

## write the difference between the nacl and security grp With hands on ?

first create vpc subnet, routetable, and accociate subnet route the igw...

create security group and enable ssh http and https

The screenshot displays two browser windows side-by-side, both showing the AWS VPC console.

**Top Window (VPC Console):**

- Left Sidebar:** 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 (New).
- Table:** Your VPCs (2) Info

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
-	vpc-0674dd9ea90d0fea	Available	Off	172.31.0.0/16	-
-	vpc-05daced423d49f90b	Available	Off	10.0.0.0/16	-

Select a VPC above

**Bottom Window (Security Groups Console):**

- Left Sidebar:** 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 (New).
- Table:** Security Groups (1/3) Info

Name	Security group ID	Security group name	VPC ID	Description
-	sg-0f5a44fb1deca9529	default	vpc-05daced423d49f90b	default VPC security
-	sg-0708317b7d308379c	default	vpc-0674dd9ea90d0fea	default VPC security
<input checked="" type="checkbox"/>	sg-01fee3b31bdd539b9	fortask	vpc-05daced423d49f90b	fortask

sg-01fee3b31bdd539b9 - fortask

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-00fc15c54c8fd023a	IPv4	SSH	TCP	22
-	sgr-07eef6bde27ad17a7	IPv4	HTTP	TCP	80
-	sgr-07f6fba73b4288621	IPv4	HTTPS	TCP	443

In network ACLs allow the ssh http and https if its deny we don't get output

The screenshot shows the AWS VPC Network ACLs console. On the left, there's a navigation sidebar with sections like Virtual private cloud, Security (Network ACLs selected), and PrivateLink and Lattice. The main area displays 'Network ACLs (1/2)'. There are two entries:

Name	Network ACL ID	Associated with	Default	VPC ID	Inbound
acl-070161157657d1209	acl-070161157657d1209	subnet-07ddd8492941f07	Yes	vpc-05daced423d49f90b	3 Inbound
acl-01f0902f5688eb114	acl-01f0902f5688eb114	3 Subnets	Yes	vpc-0674dd9ea90d0feaf	2 Inbound

For the first entry, 'acl-070161157657d1209', the 'Inbound rules (3)' section is expanded, showing:

Rule number	Type	Protocol	Port range	Source	Allow/Deny
90	SSH (22)	TCP (6)	22	0.0.0.0/0	Allow
100	HTTP (80)	TCP (6)	80	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

You see the output



**It works!**

## Security group

Act as a firewall for

Associates AWS EC2 instance

Controls both inbound/outbound traffic at the instance level

You can secure your VPC instance using only SG

Supports allow rules only

it is statefull (Return traffic is automatically allowed, regardless of any rule).

Evaluates all rules before deciding whether to allow traffic.

Affirms only to the instance that it is associated with

Has separate rules for inbound and outbound traffic

A newly created SG denies all inbound traffic as default.

A newly created SG has an outbound rule that allows all outbound traffic as default. instance associated with a single VPC talk to each other unless you add rules allowing it.

## NACL

Act as a firewall associated Subnets

Controls both inbound and outbound traffic at subnet level

Network layer, one additional level of defense

Supports allow and deny rules

Stateless (Return traffic must be explicitly allowed by rule).

Evaluates rules in number order when deciding

whether to allow traffic starting with the lowest number rule.

Applies to all instances in the subnet if it is associated with

some.

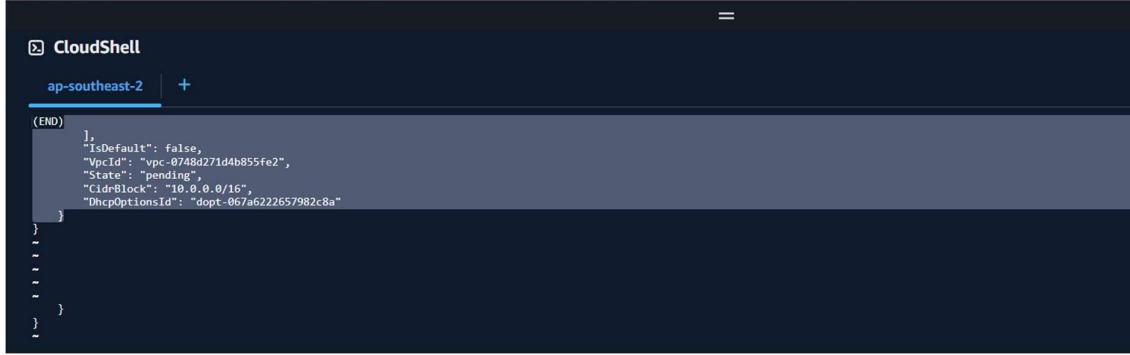
Some in NACL

A newly created NACL denies all outbound traffic as default

Each subnet in your VPC must be associated with a network ACL. If none is associated, the default NACL is selected.

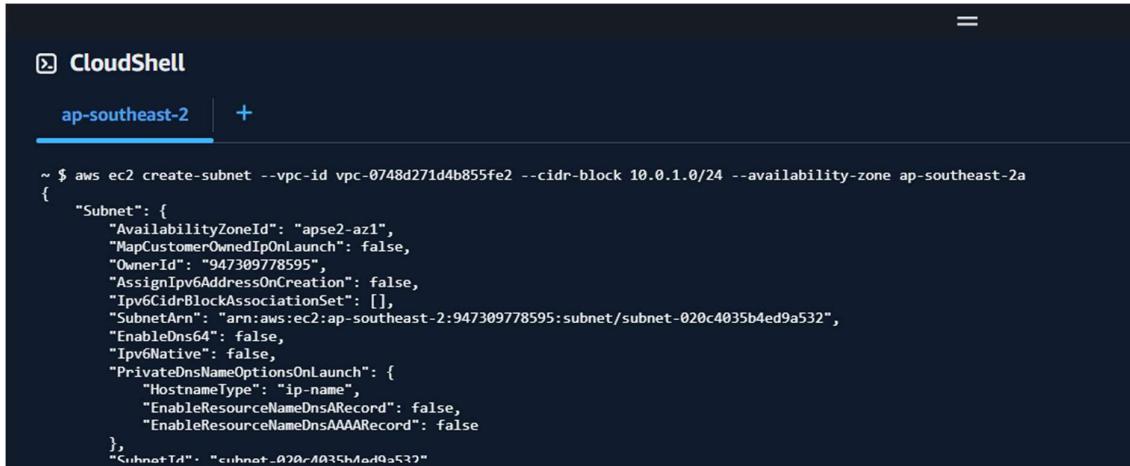
## create the VPC , Subnet , Security grp through AWS CLI

to create vpc: aws ec2 create-vpc --cidr-block 10.0.0.0/16



```
CloudShell
ap-southeast-2 + (END)
    ],
    "isDefault": false,
    "vpcId": "vpc-0748d271d4b855fe2",
    "state": "pending",
    "cidrBlock": "10.0.0.0/16",
    "dhcpOptionsId": "dopt-067a6222657982c8a"
}
{
}
{
}
{
}
{
}
```

To create subnet: aws ec2 create subnet --vpc-id id --cidr-block 10.0.1.0/24 --availability-zone ap-southeast-2a



```
CloudShell
ap-southeast-2 + (END)
~ $ aws ec2 create-subnet --vpc-id vpc-0748d271d4b855fe2 --cidr-block 10.0.1.0/24 --availability-zone ap-southeast-2a
{
    "Subnet": {
        "AvailabilityZoneId": "apse2-az1",
        "MapCustomerOwnedIpOnLaunch": false,
        "OwnerId": "947309778595",
        "AssignIpv6AddressOnCreation": false,
        "Ipv6CidrBlockAssociationSet": [],
        "SubnetArn": "arn:aws:ec2:ap-southeast-2:947309778595:subnet/subnet-020c4035b4ed9a532",
        "EnableDns64": false,
        "Ipv6Native": false,
        "PrivateDnsNameOptionsOnLaunch": {
            "HostnameType": "ip-name",
            "EnableResourceNameDnsARecord": false,
            "EnableResourceNameDnsAAAARecord": false
        },
        "SubnetId": "subnet-020c4035b4ed9a532"
    }
}
```

To create route table: aws ec2 create-route-table --vpc-id vpc-0182ff060926cc4ea

```
~ $  
~ $ aws ec2 create-route-table --vpc-id vpc-0748d271d4b855fe2  
{  
    "RouteTable": {  
        "Associations": [],  
        "PropagatingVgws": [],  
        "RouteTableId": "rtb-0169d32a9c86feb6b",  
        "Routes": [  
            {  
                "DestinationCidrBlock": "10.0.0.0/16",  
                "GatewayId": "local",  
                "Origin": "CreateRouteTable",  
                "State": "active"  
            }  
        ],  
        "Tags": [],  
        "VpcId": "vpc-0748d271d4b855fe2".
```

To create igw: aws ec2 create-internet-gateway

```
~ $ aws ec2 create-internet-gateway  
{  
    "InternetGateway": {  
        "Attachments": [],  
        "InternetGatewayId": "igw-0cd36bff4180150d6",  
        "OwnerId": "947309778595",  
        "Tags": []  
    }  
}  
~ $ █
```

To create sg: aws ec2 create-security-group --group-name sgwww --description "Allow SSH and HTTP access" --vpc-id vpc-0182ff060926cc4ea

```
~ $ aws ec2 create-security-group --group-name mysgpcli --description "Allow SSH and HTTP access" --vpc-id vpc-0748d271d4b855fe2  
{  
    "GroupId": "sg-02592d481073d0dda",  
    "SecurityGroupArn": "arn:aws:ec2:ap-southeast-2:947309778595:security-group/sg-02592d481073d0dda"  
}  
~ $ █
```

**Change the port number for the http as 82 .. and run the web hosting**

**sudo nano /etc/httpd/conf/httpd.conf**

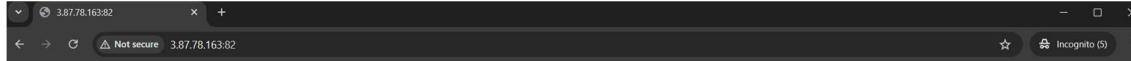
```
apr-1.7.5-1.amzn2023.0.4.x86_64
generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch
httpd-fiesystem-2.4.65-1.amzn2023.0.1.noarch
mailcap-2.1.49-3.amzn2023.0.3.noarch

apr-util-1.6.3-1.amzn2023.0.1.x86_64
httpd-2.4.65-1.amzn2023.0.1.x86_64
httpd-tools-2.4.65-1.amzn2023.0.1.x86_64
mod_http2-2.0.27-1.amzn2023.0.3.x86_64
apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
httpd-core-2.4.65-1.amzn2023.0.1.x86_64
libbrotli-1.0.9-4.amzn2023.0.2.x86_64
mod_lua-2.4.65-1.amzn2023.0.1.x86_64

Complete!
[root@ip-172-31-25-23 ec2-user]# sudo nano /etc/httpd/conf/httpd.conf
[root@ip-172-31-25-23 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:httpd.service(8)
[root@ip-172-31-25-23 ec2-user]# systemctl start httpd
[root@ip-172-31-25-23 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 Wed 2025-10-22 08:14:44 UTC; 15 ago
     Docs: man:httpd.service(8)
   Main PID: 25760 (httpd)
      Status: "Started, listening on: port 82"
        Tasks: 177 (limit: 1053)
       Memory: 13.3M
          CPU: 62ms
         CGroup: /system.slice/httpd.service
             ├─25760 /usr/sbin/httpd -DFOREGROUND
             ├─25761 /usr/sbin/httpd -DFOREGROUND
             ├─25762 /usr/sbin/httpd -DFOREGROUND
             ├─25764 /usr/sbin/httpd -DFOREGROUND

i-0fd25f5d9f12ff3f4 (linux)
PublicIPs: 3.87.78.165 PrivateIPs: 172.31.25.23
```

In sgp add 82 port and save you get the output



**It works!**