maintain application availability and allows you to scale your Amazon EC2 capacity up or down automatically according to conditions you define. You can use Auto Scaling to help ensure that you are running your desired number of Amazon EC2 instances during demand spikes to maintain performance and decrease capacity during lulls to reduce costs.

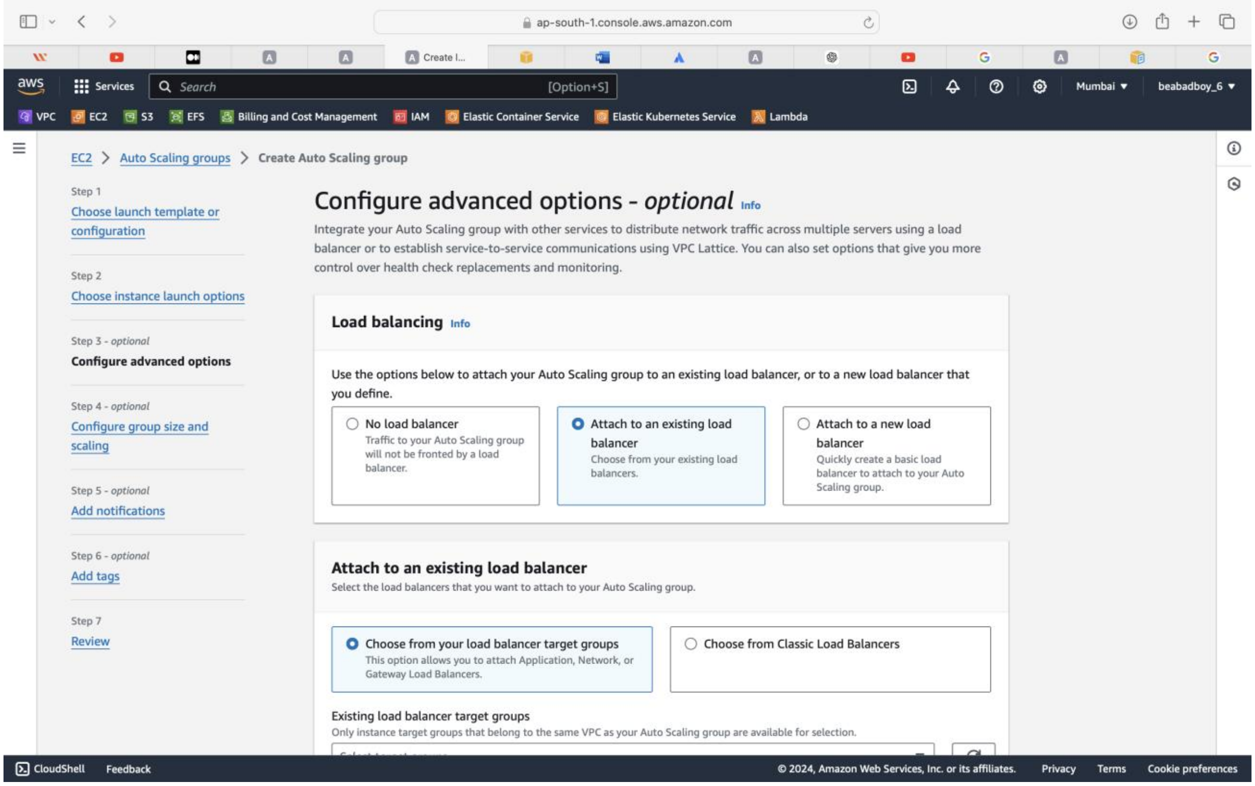
Create Auto Scaling group

Create Spot Fleet

A Spot Instance is an unused EC2 instance that is available for less than the On-Demand price. Because Spot Instances enable you to request unused EC2 instances at steep discounts, you can lower your Amazon EC2 costs significantly. The hourly price for a Spot Instance (of each instance type in each Availability Zone) is set by Amazon EC2, and adjusted gradually based on the long-term supply of and demand for Spot Instances. Spot instances are well-suited for data-analysis, batch jobs, background processing, and optional tasks.

Create Spot Fleet

CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



ap-south-1.console.aws.amazon.com
Create l...
aws
Services
Search
[Option+S]
Mumbai ▾
beabadboy_6 ▾
VPC
EC2
S3
EFS
Billing and Cost Management
IAM
Elastic Container Service
Elastic Kubernetes Service
Lambda

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
Choose launch template or configuration

Step 2
Choose instance launch options

Step 3 - *optional*
Configure advanced options

Step 4 - *optional*
Configure group size and scaling

Step 5 - *optional*
Add notifications

Step 6 - *optional*
Add tags

Step 7
Review

Configure advanced options - *optional* Info

Integrate your Auto Scaling group with other services to distribute network traffic across multiple servers using a load balancer or to establish service-to-service communications using VPC Lattice. You can also set options that give you more control over health check replacements and monitoring.

Load balancing Info

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

○ No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.

● Attach to an existing load balancer
Choose from your existing load balancers.

○ Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.

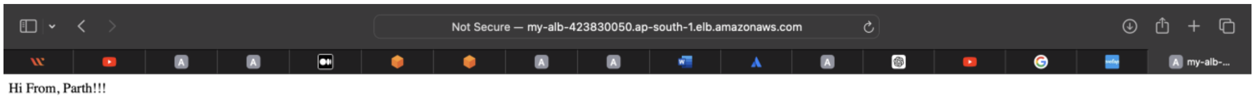
Attach to an existing load balancer
Select the load balancers that you want to attach to your Auto Scaling group.

● Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.

○ Choose from Classic Load Balancers

Existing load balancer target groups
Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Step 10: After , editing our index.html file our final text is being displayed on web server.



Hi From, Parth!!!