



## Day 13



# “CLOUD SECURITY”

### AWS Network ACLs (NACLs) to secure a VPC:

1. NACLs provide subnet-level, stateless firewall protection, requiring explicit rules for both inbound and outbound traffic.
2. They support both allow and deny rules, enabling fine-grained control over protocols, ports, and IPs—evaluated in order by rule number.
3. Best used alongside Security Groups for layered security; customize default NACLs to block unwanted traffic and mitigate DDoS attacks.

### Where can we find this NACL?

The screenshot shows the AWS CloudFront service page. At the top, there's a search bar with 'NACL' typed into it. On the left, there's a sidebar with 'Services' and 'Features' sections. Under 'Services', there's a card for 'VPC' with the subtext 'Isolated Cloud Resources'. Under 'Features', there's a card for 'Network ACLs' with the subtext 'VPC feature'. At the bottom, there's a button labeled 'Create web ACL' with the subtext 'WAF & Shield feature'.

### What is NACL?

A Network Access Control List (NACL) is a stateless, subnet-level firewall in AWS used to control inbound and outbound traffic for one or more subnets within a VPC.

### Why does NACL exist?

NACLs exist to provide:

- An additional layer of security in your AWS VPC.
- Fine-grained control over traffic at the network level, complementing Security Groups (which work at the instance level).
- Support for both allow and deny rules, which Security Groups do not offer (SGs only allow traffic).

This helps:

- Filter out unwanted traffic.
- Mitigate threats like DDoS, unauthorized access, or scanning.
- Ensure compliance with network-level access policies.

## Why and How is NACL associated with a VPC?

- Why: Every VPC automatically comes with a default NACL to control traffic at the subnet level.
- How: When you create a custom VPC, AWS creates a default NACL which you can customize. You can also create new NACLs and associate them with subnets in your VPC.

## Why and How is NACL associated with a Subnet?

- Why: NACLs operate at the subnet level, not at individual instance level. Each subnet in a VPC must be associated with exactly one NACL.
- How:
  - When you create a subnet, it is automatically associated with the VPC's default NACL.
  - You can change this by explicitly associating a different NACL with that subnet.
  - The same NACL can be reused across multiple subnets if desired.

## Creating a NACL:

Steps:

Open the dashboard of NACL:

The screenshot shows the AWS VPC Network ACLs dashboard. On the left, there is a sidebar with options like 'VPC dashboard', 'EC2 Global View', 'Virtual private cloud' (with 'Your VPCs', 'Subnets', 'Route tables', 'Internet gateways', and 'Egress-only internet' listed), and a 'Filter by VPC' dropdown. The main area is titled 'Network ACLs (2) info'. It contains a table with two rows of data:

Name	Network ACL ID	Associated with	Default	VPC ID	In
-	acl-048215e5962e62d86	3 Subnets	Yes	ypc-0a8b5c2749be9c3d1	2
-	acl-0c0c16beec66025d0	3 Subnets	Yes	ypc-0ad9338a47f6e65ce	2

Click on the “Create Network ACL” button: following screen will appear.

The screenshot shows the 'Create network ACL' configuration page. At the top, there is a breadcrumb navigation: 'VPC > Network ACLs > Create network ACL'. The main section is titled 'Create network ACL info' with the sub-instruction: 'A network ACL is an optional layer of security that acts as a firewall for controlling traffic in and out of a subnet.' Below this, there are two main sections: 'Network ACL settings' and 'Tags'.

**Network ACL settings**

- Name - optional**: A text input field containing 'my-acl-01'.
- VPC**: A dropdown menu labeled 'Select a VPC'.

**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.

No tags associated with the resource.

**Add tag**

You can add 50 more tags.

At the bottom right, there are 'Cancel' and 'Create network ACL' buttons.

You may add the details and create the NACL associated with a particular VPC.

## Key Reasons for NACL (even with VPC):

- Traffic Control at Subnet Level: NACLs apply before traffic even reaches EC2 or Security Groups. You can block malicious IPs or ports at subnet level.
- Support for Deny Rules: Unlike Security Groups, NACLs support explicit deny (e.g., block all traffic from a known bad IP).
- Stateless Filtering: Useful for fine-grained control of both inbound and outbound traffic — good for compliance and edge filtering.

## What are the various information we can get from the NACL?

The screenshot shows the AWS VPC Network ACL details page. The top navigation bar includes 'VPC' > 'Network ACLs' > 'acl-048215e5962e62d86'. The main title is 'acl-048215e5962e62d86'. On the left, there's a sidebar with 'Virtual private cloud' and 'Security' sections. The main content area shows 'Details' for the Network ACL, including its ID, association with 3 subnets, owner information, and VPC ID. Below this are tabs for 'Inbound rules', 'Outbound rules', 'Subnet associations', and 'Tags'. The 'Inbound rules' tab is selected, showing a table with two entries:

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

## **Think of It Like This:**

- **VPC** = City boundary
- **Subnet** = Neighborhood
- **NACL** = Police checkpoint at the neighborhood entrance
- **Security Group** = Security guard at the individual house (instance)

## Best practices of NACL:

### **1. Principle of Least Privilege**

- Only allow the minimum traffic needed for a subnet to function.
- Deny all unnecessary ports and IP ranges.

### **2. Configure Both Inbound and Outbound Rules**

- NACLs are stateless, so for every allowed inbound rule, a corresponding outbound rule must also be created.
  - e.g., Allow HTTP in → Allow ephemeral ports out (1024–65535).

### **3. Use Rule Numbering Wisely**

- Rules are evaluated from lowest to highest number.
- Keep DENY rules at lower numbers to take precedence over ALLOW if needed.
- Use spacing (e.g., 100, 110, 120...) to make room for future rules.

### **4. Deny Known Malicious IP Ranges**

- If you know malicious IPs or geographies, explicitly deny them at the NACL level.

### **5. Use Custom NACLs Instead of Default**

- Default NACL allows all traffic.
- Replace with custom NACLs to apply security best practices.

## **6. Combine NACLs with Security Groups**

- NACL = subnet-level, Security Group = instance-level
- Use both together for layered defense.

## **7. Separate Public & Private Subnets**

- Apply restrictive NACLs to private subnets.
- Public subnets can be less strict, but should still block unnecessary ports (e.g., Deny port 22 from 0.0.0.0/0).

## **8. Enable VPC Flow Logs**

- Use Flow Logs to monitor traffic going through subnets and validate whether NACLs are working as expected.

## **9. Use Descriptive Rule Comments (in console or IaC)**

- When using tools like Terraform or CloudFormation, comment your rules clearly for better audit and management.

## **10. Review & Audit Regularly**

- Periodically review NACLs to:
  - Remove unused rules.
  - Update ranges (e.g., if IP ownership changes).
  - Ensure they reflect current application needs.

--The End--