Enterprise Secure Network Architecture on AWS.

1. Problem Statement

A high-availability e-commerce platform required a secure network architecture ensuring public-facing services remained accessible while backend systems remained isolated. The organization required strong segmentation and monitoring controls.

I designed this architecture to support high availability, strong network segmentation, least privilege access, and continuous security monitoring.

2. Objectives

The primary objectives of this implementation were to:

- Isolate application and database resources from the public internet
- Enforce strong network segmentation
- Support high availability across Availability Zones
- Minimize attack surface
- Enable secure internal communication
- Maintain continuous monitoring and audit visibility

3. VPC and Subnet Architecture

3.1 VPC Design

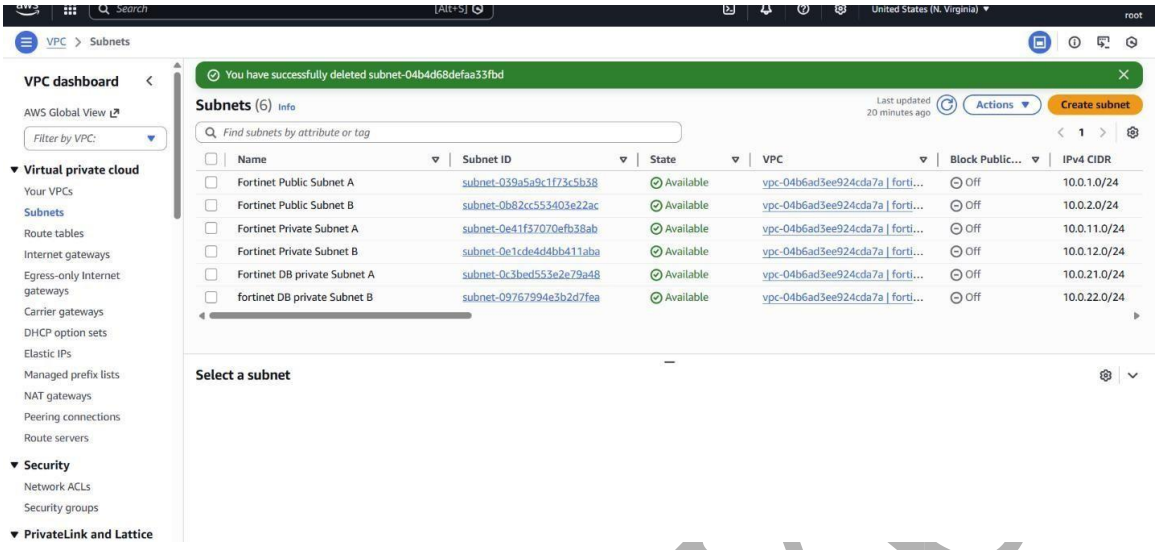
I created a Virtual Private Cloud (VPC) with a CIDR range of 10.0.0.0/16 to support long term scalability.

3.2 Subnet Layout

Six subnets were created across two Availability Zones:

- Two public subnets for internet-facing resources
- Two private subnets for application workloads

- Two private subnets for database workloads



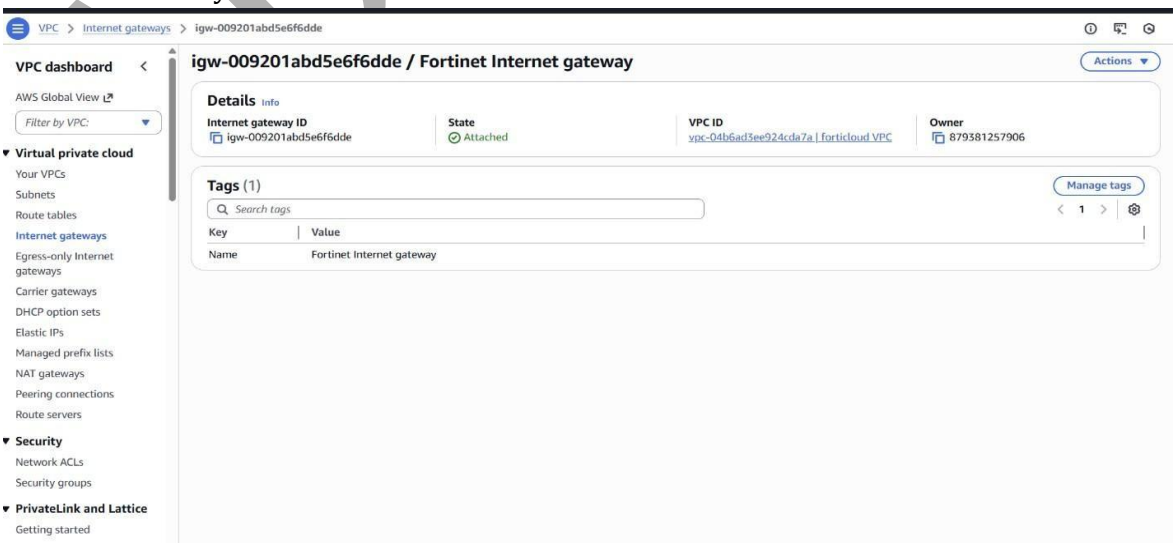

Each Availability Zone contains:

- One public subnet
- One private application subnet
- One private database subnet

This design ensures fault tolerance and strong isolation between tiers.

4. Internet Gateway and Routing

An Internet Gateway was created and attached to the VPC.

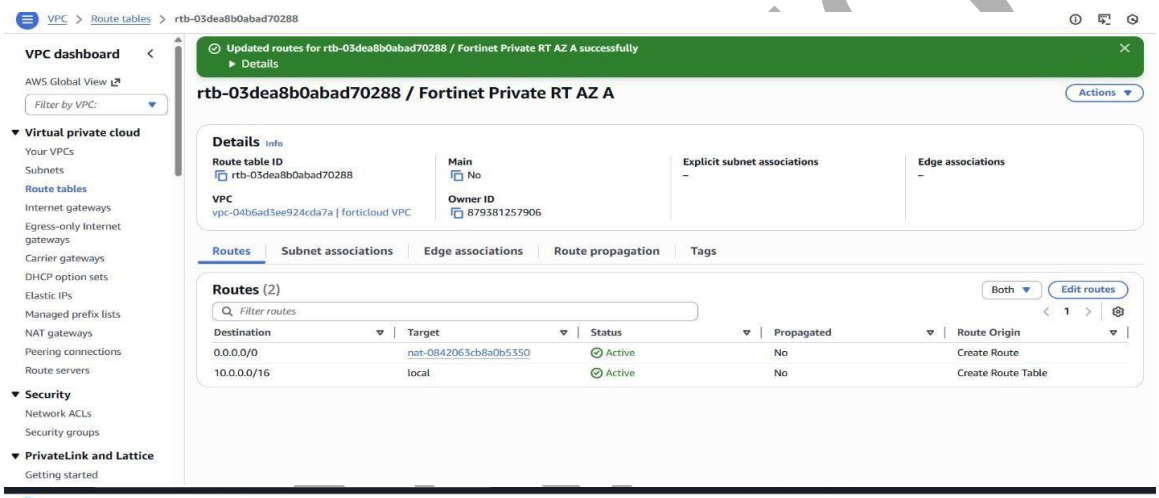


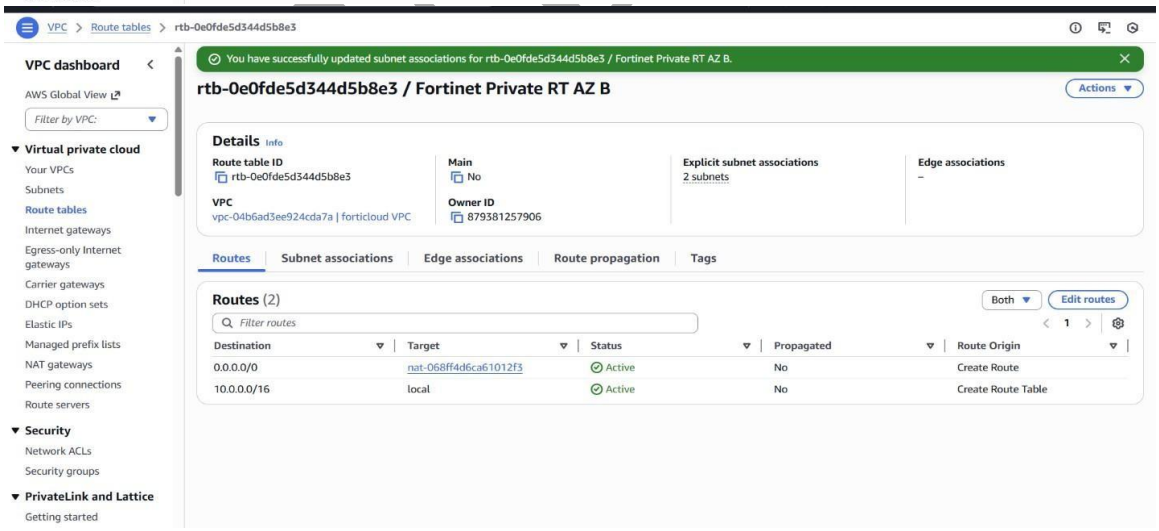
Public route tables were configured to direct outbound internet traffic to the Internet Gateway and associated with public subnets.

This ensured that only approved public resources had direct internet access.

5. NAT Gateway Configuration

I deployed a NAT Gateway in each Availability Zone and routed private subnet traffic to the local NAT Gateway.




This design:

- Prevented cross-AZ dependencies

- Improved fault isolation
- Increased availability
- Followed AWS best practices

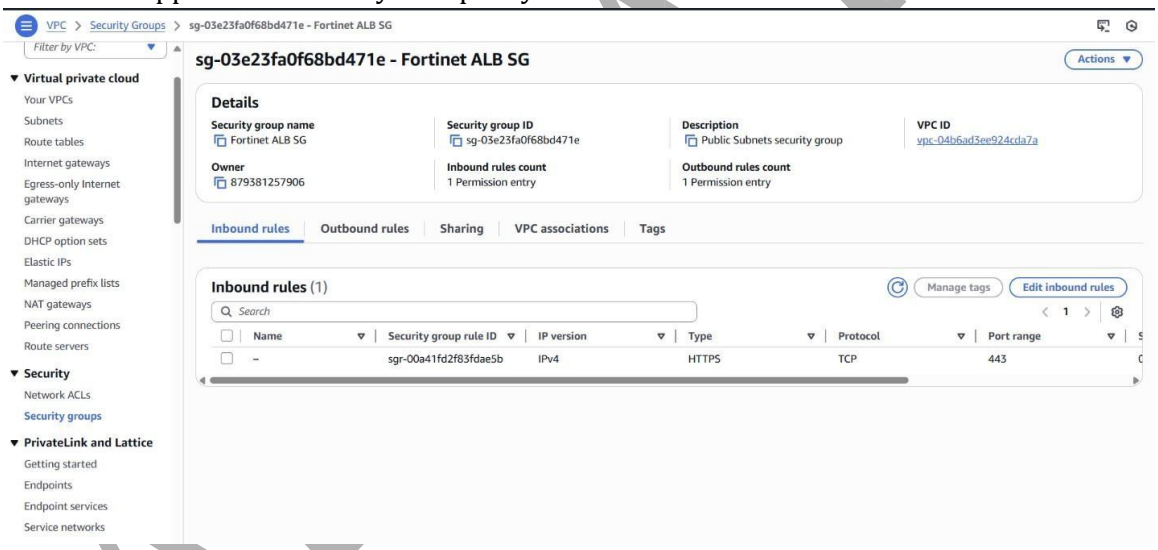
Private resources were able to access required external services without being directly exposed.

6. Security Group Architecture

Security groups were implemented for each architectural tier.

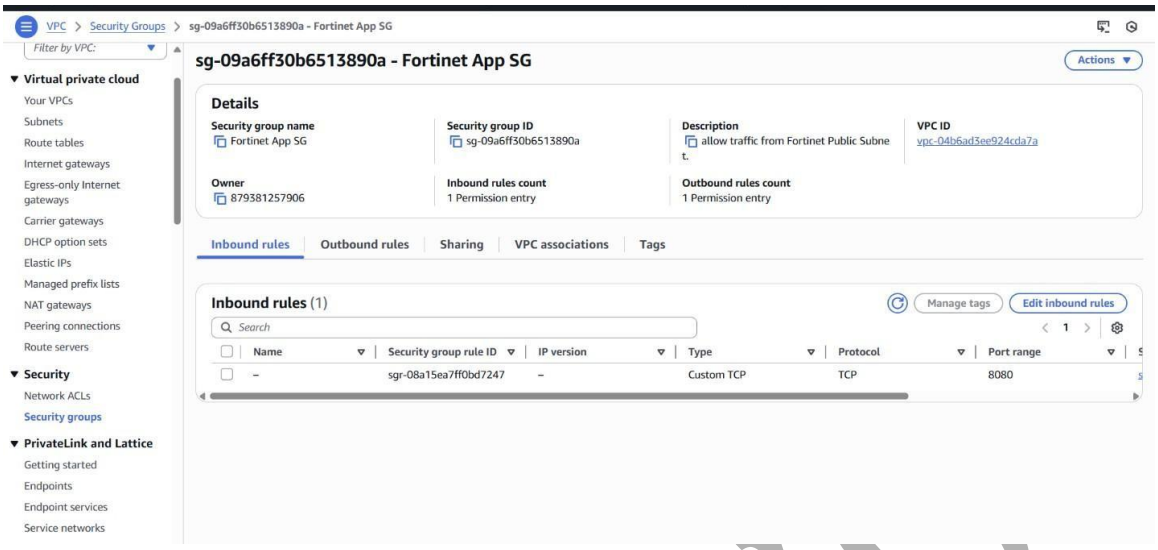
6.1 Load Balancer Security Group

- Inbound: HTTPS (443) from 0.0.0.0/0
- Outbound: Application Security Group only

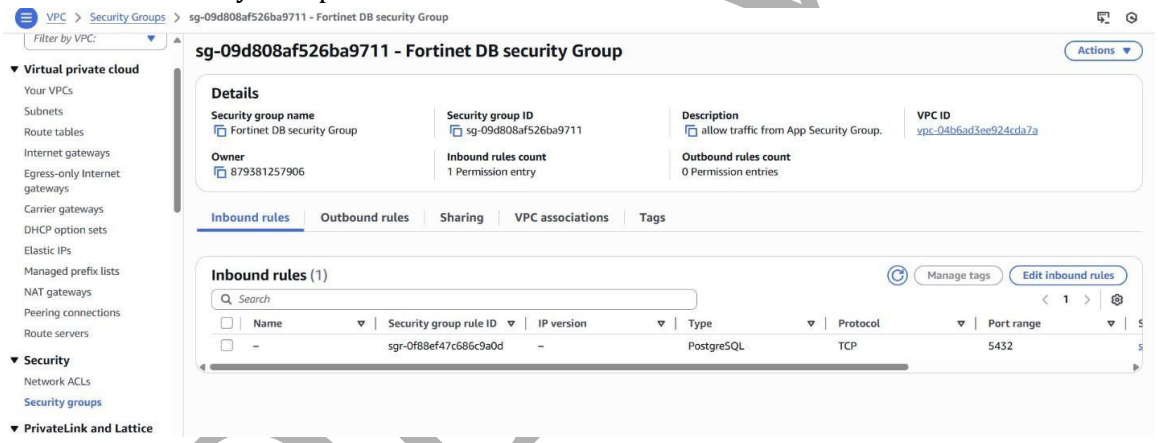



6.2 Application Security Group

- Inbound: Application traffic from ALB Security Group
- Outbound: PostgreSQL (5432) to Database Security Group




6.3 Database Security Group




- Inbound: PostgreSQL (5432) from Application Security Group
- Outbound: None

This structure enforced strict tier-to-tier communication and least privilege.

7. Network Access Control Lists (NACLs)

Network ACLs were configured to provide an additional layer of subnet-level protection.

7.1 Public Subnet NACL

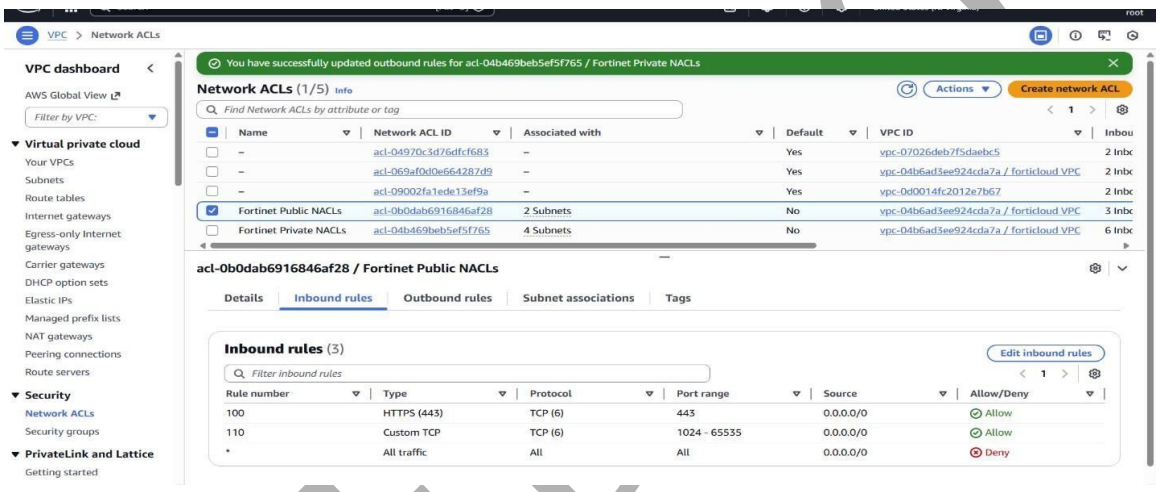
Inbound:

- HTTPS (443) from internet
- Ephemeral ports (1024–65535)

Outbound:

- HTTPS (443)
- Ephemeral ports

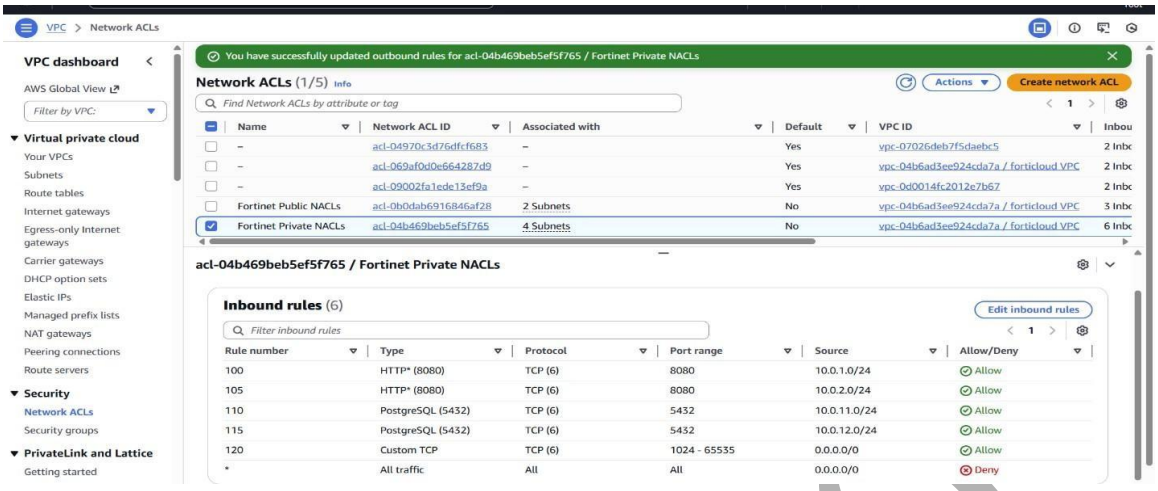
All other traffic was denied.




7.2 Private Subnet NACL

Inbound:

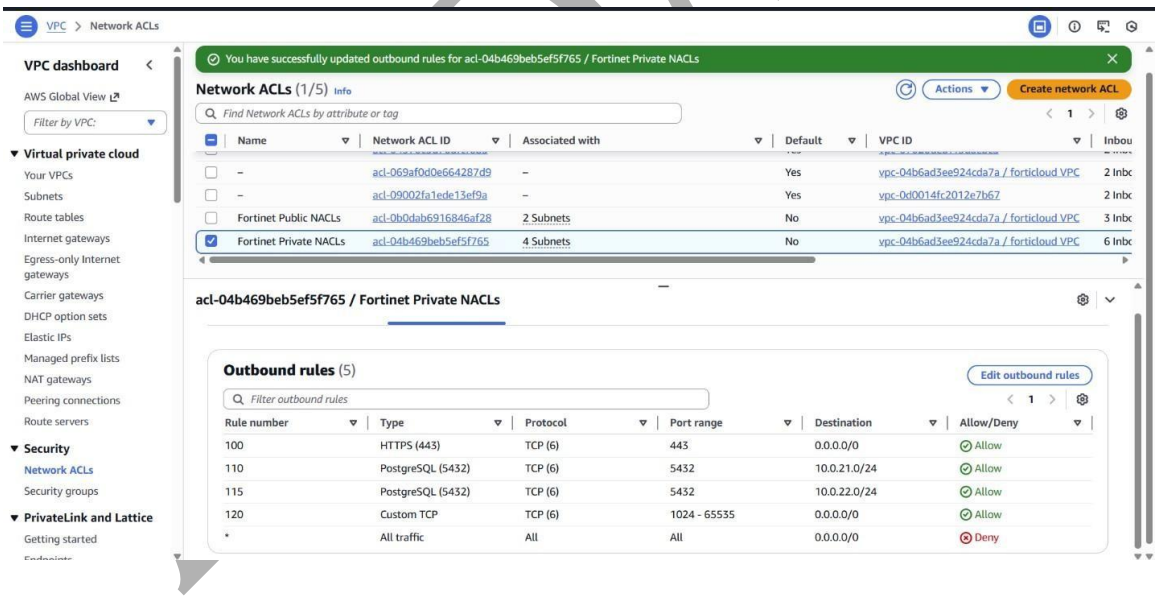
- Application traffic from public subnets
- PostgreSQL from application subnets
- Ephemeral ports




Outbound:

- HTTPS via NAT Gateway
- Database communication
- Ephemeral ports

All other traffic was denied.




This configuration enforced hard boundaries between network zones.

8. Compute and Database Deployment

EC2 instances were deployed in private application subnets with restricted security groups.

An RDS PostgreSQL database was deployed in private database subnets using a dedicated subnet group.

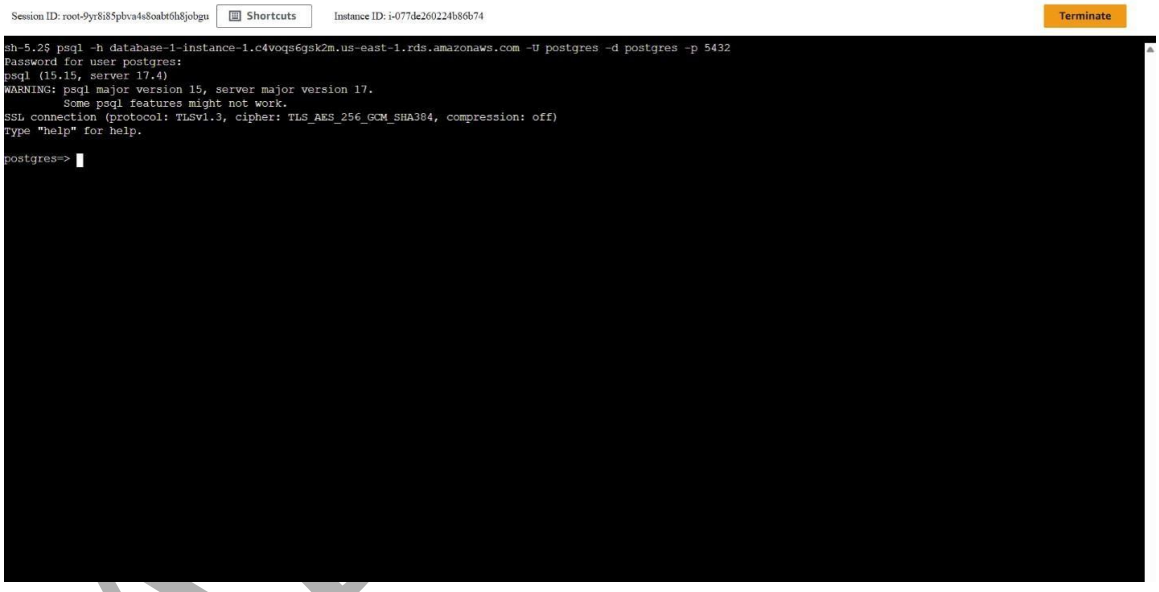
No database resources were exposed to the public internet.

9. Secure Connectivity Testing

Connectivity testing was performed using AWS Session Manager.

An IAM role was attached to EC2 instances to enable Session Manager access.

PostgreSQL client tools were installed to validate database connectivity.




Successful tests confirmed:

- Correct routing
- Proper security group configuration
- Valid NACL rules
- No direct internet exposure

10. Monitoring and Visibility

To maintain continuous visibility, the following services were enabled:

- VPC Flow Logs to CloudWatch Logs
- AWS Security Hub
- AWS Config

These services provided insight into network activity, misconfigurations, and compliance status.

11. Incident Response Procedures

An incident response process was designed to address potential exposure.

Detection:

- Security Hub findings
- AWS Config change history
- VPC Flow Log analysis

Investigation:

- Review affected resources
- Analyze configuration changes
- Examine access patterns

Containment:

- Tighten security group rules - Update NACL configurations
- Remove unintended public access

Validation:

- Re-test connectivity
- Confirm isolation
- Verify compliance

12. Outcomes and Impact

This implementation delivered the following results:

- Secure multi-tier network architecture
- Strong isolation of sensitive resources
- Reduced attack surface
- High availability design
- Improved audit readiness
- Continuous security visibility

13. Conclusion

I designed and implemented a secure, highly available AWS network architecture that enforces least privilege, strong segmentation, and continuous monitoring.

This solution supports secure operations, protects sensitive workloads, and aligns with enterprise security standards.