

## Task – 16

### NACL and Security group

#### Creation of VPC

The screenshot shows the AWS VPC console interface. On the left, a sidebar navigation includes 'VPC dashboard', 'Virtual private cloud' (with 'Your VPCs' selected), 'Security', and 'CloudShell'. The main area displays 'Your VPCs (1/2) Info' with two entries: one for a default VPC and another for 'my\_vpc' which is currently selected. The 'my\_vpc' entry has a CIDR of 10.0.0.0/16. Below this, a detailed view for 'vpc-088da4e83bda295b1 / my\_vpc' is shown, including sections for 'Details', 'Resource map', 'CIDRs', 'Flow logs', 'Tags', and 'Integrations'. The 'Details' section provides specific configuration details like VPC ID, State, and DNS resolution.

#### Creation of Subnet

The screenshot shows the AWS VPC console interface, similar to the previous one but focused on subnets. The sidebar navigation includes 'VPC dashboard', 'Virtual private cloud' (with 'Your VPCs' selected), 'Subnets', 'Security', and 'CloudShell'. The main area displays 'Subnets (1/4) Info' with four entries, one of which is 'my\_pub\_subnet' which is currently selected. The 'my\_pub\_subnet' entry has a CIDR of 10.0.1.0/24. Below this, a detailed view for 'subnet-0c2b5472e4fdd5999 / my\_pub\_subnet' is shown, including sections for 'Details', 'Flow logs', 'Route table', 'Network ACL', 'CIDR reservations', 'Sharing', and 'Tags'. The 'Details' section provides specific configuration details like Subnet ID, State, and Availability Zone.

## Creation of Route table

The screenshot shows the AWS VPC Console interface. In the left sidebar, under the 'Route tables' section, a new route table is being created. The table is titled 'rtb-02f9a622f28118d62 / my\_rt'. It contains two routes: one for destination 0.0.0.0/0 targeting the internet gateway 'igw-0f9fa98b5f617c133' (status Active, propagated No) and another for destination 10.0.0.0/16 targeting 'local' (status Active, propagated No). The 'Details' tab is selected.

## Creation of Internet gateway

The screenshot shows the AWS VPC Console interface. In the left sidebar, under the 'Internet gateways' section, a new internet gateway is being created. The gateway is titled 'igw-0f9fa98b5f617c133 / igw1'. It is currently attached to the VPC 'my\_vpc'. The 'Details' tab is selected.

## Creation of security group

The screenshot shows the AWS VPC console with the 'Security Groups' page. A success message at the top states: "Security group (sg-0596412164b45d7fc | my\_sec\_grp) was created successfully". The main table lists two security groups:

| Name                 | Security group ID    | Security group name | VPC ID                | Description    |
|----------------------|----------------------|---------------------|-----------------------|----------------|
| sg-0596412164b45d7fc | sg-0596412164b45d7fc | my_sec_grp          | vpc-088da4e83bda295b1 | sg             |
| sg-044168cc1028c7e84 | sg-044168cc1028c7e84 | launch-wizard-1     | vpc-00aa32b792ae81dee | launch-wizard- |

The 'Details' tab is selected. Key details shown include:

- Security group name: my\_sec\_grp
- Security group ID: sg-0596412164b45d7fc
- Owner: 393827457998
- Inbound rules count: 2 Permission entries
- Description: sg
- VPC ID: vpc-088da4e83bda295b1
- Outbound rules count: 1 Permission entry

## Creation of NACL

Here it has the inbound rule as (all traffic: allow) that's why the web hosting happened

The screenshot shows the AWS VPC console with the 'Network ACLs' page. A specific Network ACL named 'acl-0de10ef26a5726fcf' is selected. The 'Details' tab is active, showing the following information:

| Associated with                          | Default | VPC ID                         |
|--|---------|--------------------------------|
| subnet-0c2b5472e4fdd5999 / my_pub_subnet | Yes     | vpc-088da4e83bda295b1 / my_vpc |

The 'Inbound rules' tab is selected, displaying two rules:

| Rule number | Type        | Protocol | Port range | Source    | Allow/Deny |
|-------------|-------------|----------|------------|-----------|------------|
| 100         | All traffic | All      | All        | 0.0.0.0/0 | Allow      |
| *           | All traffic | All      | All        | 0.0.0.0/0 | Deny       |

## Creation of Instance

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with sections like EC2, Instances, Images, and Elastic Block Store. The main area displays a table titled 'Instances (1/1) Info'. It shows one instance: 'my\_instance' (ID: i-0d3db0bcafe57337), which is 'Running' on an 't2.micro' instance type. The public IP is 54.255.193.105. The details tab is selected, showing the instance ID, public and private IPv4 addresses, instance state (Running), and public DNS.

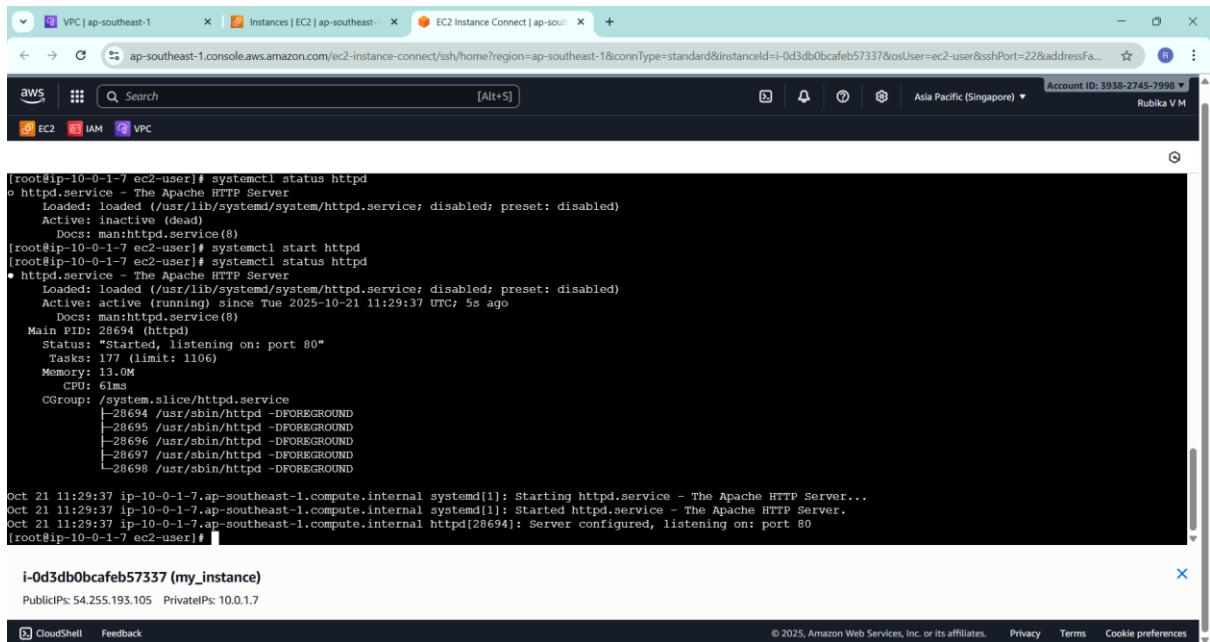
The screenshot shows an EC2 Instance Connect session. The terminal window displays the following text:

```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Tue Oct 21 11:26:30 2025 from 3.0.5.37
[ec2-user@ip-10-0-1-7 ~]$ sudo su
[root@ip-10-0-1-7 ec2-user]# yum install httpd -y
Last metadata expiration check: 0:00:36 ago on Tue Oct 21 11:28:24 2025.
Dependencies resolved.

      Package           Architecture   Version            Repository    Size
Installing:
  httpd              x86_64        2.4.65-1.amzn2023.0.1  amazonlinux  47 k
Installing dependencies:
  apr                x86_64        1:7.5-1.amzn2023.0.4  amazonlinux  129 k
  apr-util           x86_64        1:6.3-1.amzn2023.0.1  amazonlinux  98 k
  generic-logos-httpd x86_64        18.0.0-12.amzn2023.0.3  amazonlinux  19 k
  httpd-compat       x86_64        2.4.65-1.amzn2023.0.1  amazonlinux  1 M
  httpd-filesystem  x86_64        2.4.65-1.amzn2023.0.1  amazonlinux  13 k
  httpd-tools         x86_64        2.4.65-1.amzn2023.0.1  amazonlinux  81 k
  libxml2             x86_64        1.0.9-4.amzn2023.0.2  amazonlinux  315 k
```

At the bottom, it shows the public and private IP addresses: PublicIP: 54.255.193.105, PrivateIP: 10.0.1.7.



```
[root@ip-10-0-1-7 ec2-user]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Active: inactive (dead)
       Docs: man:httdp.service(8)
[root@ip-10-0-1-7 ec2-user]# systemctl start httpd
[root@ip-10-0-1-7 ec2-user]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Active: active (running) since Tue 2025-10-21 11:29:37 UTC; 5s ago
       Docs: man:httdp.service(8)
     Main PID: 28694 (httpd)
      Status: "Started, listening on: port 80"
        Tasks: 177 (limit: 1106)
       Memory: 13.0M
          CPU: 61ms
         CGroupl: /system.slice/httpd.service
           └─28694 /usr/sbin/httpd -DFOREGROUND
              ├─28695 /usr/sbin/httpd -DFOREGROUND
              ├─28696 /usr/sbin/httpd -DFOREGROUND
              ├─28697 /usr/sbin/httpd -DFOREGROUND
              └─28698 /usr/sbin/httpd -DFOREGROUND
Oct 21 11:29:37 ip-10-0-1-7.ap-southeast-1.compute.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Oct 21 11:29:37 ip-10-0-1-7.ap-southeast-1.compute.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Oct 21 11:29:37 ip-10-0-1-7.ap-southeast-1.compute.internal httpd[28694]: Server configured, listening on: port 80
[root@ip-10-0-1-7 ec2-user]#
```

i-0d3db0cafeb57337 (my\_instance)  
PublicIPs: 54.255.193.105 PrivateIPs: 10.0.1.7

**Here we get the targeted output of the webhosting when the inbound rule's all traffic is allowed.**



**It works!**

When the inbound rule is set to (http: deny) like the below screenshot

The screenshot shows the AWS VPC Network ACLs configuration page. The URL is [ap-southeast-1.console.aws.amazon.com/vpcconsole/home?region=ap-southeast-1#NetworkAclDetails:networkAclId=acl-0de10ef26a5726fcf](https://ap-southeast-1.console.aws.amazon.com/vpcconsole/home?region=ap-southeast-1#NetworkAclDetails:networkAclId=acl-0de10ef26a5726fcf). The page displays a success message: "You have successfully updated inbound rules for acl-0de10ef26a5726fcf". The Network ACL ID is "acl-0de10ef26a5726fcf". The Default setting is "Yes". The VPC ID is "vpc-088da4e83bda295b1 / my\_vpc". The Inbound rules section shows two rules:

| Rule number | Type        | Protocol | Port range | Source    | Allow/Deny |
|-------------|-------------|----------|------------|-----------|------------|
| 100         | HTTP (80)   | TCP (6)  | 80         | 0.0.0.0/0 | Deny       |
| *           | All traffic | All      | All        | 0.0.0.0/0 | Deny       |

We don't get the desired output of web hosting.

The screenshot shows a browser window with the address bar showing "54.255.193.105". The page content is as follows:

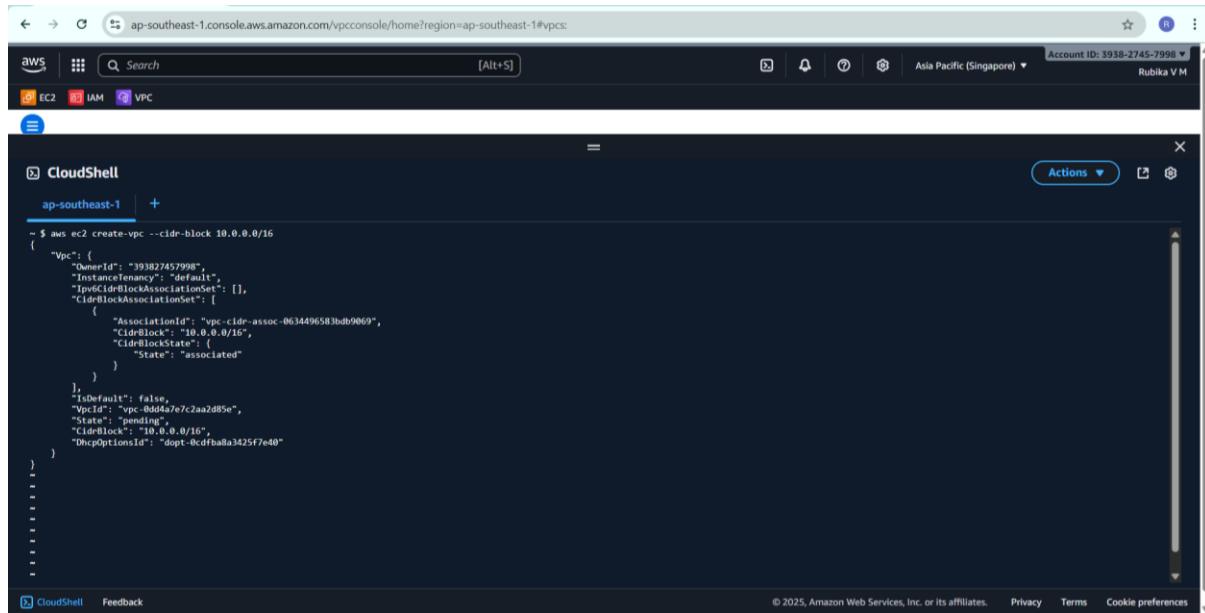
This site can't be reached  
54.255.193.105 took too long to respond.  
Try:

- Checking the connection
- Checking the proxy and the firewall

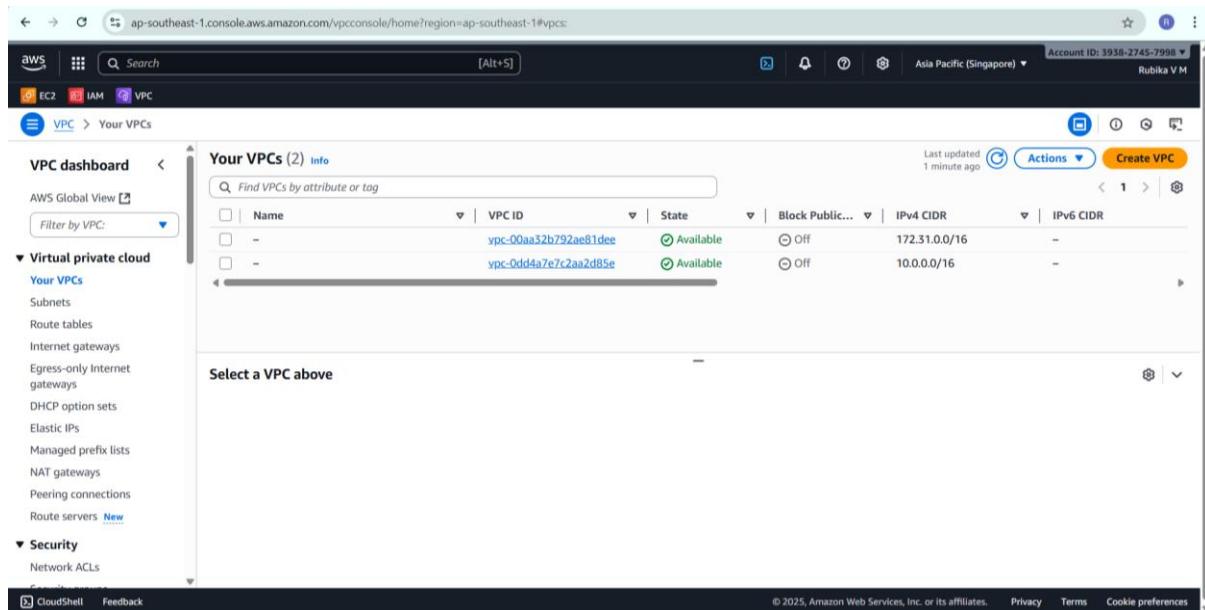
ERR\_CONNECTION\_TIMED\_OUT

Reload Details

## Create VPC through AWS CLI



```
$ aws ec2 create-vpc --cidr-block 10.0.0.0/16
{
  "Vpc": {
    "OwnerId": "393827457998",
    "InstanceTenancy": "default",
    "Ipv4CidrBlockAssociationSet": [],
    "CidrBlockAssociationSet": [
      {
        "AssociationId": "vpc-cidr-assoc-063496583bd9969",
        "CidrBlock": "10.0.0.0/16",
        "CidrBlockState": {
          "State": "associated"
        }
      }
    ],
    "IsDefault": false,
    "VpcId": "vpc-0dd4a7e7c2aa2d85e",
    "State": "pending",
    "CidrBlock": "10.0.0.0/16",
    "DhcpOptionsId": "dopt-0cdfba8a3425f7e40"
  }
}
```

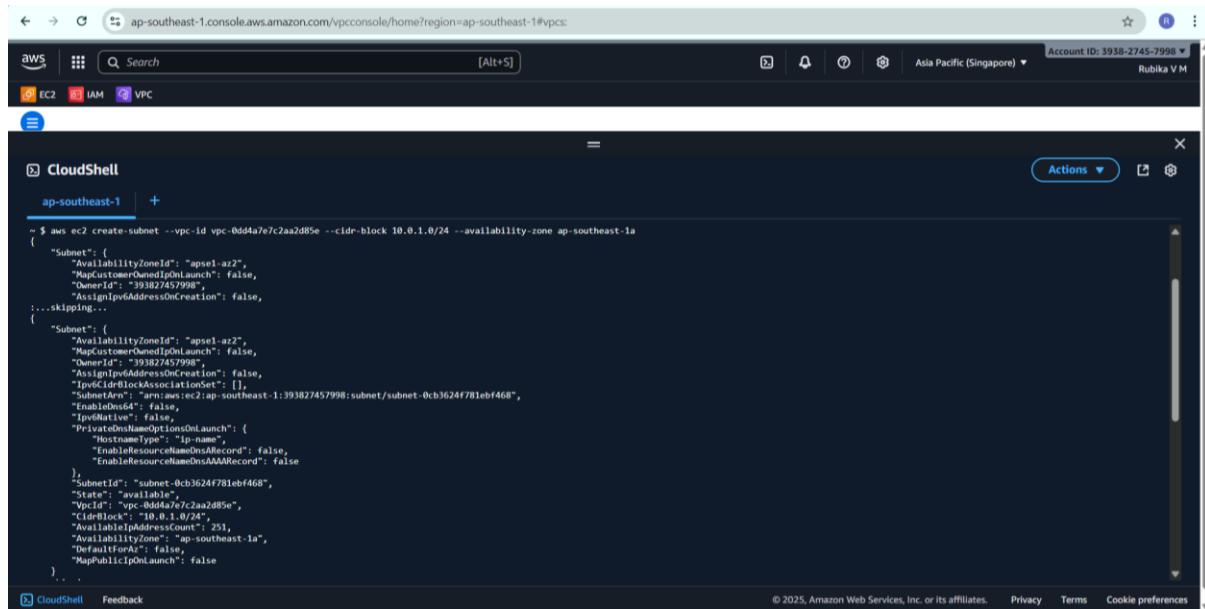


Your VPCs (2) [Info](#)

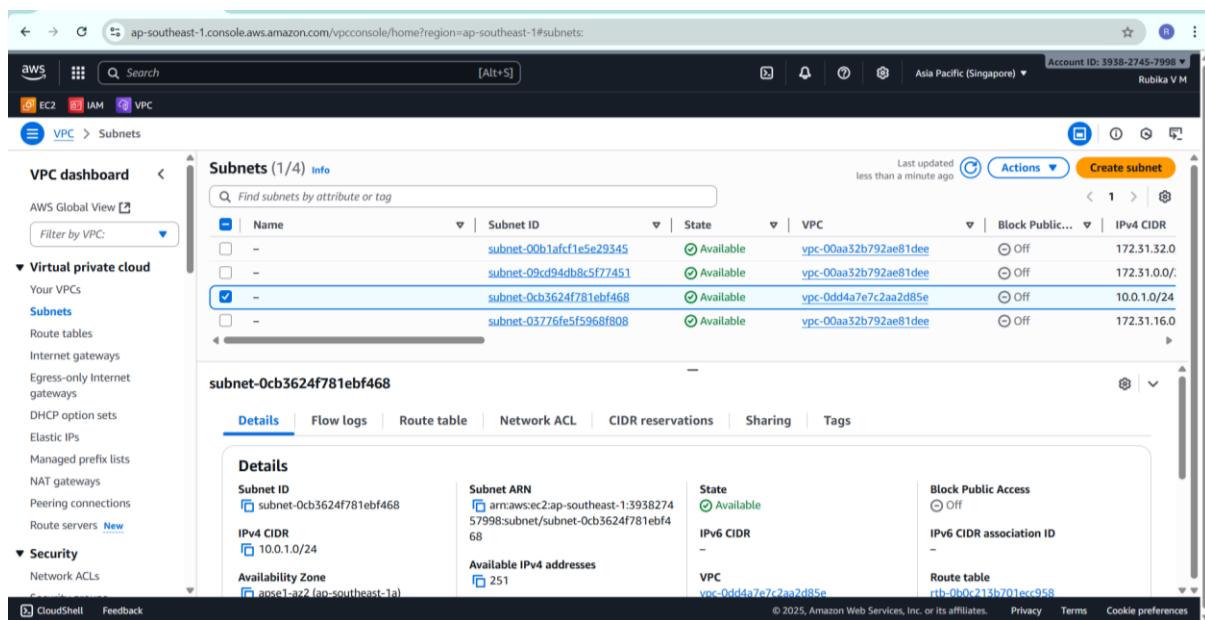
| Name | VPC ID                | State     | Block Public... | IPv4 CIDR     | IPv6 CIDR |
|------|-----------------------|-----------|-----------------|---------------|-----------|
| -    | vpc-0aa52b792ae81dee  | Available | Off             | 172.31.0.0/16 | -         |
| -    | vpc-0dd4a7e7c2aa2d85e | Available | Off             | 10.0.0.0/16   | -         |

Select a VPC above

## Creation of subnet through CLI



```
$ aws ec2 create-subnet --vpc-id vpc-0dd4a7e7c2aa2d85e --cidr-block 10.0.1.0/24 --availability-zone ap-southeast-1a
{
  "Subnet": {
    "AvailabilityZoneId": "apsel-az2",
    "MapCustomerOwnedIpOnLaunch": false,
    "OwnerId": "393827457998",
    "AssignIpv6AddressOnCreation": false,
    "...skipping...",
    "Subnet": {
      "AvailabilityZoneId": "apsel-az2",
      "MapCustomerOwnedIpOnLaunch": false,
      "OwnerId": "393827457998",
      "AssignIpv6AddressOnCreation": false,
      "Ipv4CidrBlockAssociationSet": [],
      "SubnetId": "subnet-0cb3624f781ebf468",
      "EnableDns64": false,
      "Ipv6Native": false,
      "PrivateDnsNameOptionsOnLaunch": {
        "HostType": "ip-name",
        "EnableResourceNameCollisionAvoidance": false,
        "EnableResourceNameAAARecord": false
      },
      "SubnetId": "subnet-0cb3624f781ebf468",
      "State": "available",
      "VpcId": "vpc-0dd4a7e7c2aa2d85e",
      "CidrBlock": "10.0.1.0/24",
      "AvailableIpAddressCount": 251,
      "AvailabilityZone": "ap-southeast-1a",
      "DefaultIpv6": false,
      "MapPublicIpOnLaunch": false
    }
  }
}
```



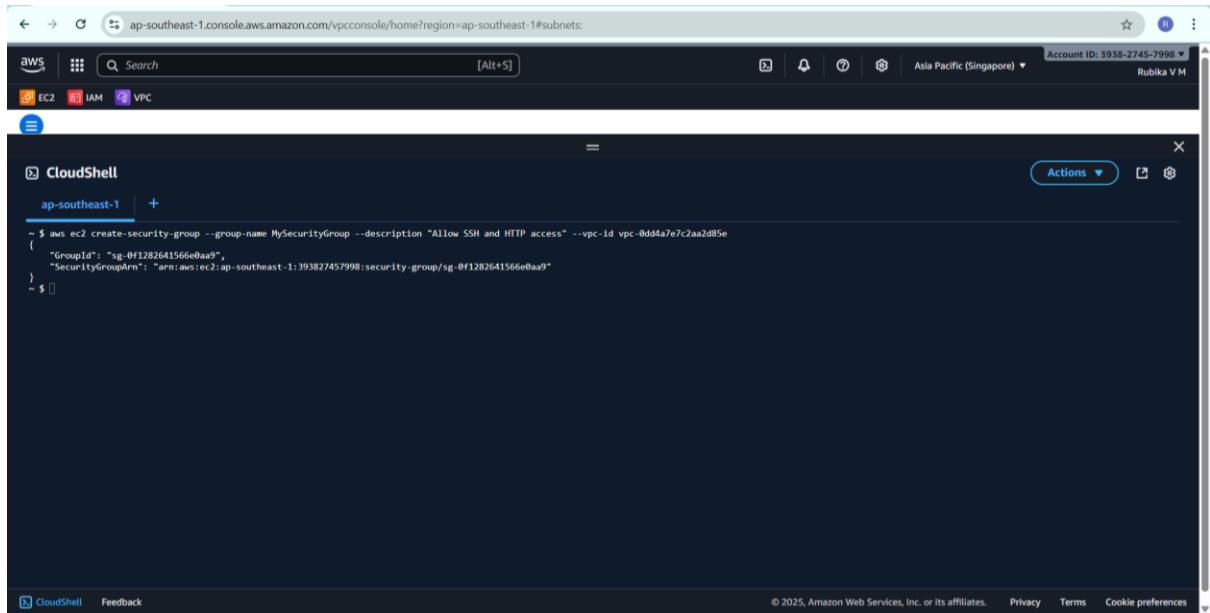
The screenshot shows the AWS VPC Subnets page. On the left, there's a sidebar with 'VPC dashboard' and sections for 'Virtual private cloud' (Your VPCs, Subnets), 'Internet gateways', 'Egress-only Internet gateways', 'DHCP option sets', 'Elastic IPs', 'Managed prefix lists', 'NAT gateways', 'Peering connections', and 'Route servers'. The 'Subnets' section is expanded, showing a table of subnets:

| Name                                  | Subnet ID                | State     | VPC                   | Block Public Access | IPv4 CIDR   |
|---------------------------------------|--------------------------|-----------|-----------------------|---------------------|-------------|
| -                                     | subnet-00b1afcf1e5e29345 | Available | vpc-0aa32b792ae81dee  | Off                 | 172.31.32.0 |
| -                                     | subnet-09cd94db8c5f77451 | Available | vpc-0aa32b792ae81dee  | Off                 | 172.31.0.0/ |
| <input checked="" type="checkbox"/> - | subnet-0cb3624f781ebf468 | Available | vpc-0dd4a7e7c2aa2d85e | Off                 | 10.0.1.0/24 |
| -                                     | subnet-03776fe5f5968fb08 | Available | vpc-0aa32b792ae81dee  | Off                 | 172.31.16.0 |

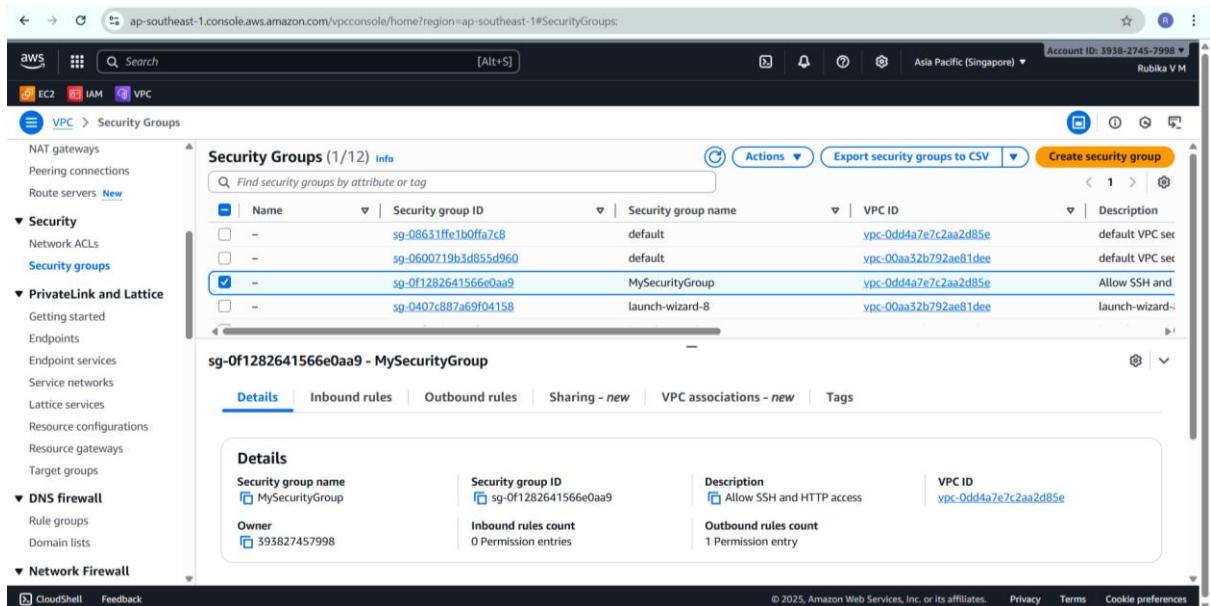
The subnet 'subnet-0cb3624f781ebf468' is selected. Below the table, the 'subnet-0cb3624f781ebf468' details are shown:

| Details  | Subnet ARN  | State                        | Block Public Access                  |
|--|---|------------------------------|--------------------------------------|
| Subnet ID<br>subnet-0cb3624f781ebf468            | arn:aws:ec2:ap-southeast-1:393827457998:subnet/subnet-0cb3624f781ebf468 | Available                    | Off                                  |
| IPv4 CIDR<br>10.0.1.0/24                         | -   | -                            | IPv6 CIDR association ID<br>-        |
| Availability Zone<br>apsel-az2 (ap-southeast-1a) | Available IPv4 addresses<br>251   | VPC<br>vpc-0dd4a7e7c2aa2d85e | Route table<br>rtb-0b0c213b701ecc958 |

## Creation of security group through CLI



```
$ aws ec2 create-security-group --group-name MySecurityGroup --description "Allow SSH and HTTP access" --vpc-id vpc-0dd4a7e7c2aa2d85e
{
  "GroupId": "sg-0f1282641566e0aa9",
  "SecurityGroupArn": "arn:aws:ec2:ap-southeast-1:393827457998:security-group/sig-0f1282641566e0aa9"
} $
```



The screenshot shows the AWS VPC console interface. On the left, there's a navigation sidebar with sections like NAT gateways, Peering connections, Route servers, Security (Network ACLs, Security groups), PrivateLink and Lattice, DNS firewall, and Network Firewall. The main area displays a table titled "Security Groups (1/12) Info". The table has columns for Name, Security group ID, Security group name, VPC ID, and Description. One row is selected, showing "sg-0f1282641566e0aa9" as the Name, "MySecurityGroup" as the Security group name, "vpc-0dd4a7e7c2aa2d85e" as the VPC ID, and "Allow SSH and HTTP access" as the Description. Below the table, a detailed view for "sg-0f1282641566e0aa9 - MySecurityGroup" is shown, with tabs for Details, Inbound rules, Outbound rules, Sharing - new, VPC associations - new, and Tags. The "Details" tab is active, displaying information such as Security group name (MySecurityGroup), Security group ID (sg-0f1282641566e0aa9), Owner (393827457998), Inbound rules count (0 Permission entries), Description (Allow SSH and HTTP access), and Outbound rules count (1 Permission entry).

## Changing port number:

The screenshot shows the AWS EC2 Instances page. A single instance, 'port\_instance' (i-09627ebf5879b38ab), is listed as 'Running'. The instance type is t2.micro and it is in the 'Initializing' status check. It is associated with the availability zone ap-southeast-1b and the public IP ec2-52-221-191-87.ap-southeast-1.amazonaws.com. The instance summary details show the Public IPv4 address 52.221.191.87 and the Private IPv4 address 172.31.22.161. The Public DNS is ec2-52-221-191-87.ap-southeast-1.amazonaws.com.

The screenshot shows an EC2 Instance Connect terminal session for the instance 'port\_instance'. The session is connected via SSH to the Public IP 52.221.191.87. The terminal window displays a root shell on Amazon Linux 2023. The user runs several commands: 'sudo su' to switch to root, 'nano /etc/ssh/sshd\_config' to edit the SSH configuration file, and 'systemctl restart sshd' to apply changes. The session ends with a command prompt for the root user.

```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-172-31-22-161 ~]$ sudo su
[root@ip-172-31-22-161 ec2-user]# nano /etc/ssh/sshd_config
[root@ip-172-31-22-161 ec2-user]# systemctl restart sshd
[root@ip-172-31-22-161 ec2-user]#
```

```
ec2-user@ip-172-31-22-161:~
```

```
vmrub@Rubiii MINGW64 ~
$ cd downloads

vmrub@Rubiii MINGW64 ~/downloads
$ ssh -i awspem.pem -p 24 ec2-user@52.221.191.87
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
,      #
~\_\ #####
~~ \####\ Amazon Linux 2023
~~  \###\ https://aws.amazon.com/linux/amazon-linux-2023
~~   \#/ /-->
~~   \~\ / \
~~ .-. / / \
~~ \m/ / \
Last login: Tue Oct 21 16:35:32 2025 from 157.51.19.131
[ec2-user@ip-172-31-22-161 ~]$ |
```