01

# 3-Tier Cloud Architecture on AWS

- Designed and deployed on Amazon Web Services
- Multi-AZ architecture
- Secure and isolated network design
- Load-balanced web application

This project demonstrates the deployment of a secure and highly available three-tier architecture on AWS, separating presentation, application, and data layers.

02

# Project Objective

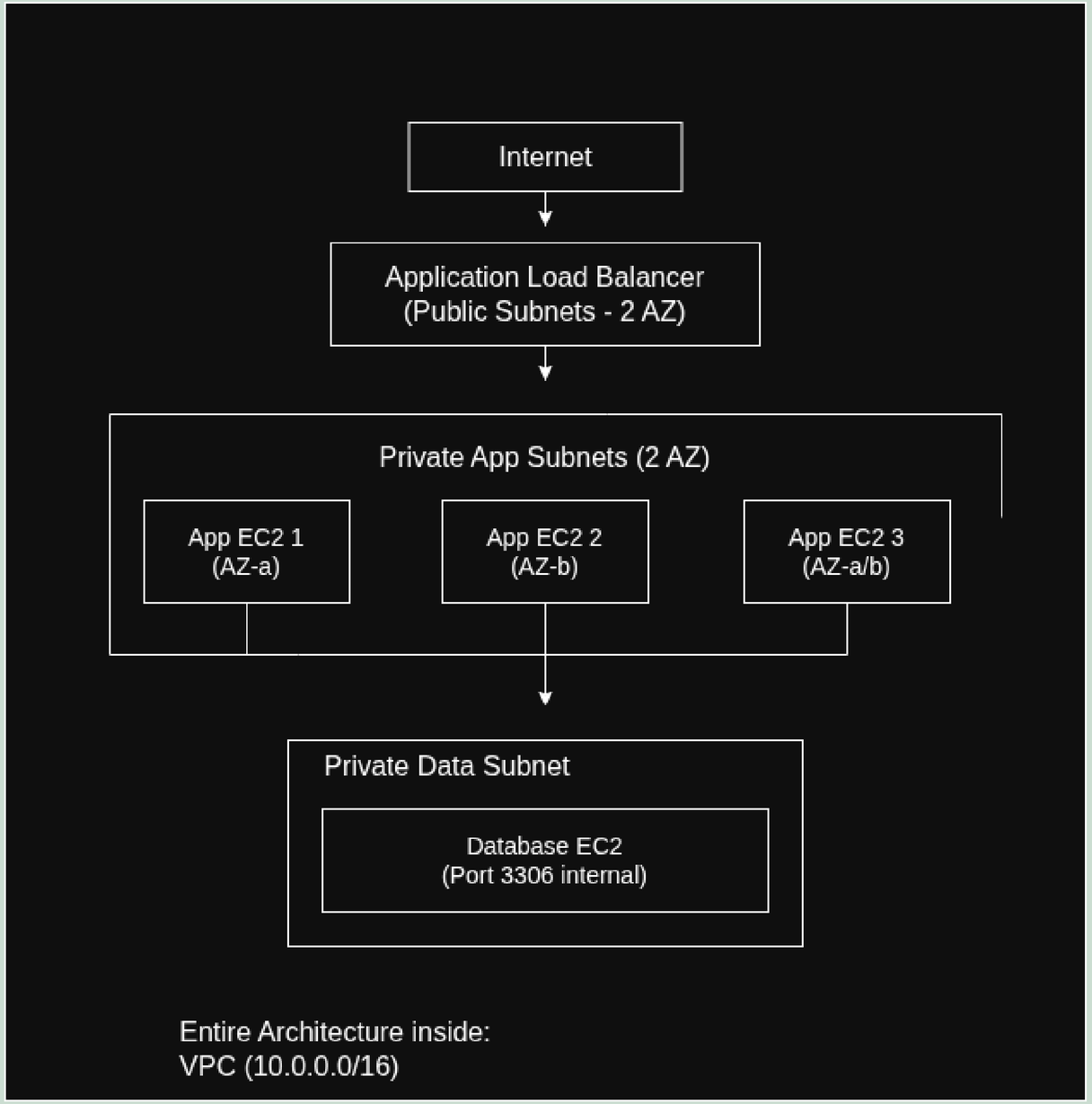
- Implement network segmentation
- Deploy across two Availability Zones
- Enforce controlled communication between tiers
- Ensure high availability and scalability

The goal was to implement network segmentation, controlled communication between layers, and load balancing across multiple Availability Zones while keeping application and data tiers private.

03

# Overall Architecture

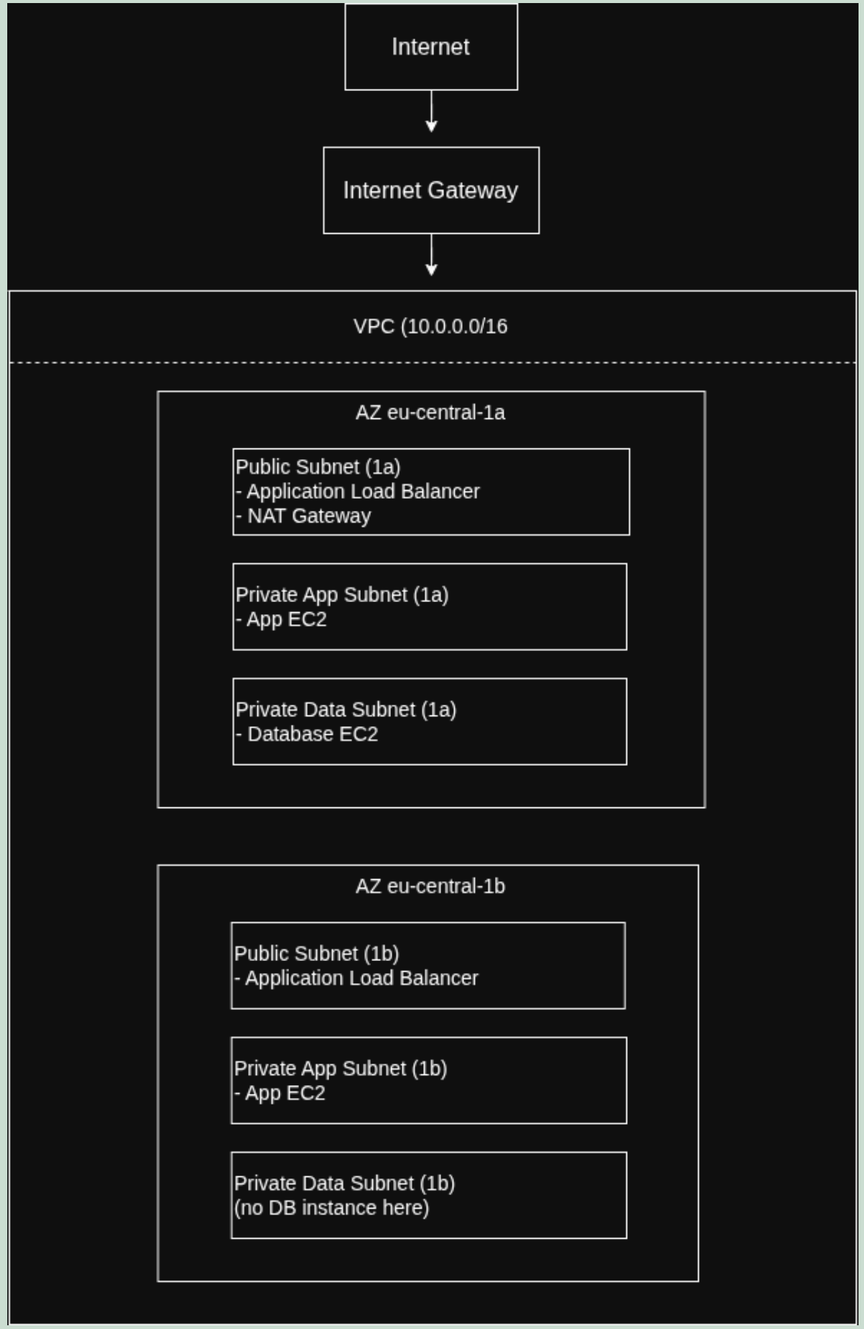
This diagram shows the complete three-tier structure.
Traffic flows from the Load Balancer to private EC2 instances, which communicate internally with the data layer.



04

# Network & Subnet Configuration

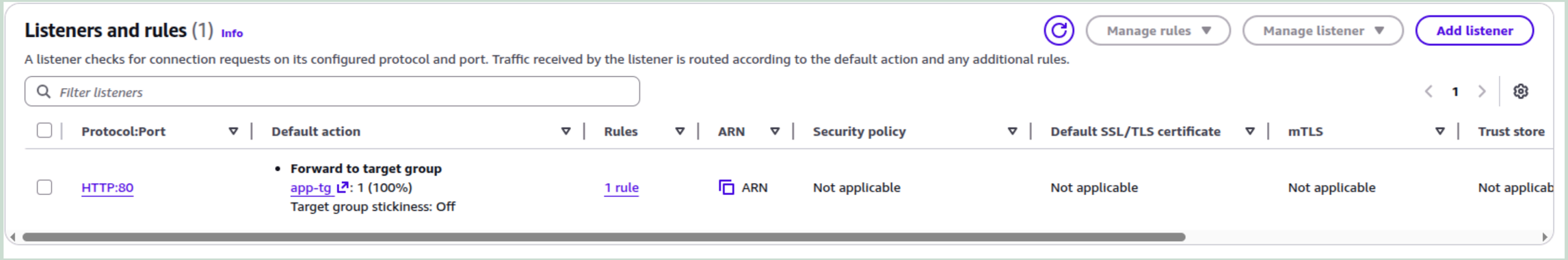
The VPC uses a /16 CIDR block and is divided into six subnets across two Availability Zones.
Public subnets host the Load Balancer, while application and data subnets remain private.



05

# Application Load Balancer Configuration

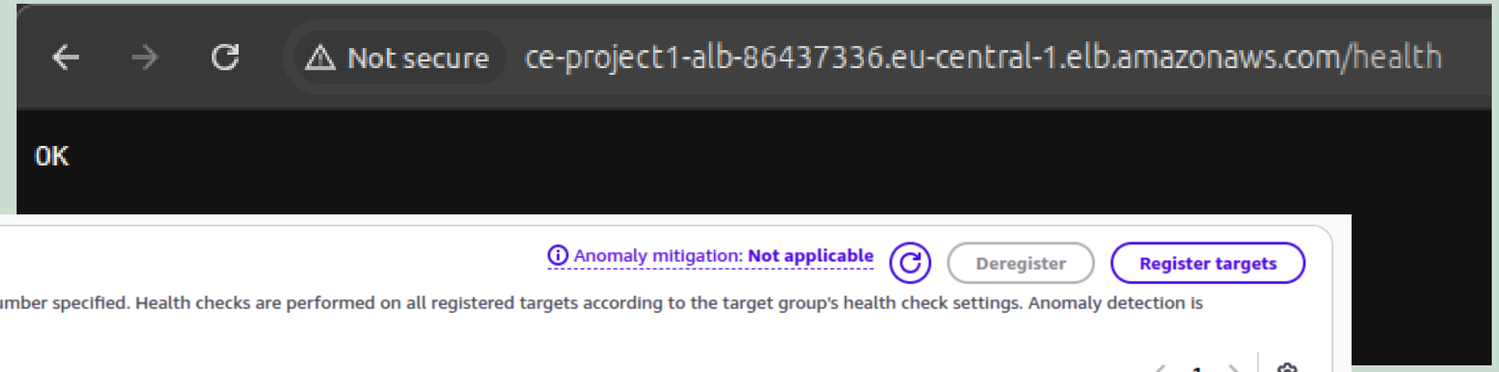
The Application Load Balancer listens on port 80 and forwards traffic to a target group.
It is deployed across public subnets to ensure internet accessibility and high availability.



06

# Target Group & Health Checks

The Load Balancer performs health checks using the /health endpoint.
Only healthy instances remain registered and receive traffic.




**Registered targets** (3) Info

Anomaly mitigation: **Not applicable** | Deregister | Register targets

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

Filter targets

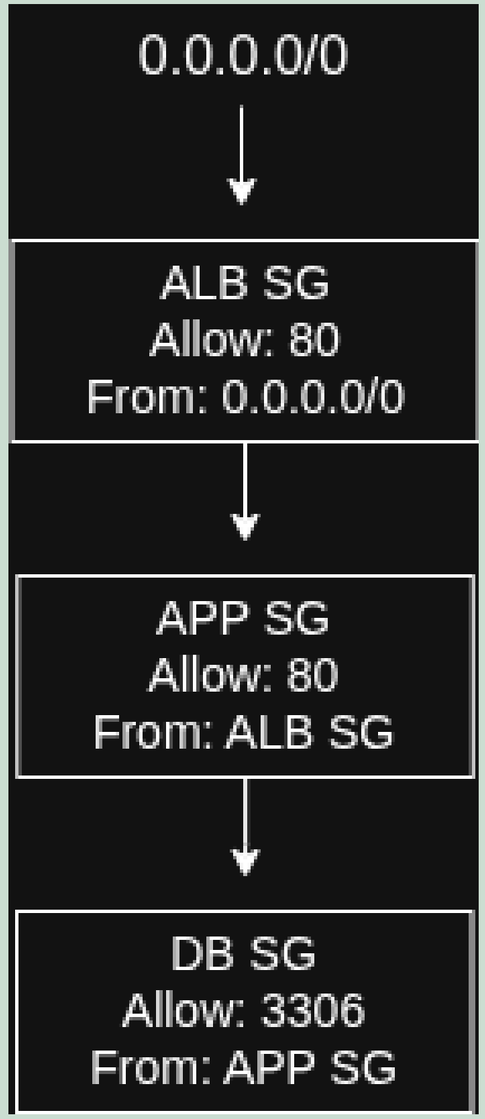
1

| Instance ID | Name | Port | Zone | Health status | Health status details | Administrative o... | Override details | Launch... | Anomaly |
|---|---|---|---|---|---|---|---|---|---|
| i-0503de32f5059b794 | app-server-3 | 80 | eu-central-1a (... | Healthy | - | No override | No override is curre... | February ... | Normal |
| i-0ddd35b522cb1cdae | app-server-2 | 80 | eu-central-1b (... | Healthy | - | No override | No override is curre... | February ... | Normal |
| i-03e4bb01ad4afaae7 | app-server-1 | 80 | eu-central-1a (... | Healthy | - | No override | No override is curre... | February ... | Normal |

07

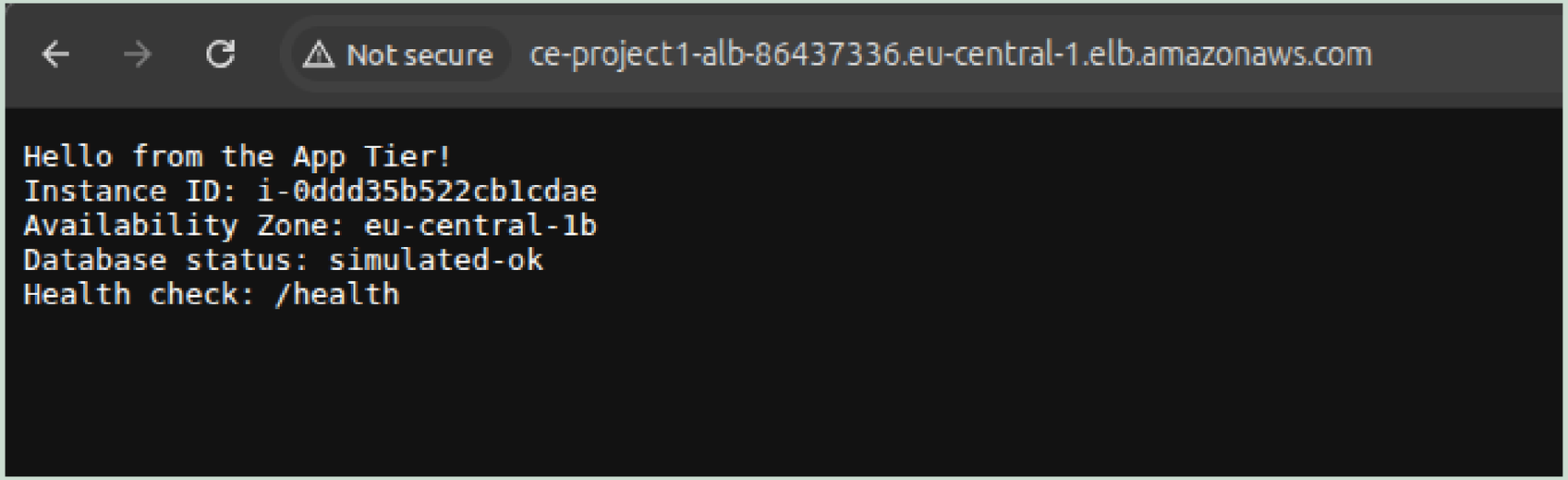
# Security Architecture & Network Isolation

Each tier in the architecture is protected by its own Security Group, ensuring that traffic only flows between the required components. The Application Load Balancer is the only resource exposed to the Internet, while the application and data layers remain private and inaccessible from outside the VPC.
Communication between tiers is strictly controlled, applying the principle of least privilege to reduce unnecessary exposure.



# Application in Action

The application displays Instance ID and Availability Zone, demonstrating load distribution across multiple EC2 instances.



# End-to-End Traffic Flow

This diagram illustrates how a user request travels through the system. A request is first received by the Application Load Balancer in the public subnet, then forwarded to a healthy EC2 instance in a private subnet. The application processes the request and communicates internally with the data layer when needed. The response then follows the same secure path back to the user.