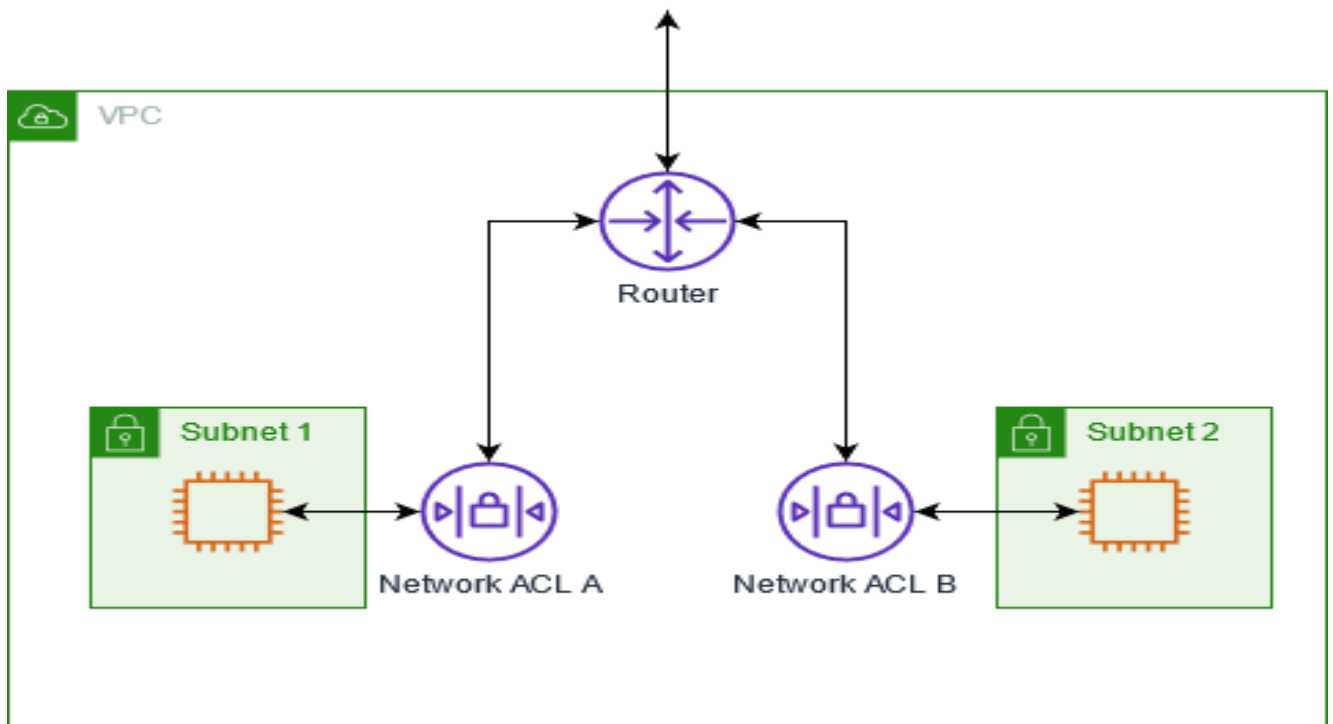


AWS Documentation

No:-	Content
1.	NACL
2.	NIC
3.	Placement Groups

1.NACL

A *network access control list (ACL)* allows or denies specific inbound or outbound traffic at the subnet level. You can use the default network ACL for your VPC, or you can create a custom network ACL for your VPC with rules that are similar to the rules for your security groups in order to add an additional layer of security to your VPC.



Steps:

- first go to VPC → select Network CLs → Provide name → Select VPC → create subnet.

The screenshot shows the 'Create network ACL' page in the AWS Management Console. The page has a dark header with the AWS logo, 'Services' menu, a search bar, and user information (N. Virginia, Rutugandh Shete). The breadcrumb trail is 'VPC > Network ACLs > Create network ACL'. The main heading is 'Create network ACL' with an 'Info' link. Below it is a subheading: 'A network ACL is an optional layer of security that acts as a firewall for controlling traffic in and out of a subnet.' The form is divided into two sections: 'Network ACL settings' and 'Tags'. In the 'Network ACL settings' section, there is a 'Name - optional' field with the value 'NACL' and a 'VPC' dropdown menu showing 'vpc-05ddac7397bf4a7fd (Default)'. The 'Tags' section has a 'Key' field with 'Name' and a 'Value - optional' field with 'NACL'. There is an 'Add tag' button and a note 'You can add 49 more tags'. At the bottom of the form are 'Cancel' and 'Create network ACL' buttons.

- NACL is created.

The screenshot shows the 'Network ACLs' page in the AWS Management Console. A green banner at the top says 'You successfully created acl-0a097ed1f33b6f179 / NACL.' The page title is 'Network ACLs (1/3)' with an 'Info' link. There is a search bar and a 'Find resources by attribute or tag' input. A table lists the created NACLs:

Name	Network ACL ID	Associated with	Default	VPC ID	Inbound rules
NACL	acl-02da0ec66795b8564	subnet-095d9bde1b5163fa	No	vpc-05ddac7397bf4a7fd / Default	3 Inb
-	acl-06eacccc38cccd4ab	5 Subnets	Yes	vpc-05ddac7397bf4a7fd / Default	2 Inb
NACL	acl-0a097ed1f33b6f179	-	No	vpc-05ddac7397bf4a7fd / Default	1 Inb

Below the table, the details for 'acl-02da0ec66795b8564 / NACL' are shown. The 'Details' tab is active, displaying the Network ACL ID, Associated with (subnet-095d9bde1b5163fa), Default (No), and VPC ID (vpc-05ddac7397bf4a7fd / Default).

- Select NACL → click on edit subent association.

The screenshot shows the 'Network ACLs' page in the AWS Management Console. A green banner at the top says 'You successfully created acl-0a097ed1f33b6f179 / NACL.' The page title is 'Network ACLs (1/3)' with an 'Info' link. There is a search bar and a 'Find resources by attribute or tag' input. A table lists the created NACLs:

Name	Network ACL ID	Associated with	Default	VPC ID	Inbound rules
-	acl-02da0ec66795b8564	subnet-095d9bde1b5163fa	No	vpc-05ddac7397bf4a7fd / Default	3 Inb
-	acl-06eacccc38cccd4ab	5 Subnets	Yes	vpc-05ddac7397bf4a7fd / Default	2 Inb
NACL	acl-0a097ed1f33b6f179	-	No	vpc-05ddac7397bf4a7fd / Default	1 Inb

Below the table, the details for 'acl-0a097ed1f33b6f179 / NACL' are shown. The 'Details' tab is active, displaying the Network ACL ID, Associated with (-), Default (No), and VPC ID (vpc-05ddac7397bf4a7fd / Default). A context menu is open over the 'Edit subnet associations' link, showing options: 'View details', 'Edit inbound rules', 'Edit outbound rules', 'Edit subnet associations' (highlighted), 'Manage tags', and 'Delete network ACLs'.

- Select one of the subnet → save changes.

Edit subnet associations [Info](#)

Change which subnets are associated with this network ACL.

Available subnets (1/6)

Filter subnet associations

	Name	Subnet ID	Associated with	Availability Zone	IPv4 CIDR	IPv6 CIDR
<input type="checkbox"/>	-	subnet-002adff84587e66a	acl-06eacccc38cccd4ab	us-east-1b	172.31.0.0/20	-
<input type="checkbox"/>	-	subnet-077b7165a4ea267d1	acl-06eacccc38cccd4ab	us-east-1e	172.31.48.0/20	-
<input checked="" type="checkbox"/>	-	subnet-095d9bde1b5163fa	acl-02da0ec66795b8564	us-east-1c	172.31.80.0/20	-
<input type="checkbox"/>	-	subnet-0d61eac77a6d87021	acl-06eacccc38cccd4ab	us-east-1d	172.31.16.0/20	-
<input type="checkbox"/>	-	subnet-030c0cde51756022e	acl-06eacccc38cccd4ab	us-east-1f	172.31.64.0/20	-
<input type="checkbox"/>	-	subnet-06d7fa7f8dfa3f927	acl-06eacccc38cccd4ab	us-east-1a	172.31.32.0/20	-

Selected subnets

subnet-095d9bde1b5163fa X

Cancel **Save changes**

- Successfully associated subnet

You have successfully updated subnet associations for acl-0a097ed1f33b6f179 / NACL.

Network ACLs (1/3) [Info](#)

Find resources by attribute or tag

	Name	Network ACL ID	Associated with	Default	VPC ID	Inbound rules
<input type="checkbox"/>	-	acl-02da0ec66795b8564	-	No	ypc-05ddac7397bf4a7fd / Default	3 Inb
<input type="checkbox"/>	-	acl-06eacccc38cccd4ab	5 Subnets	Yes	ypc-05ddac7397bf4a7fd / Default	2 Inb
<input checked="" type="checkbox"/>	NACL	acl-0a097ed1f33b6f179	subnet-095d9bde1b5163fa	No	ypc-05ddac7397bf4a7fd / Default	1 Inb

acl-0a097ed1f33b6f179 / NACL

[Details](#) [Inbound rules](#) [Outbound rules](#) [Subnet associations](#) [Tags](#)

Details

Network ACL ID	Associated with	Default	VPC ID
acl-0a097ed1f33b6f179	subnet-095d9bde1b5163fa	No	ypc-05ddac7397bf4a7fd / Default

- Click on NACL created → Actions → edit inbound rule and add ssh rule (rule number denotes priority).

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the VPC.

Rule number	Type	Protocol	Port range	Source	Allow/Deny
22	SSH (22)	TCP (6)	22	0.0.0.0/0	Allow
+	All traffic	All	All	0.0.0.0/0	Deny

Add new rule **Sort by rule number**

Cancel **Preview changes** **Save changes**

- Create one instance and try to connect to terminal.(select subnet of same region).

The screenshot shows the AWS Management Console 'Launch wizard' for an EC2 instance. The 'Network' step is active, showing the following configuration:

- VPC:** vpc-05ddac7397bf4a7fd (Default) [172.31.0.0/16]
- Subnet:** subnet-095d9bde1b5163fa [VPC: vpc-05ddac7397bf4a7fd, Availability Zone: us-east-1c, IP addresses available: 4090, CIDR: 172.31.80.0/20]
- Auto-assign public IP:** Enable
- Firewall (security groups):** Create security group (selected)
- Security group name:** launch-wizard-4
- Description:** launch-wizard-4 created 2024-08-15T18:13:28.701Z

The 'Summary' panel on the right shows the instance configuration:

- Number of instances:** 1
- Software Image (AMI):** Amazon Linux 2023 AMI 2023.5.2...read more [ami-0ae8f15ae66fe8cda]
- Virtual server type (instance type):** t2.micro
- Firewall (security group):** New security group
- Storage (volumes):** 1 volume(s) - 8 GiB

A 'Free tier' notification is also visible: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which)'. The 'Launch instance' button is highlighted in orange.

- We can connect intance to terminal .

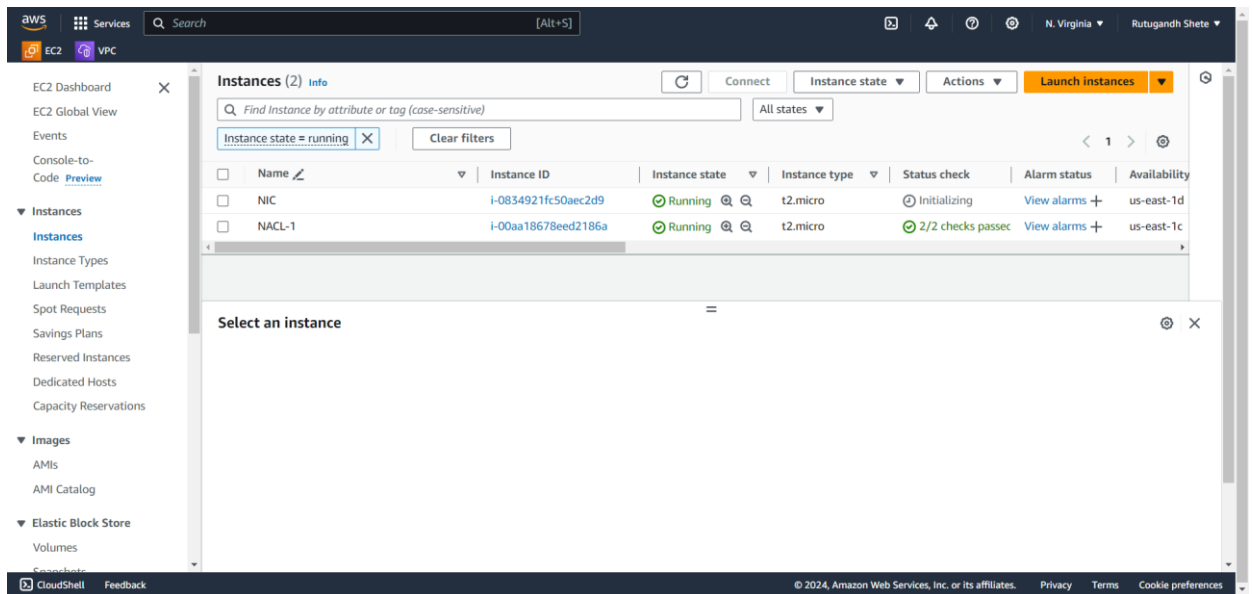
The screenshot shows the AWS CloudShell terminal with the following output:

```
64 bytes from 8.8.8.8: icmp_seq=80 ttl=117 time=0.699 ms
64 bytes from 8.8.8.8: icmp_seq=81 ttl=117 time=0.741 ms
64 bytes from 8.8.8.8: icmp_seq=82 ttl=117 time=0.685 ms
64 bytes from 8.8.8.8: icmp_seq=83 ttl=117 time=0.695 ms
64 bytes from 8.8.8.8: icmp_seq=84 ttl=117 time=0.715 ms
64 bytes from 8.8.8.8: icmp_seq=85 ttl=117 time=0.688 ms
64 bytes from 8.8.8.8: icmp_seq=86 ttl=117 time=0.705 ms
64 bytes from 8.8.8.8: icmp_seq=87 ttl=117 time=0.692 ms
64 bytes from 8.8.8.8: icmp_seq=88 ttl=117 time=0.698 ms
64 bytes from 8.8.8.8: icmp_seq=89 ttl=117 time=0.693 ms
64 bytes from 8.8.8.8: icmp_seq=90 ttl=117 time=0.669 ms
64 bytes from 8.8.8.8: icmp_seq=91 ttl=117 time=0.652 ms
64 bytes from 8.8.8.8: icmp_seq=92 ttl=117 time=0.691 ms
64 bytes from 8.8.8.8: icmp_seq=93 ttl=117 time=0.716 ms
64 bytes from 8.8.8.8: icmp_seq=94 ttl=117 time=0.704 ms
64 bytes from 8.8.8.8: icmp_seq=95 ttl=117 time=0.686 ms
64 bytes from 8.8.8.8: icmp_seq=96 ttl=117 time=0.715 ms
64 bytes from 8.8.8.8: icmp_seq=97 ttl=117 time=0.729 ms
64 bytes from 8.8.8.8: icmp_seq=98 ttl=117 time=0.687 ms
64 bytes from 8.8.8.8: icmp_seq=99 ttl=117 time=0.735 ms
64 bytes from 8.8.8.8: icmp_seq=100 ttl=117 time=0.717 ms
64 bytes from 8.8.8.8: icmp_seq=101 ttl=117 time=0.705 ms
64 bytes from 8.8.8.8: icmp_seq=102 ttl=117 time=0.712 ms
64 bytes from 8.8.8.8: icmp_seq=103 ttl=117 time=0.715 ms
64 bytes from 8.8.8.8: icmp_seq=104 ttl=117 time=0.709 ms
64 bytes from 8.8.8.8: icmp_seq=105 ttl=117 time=0.672 ms
64 bytes from 8.8.8.8: icmp_seq=106 ttl=117 time=0.682 ms
64 bytes from 8.8.8.8: icmp_seq=107 ttl=117 time=0.712 ms
```

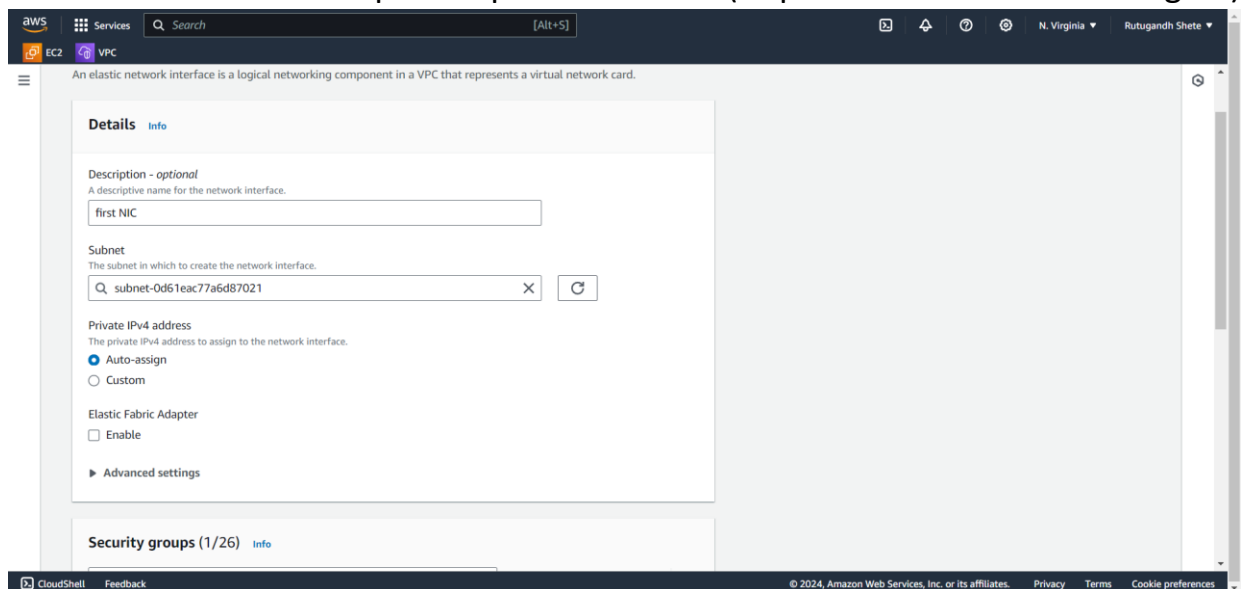
Network Interfaces (NIC)

Steps:

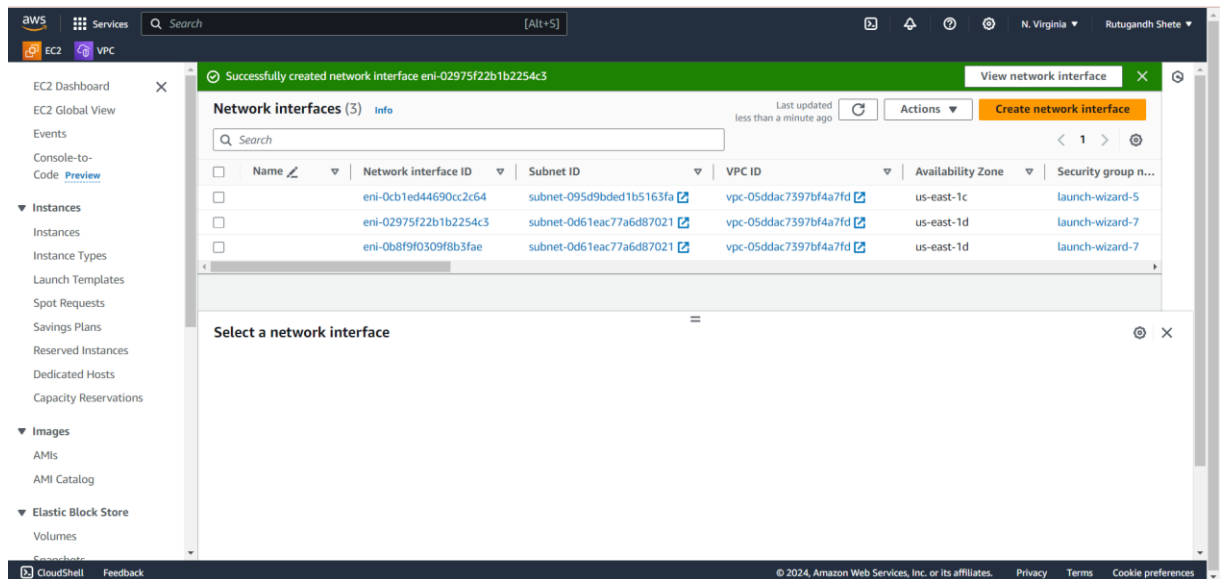
- Create one instance (private IP has one NIC).



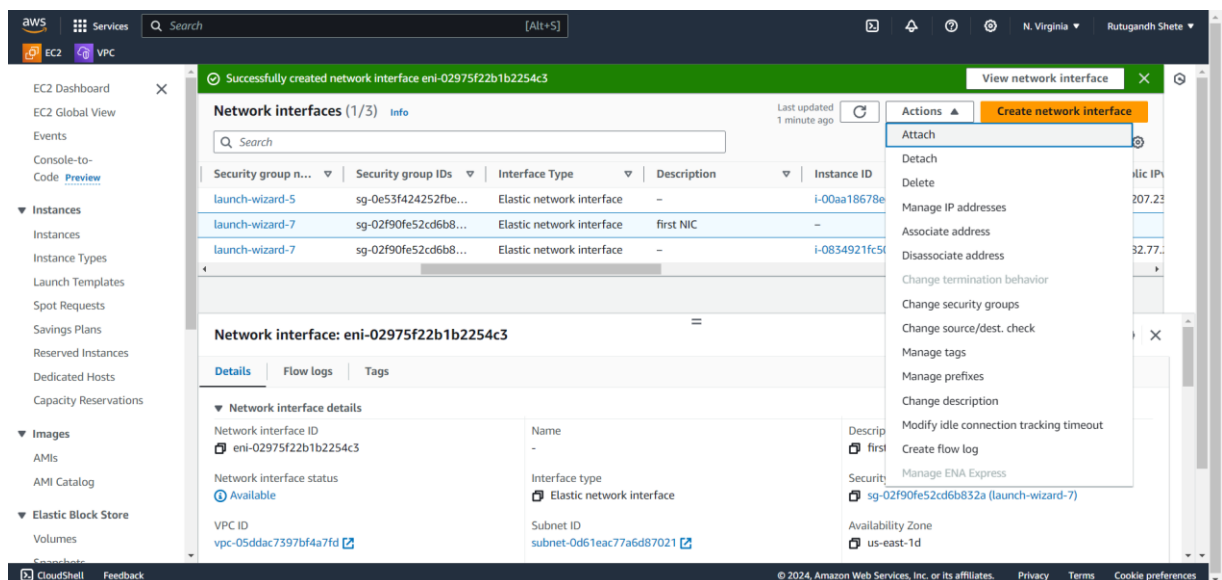
- Go to network and security → Network interfaces → create network interfaces → enter description → provide subnet (as per instance created in region).



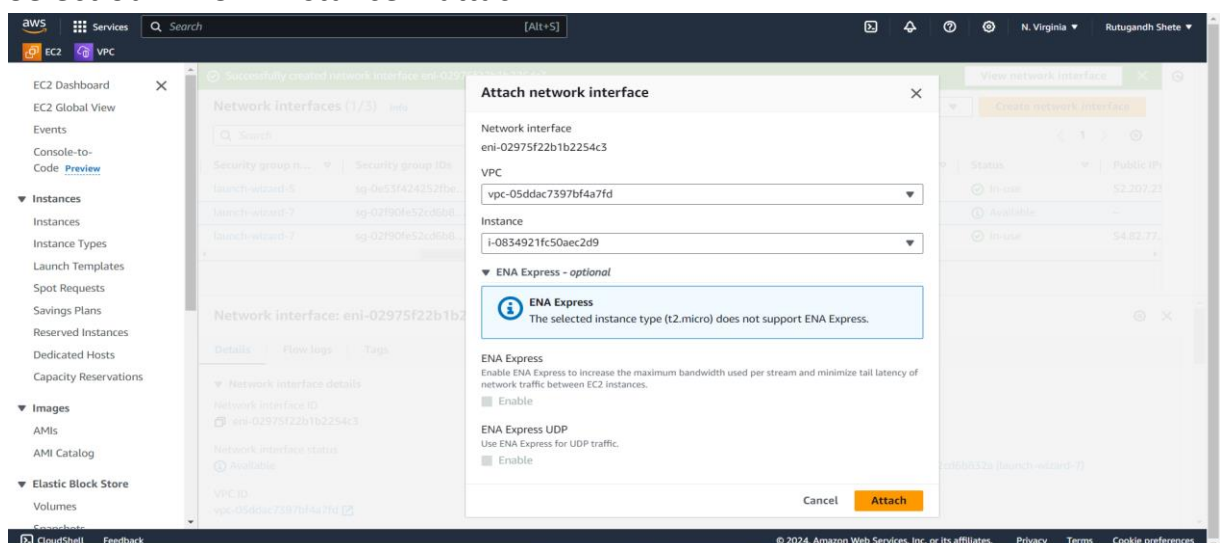
- Choose security group → create network interface.



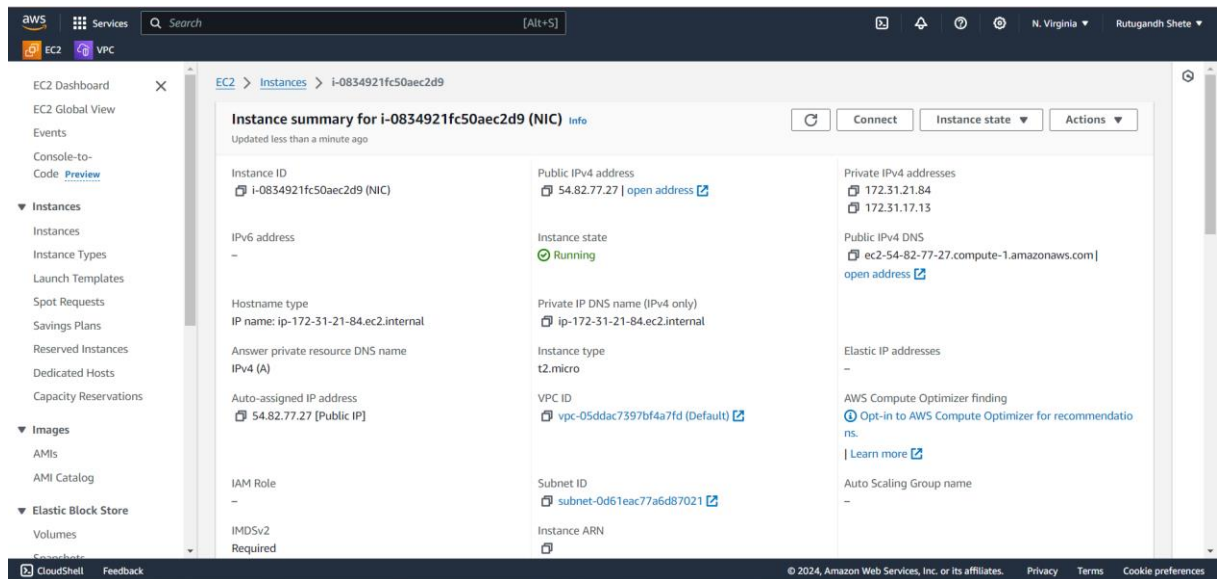
- Select our NIC → actions → attach



- Select our VPC → instance → attach



- Then go to our instance created and there will be 2 private IP address created.



Placement Group

To meet the needs of your workload, you can launch a group of *interdependent* EC2 instances into a *placement group* to influence their placement.

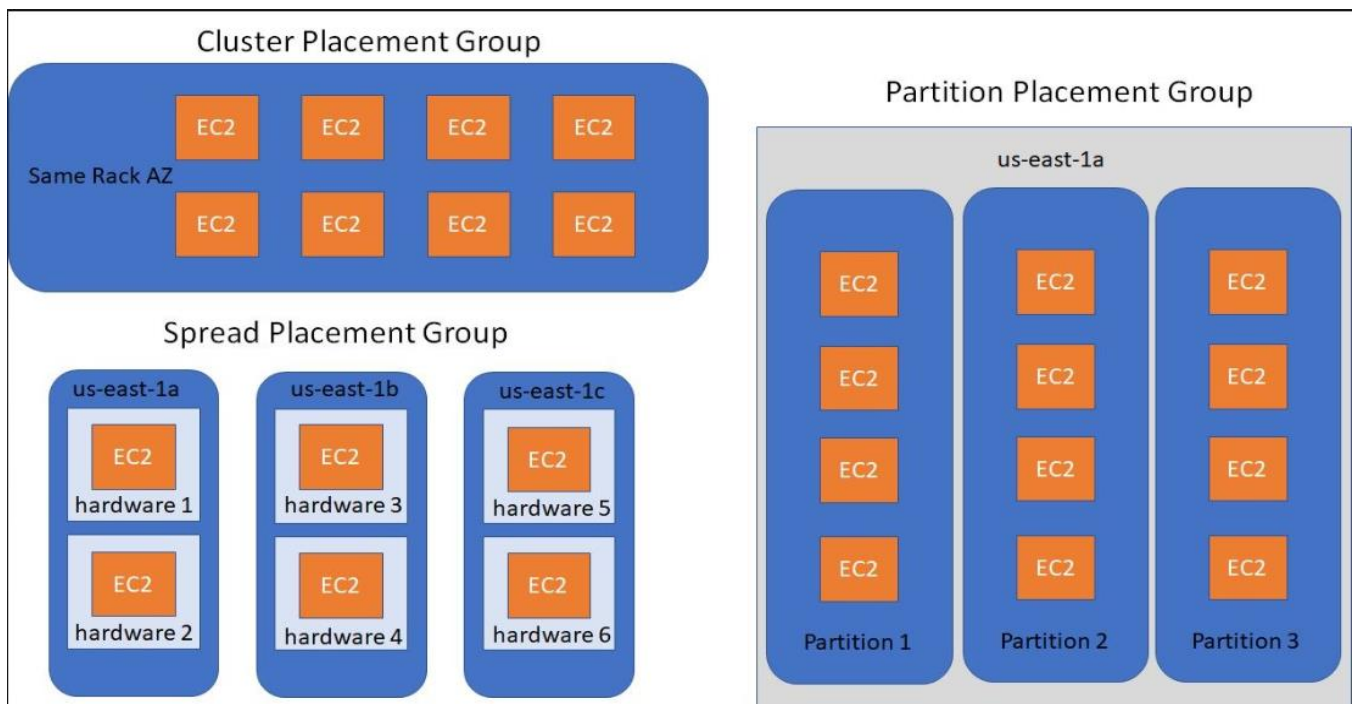
Depending on the type of workload, you can create a placement group using one of the following placement strategies:

- **Cluster** – Packs instances close together inside an Availability Zone. This strategy enables workloads to achieve the low-latency network performance necessary for tightly-coupled node-to-node communication that is typical of high-performance computing (HPC) applications.
- **Partition** – Spreads your instances across logical partitions such that groups of instances in one partition do not share the underlying hardware with groups of

instances in different partitions. This strategy is typically used by large distributed and replicated workloads, such as Hadoop, Cassandra, and Kafka.

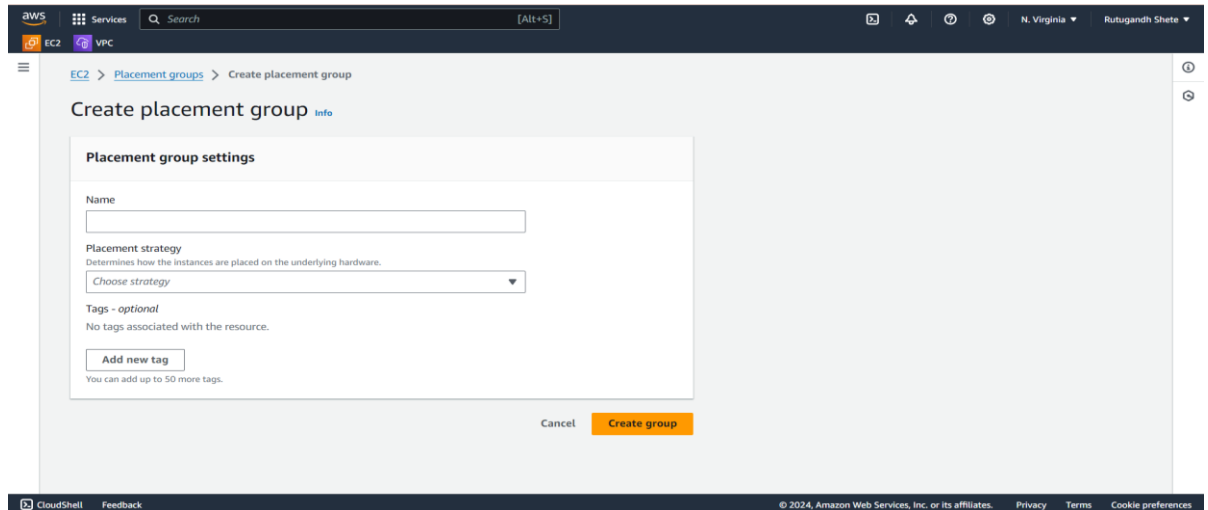
- **Spread** – Strictly places a small group of instances across distinct underlying hardware to reduce correlated failures.

Placement groups are optional. If you don't launch your instances into a placement group, EC2 tries to place the instances in such a way that all of your instances are spread out across the underlying hardware to minimize correlated failures.

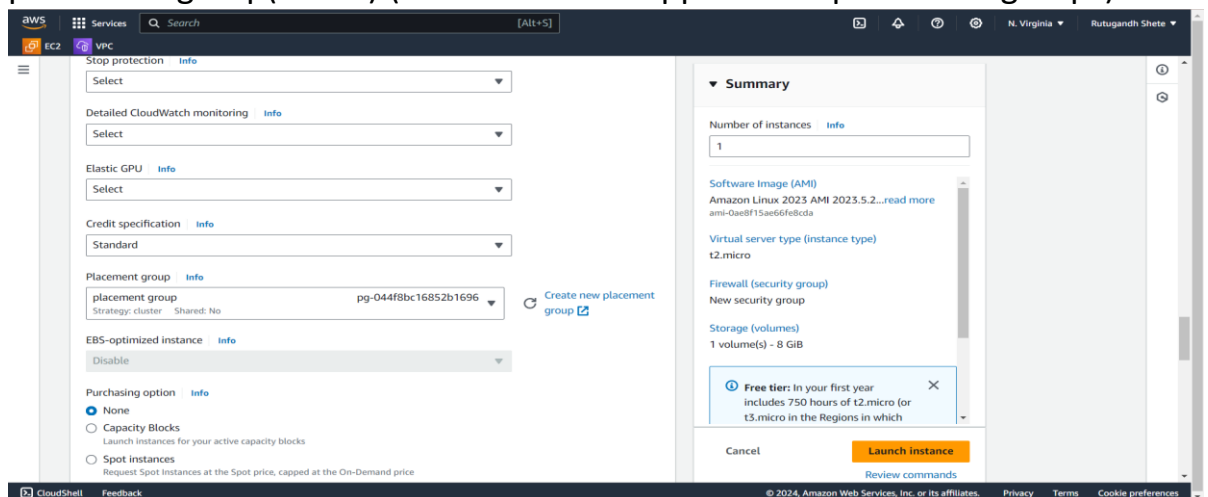


Steps:

- In EC2 dashboard go to Network and security and click on placement groups → create placement groups. Name → placement strategy → create



- Create one instance, while creating instance in Advance settings → select placement group(Name).(t2 micro is not supported to placement groups).



- Launch instance. While creating instances select our placement groups so that all instances are created in one placement group.

