Case Study: Implementing AWS NACLs in a 3-Tier Web Application Architecture (Production Environment)

Objective

Design and implement secure, subnet-level access control using AWS NACLs for a production-grade 3-tier architecture Which consist:

1. Web Tier (Public Subnet)
2. Application Tier (Private Subnet)
3. Database Tier (Private Subnet)

Each tier is logically and physically separated using dedicated subnets. NACLs will be configured to control **inbound/outbound traffic** at the subnet level, in conjunction with **Security Groups** at the instance level.

Overview of AWS NACL (Network Access Control List)

What is a NACL?

An AWS Network Access Control List (NACL) is a stateless firewall at the **subnet level** that controls traffic moving in and out of one or more subnets within a Virtual Private Cloud (VPC).

Key Characteristics:

1. **Stateless**: Unlike security groups, NACLs do not automatically allow return traffic. Both inbound and outbound rules must be explicitly defined.
2. **Rule Evaluation**: Rules are evaluated in **ascending order** by rule number, and the first match determines the outcome (allow or deny).
3. **Default Behavior**:
    a. Default NACL: Allows all traffic.
    b. Custom NACL: Denies all traffic unless explicitly allowed.
4. **Applies to All Resources**: All EC2 instances and services inside the subnet are subject to NACL rules.

How NACLs Enhance Security

- Adds a **network-layer firewall** on top of instance-level security groups.
- Helps enforce **segmentation** and isolation between tiers.
- Supports **IP-level whitelisting or blacklisting** for broad access control.
- Useful for **compliance-driven** environments and defense-in-depth strategies.
- Allows for **fast mitigation** of network threats (e.g., blocking a malicious IP across a subnet).

NACL Configuration per Tier

1. Web Tier – Public Subnet

Purpose:

This subnet hosts public-facing web servers (e.g., EC2 with Apache/Nginx). These servers must accept traffic from the internet on HTTP/HTTPS ports.

Inbound NACL Rules:

Rule #	Protocol	Port Range	Source	Action	Description
100	TCP	80	0.0.0.0/0	ALLOW	Allow inbound HTTP traffic from the internet
110	TCP	443	0.0.0.0/0	ALLOW	Allow inbound HTTPS traffic from the internet
120	TCP	22	x.x.x.x/32	ALLOW	Allow SSH access only from a trusted admin IP
130	TCP	1024-65535	0.0.0.0/0	ALLOW	Allow return traffic (ephemeral ports for responses)
*	ALL	ALL	0.0.0.0/0	DENY	Deny all other inbound traffic

Explanation:

- Ports 80 and 443 are required for web servers.
- SSH access should be tightly controlled using trusted IPs.
- Ephemeral ports (1024–65535) allow return communication for stateless behavior.
- All other traffic is denied.

2. Application Tier – Private Subnet

Purpose:

Hosts the core business logic. This tier communicates with the Web Tier and the Database Tier but should not be accessible from the internet.

Inbound NACL Rules:

Rule #	Protocol	Port Range	Source	Action	Description
100	TCP	8080	Web Subnet CIDR	ALLOW	Allow application traffic from Web Tier
110	TCP	1024-65535	Web Subnet CIDR	ALLOW	Allow return traffic from the Web Tier
120	TCP	22	x.x.x.x/32	ALLOW	Optional: SSH access from bastion/management IP
*	ALL	ALL	0.0.0.0/0	DENY	Deny all other inbound traffic

Explanation:

- Port 8080 (or any custom app port) is exposed only to the Web tier.
- Return traffic from Web Tier is enabled via ephemeral ports.
- SSH is restricted and should ideally be routed via a bastion host.

3. Database Tier – Private Subnet

Purpose:

Hosts RDS or EC2-based database servers (e.g., MySQL, PostgreSQL). Should only accept traffic from the Application Tier.

Inbound NACL Rules:

Rule #	Protocol	Port Range	Source	Action	Description
100	TCP	3306	App Subnet CIDR	ALLOW	Allow MySQL traffic from Application Tier
110	TCP	1024-65535	App Subnet CIDR	ALLOW	Allow return traffic from Application Tier
*	ALL	ALL	0.0.0.0/0	DENY	Deny all other inbound traffic

Explanation:

- Port 3306 is used for MySQL; modify if using another DB engine.
- Only application tier should be able to connect to the DB.
- No external or web tier traffic should ever reach this subnet.

NACL Outbound Rules for a 3-Tier Web Application (Production)

NACLs are stateless, so outbound rules are as important as inbound ones. Outbound rules determine how resources in a subnet can send traffic to other subnets or the internet.

1. Web Tier – Public Subnet (Web Servers)

Purpose:

Web servers initiate responses to users and may occasionally make outbound requests (e.g., update checks, logging, API calls).

Outbound NACL Rules:

Rule #	Protocol	Port Range	Destination	Action	Description
100	TCP	80	0.0.0.0/0	ALLOW	Allow HTTP requests to the internet
110	TCP	443	0.0.0.0/0	ALLOW	Allow HTTPS requests to the internet
120	TCP	1024–65535	Web Client IPs	ALLOW	Allow return traffic to clients via ephemeral ports
*	ALL	ALL	0.0.0.0/0	DENY	Deny all other outbound traffic

Explanation:

- Allows communication with external APIs or services.
- Ensures that responses to users reach them over ephemeral ports.
- All other traffic is denied to prevent unwanted egress.

2. Application Tier – Private Subnet (App Servers)

Purpose:

Application servers send requests to the DB tier and may need to communicate with the Web tier or other internal services.

Outbound NACL Rules:

Rule #	Protocol	Port Range	Destination	Action	Description
100	TCP	3306	DB Subnet CIDR	ALLOW	Allow MySQL traffic to the DB tier
110	TCP	1024–65535	Web Subnet CIDR	ALLOW	Allow response traffic to the Web Tier
120	TCP	80/443	Specific APIs/IPs	ALLOW	Allow app server to reach approved external/internal APIs
*	ALL	ALL	0.0.0.0/0	DENY	Deny all other outbound traffic

Explanation:

- Ensures the app can connect to DBs and external/internal APIs only.
- Response traffic to Web tier is enabled.
- Strict outbound control minimizes risk from compromised app servers.

3. Database Tier – Private Subnet (DB Servers)

Purpose:

Databases rarely initiate connections. Only responses to application servers are allowed.

Outbound NACL Rules:

Rule #	Protocol	Port Range	Destination	Action	Description
100	TCP	1024–65535	App Subnet CIDR	ALLOW	Allow response traffic to application tier
*	ALL	ALL	0.0.0.0/0	DENY	Deny all other outbound traffic

Explanation:

- Databases are not allowed to initiate external communication.
- Return traffic to application tier is necessary for stateless connections.
- Ensures complete isolation from internet and other subnets.

Alignment with Professional Standards

Security Standard	Application
Zero Trust Architecture	Enforces strict subnet-level access control; trust no external source by default
Defense in Depth	Adds a layer of protection beyond instance-level security groups
Principle of Least Privilege	Grants only the minimum necessary network access for each tier
Compliance (e.g., PCI, HIPAA)	Facilitates network segmentation and audit-friendly configurations
Operational Readiness	Enables quick isolation of subnets during incidents or threats

NACL Best Practices

Do's and Don'ts for Configuring AWS NACLs

Do's

- Segment tiers (Web, App, DB) into separate subnets and apply tier-specific NACLs.
- Allow only necessary ports and IPs based on the role of each subnet.
- Include ephemeral port ranges (1024–65535) in outbound/inbound rules to support return traffic for TCP sessions.
- Apply a final DENY ALL rule (*) to block any unintended traffic.
- Use narrow CIDR blocks (e.g., App subnet range instead of 0.0.0.0/0) for precise access control.
- Maintain up-to-date NACL rules aligned with application or infrastructure changes.
- Monitor traffic using AWS tools like VPC Flow Logs and AWS Config for auditing and analysis.

Don'ts

- Do not place all application tiers in a single subnet or reuse a generic NACL across the entire VPC.
- Do not allow unrestricted access using 0.0.0.0/0 unless absolutely necessary and justified.
- Do not forget that NACLs are stateless — you must allow both inbound and outbound directions explicitly.
- Do not permit outbound internet access from Database or private App subnets unless specifically required.
- Do not overcomplicate NACLs with overlapping or redundant rules; keep them clean and purposeful.
- Do not depend solely on NACLs for security — combine with Security Groups for defense-in-depth.

Conclusion: AWS NACLs for 3-Tier Architecture

In a production-grade 3-tier application (Web, App, DB), Network ACLs (NACLs) serve as a crucial stateless security layer at the subnet level. When configured correctly, they help enforce a least-privilege communication model and act as a first line of defense alongside Security Groups.

By segmenting each tier into separate subnets and applying custom NACLs:

- The Web Tier can safely communicate with the internet and upstream layers.
- The Application Tier only interacts with the Web Tier and DB Tier as needed.
- The Database Tier remains fully isolated from external networks, allowing only trusted App Tier traffic.

Through precise Inbound and Outbound rules:

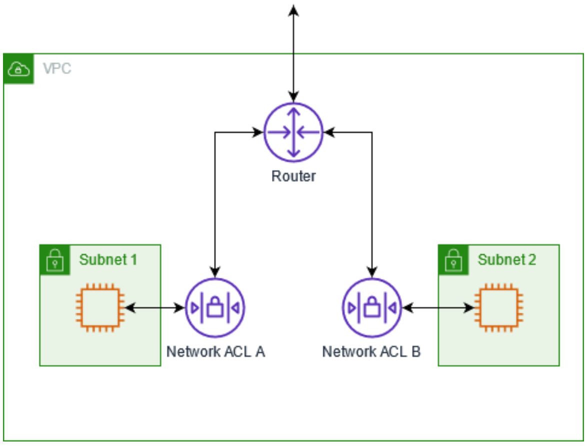
- Unnecessary traffic is explicitly denied, reducing attack surface.
- Critical traffic paths are explicitly allowed, ensuring functionality without overexposure.
- Ephemeral port ranges (1024–65535) are carefully handled to support return traffic for TCP connections.

This approach not only aligns with Zero Trust principles but also:

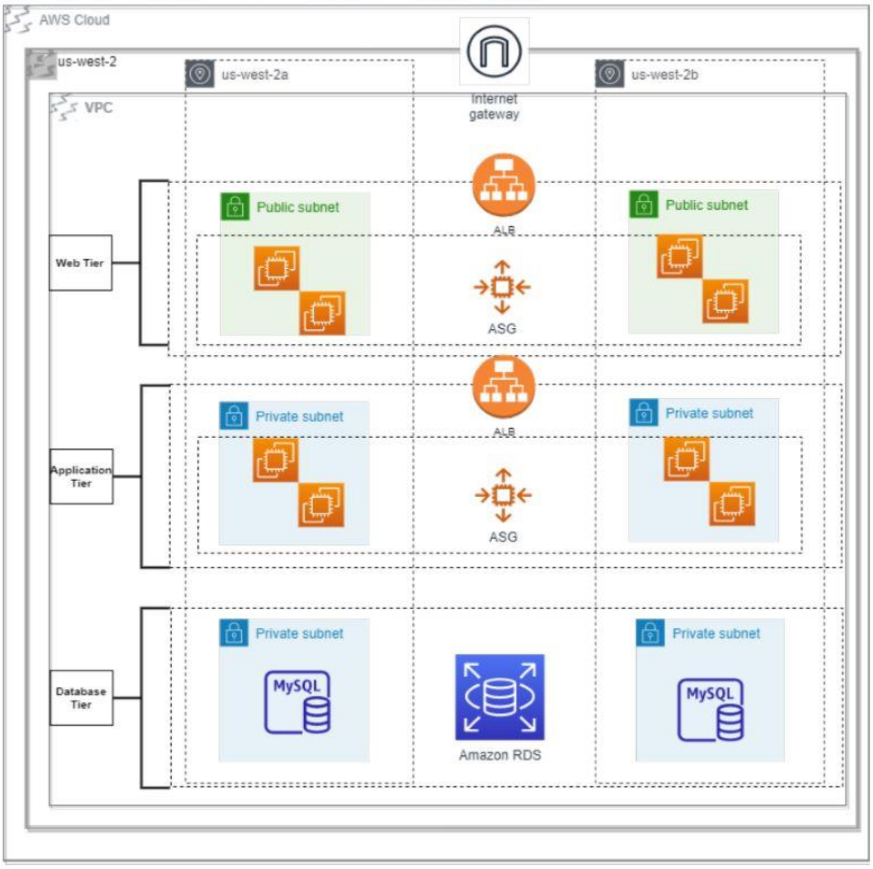
- Enhances network segmentation,
- Ensures compliance with industry security standards (like PCI-DSS or HIPAA),
- And supports auditing and operational clarity.

In essence, NACLs complement Security Groups by controlling traffic at the subnet border, offering a highly controlled and layered security posture for your AWS production environment.

Sample Diagram:



Three-Tier Architecture for AWS Cloud Infrastructure


AWS Cloud
us-west-2
us-west-2a
us-west-2b
Internet gateway
VPC
Web Tier
Public subnet
ALB
ASG
Public subnet
Application Tier
Private subnet
ALB
ASG
Private subnet
Database Tier
Private subnet
MySQL
Amazon RDS
Private subnet
MySQL