Multi-Tier Cloud Architecture Design for SONYC RAG Application

Cloud Computing (SE-315)

Class: BESE29-A

Military College of Signals, NUST

Team Members

- Abdullah
- Muhammad Azfar
- Muhammad Affan
- Muhammad Hamza Haider
- Muhammad Muhad

CEA Assignment

Complex Engineering Activity

Executive Summary

SONYC Application Overview

A production-ready **Retrieval-Augmented Generation (RAG)** chatbot application deployed on AWS cloud infrastructure.

Key Highlights

- **99.9% Availability** — Target uptime
- **Multi-AZ Deployment** — High availability
- **Production Security** — Defense-in-depth
- **Auto-Scaling** — 100-1,000+ users

Production vs Development

	Development	Production
EC2	Single m7i-flex	**Multiple t3 instances**
Availability Zone	Single (eu-north-1b)	**Multi-AZ**
RDS Database	db.t4g.micro, Single-AZ	**db.t3.medium, Multi-AZ**
Security	Broad access (dev config)	**Least privilege, restricted**
Network	Simple VPC	**Multi-tier (ALB, CloudFront)**

High-Level Architecture Overview

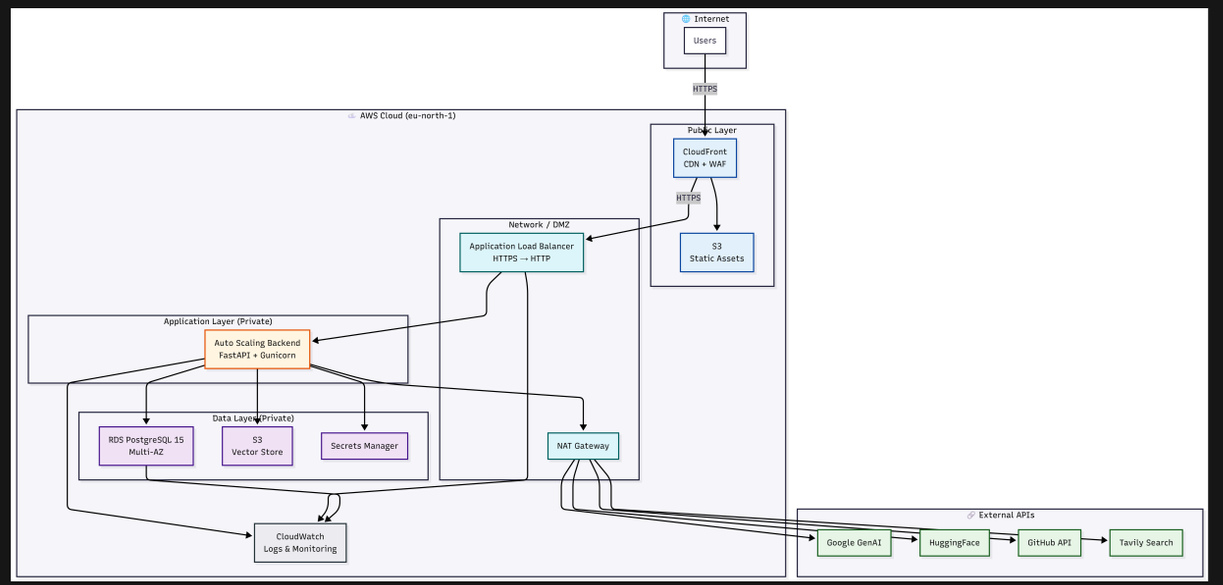


Figure 1: Multi-Tier Cloud Deployment Architecture

Presentation Layer

- CloudFront CDN
- Next.js Frontend
- S3 Static Assets

Application Layer

- Application Load Balancer
- FastAPI Backend (Auto Scaling)

Data Layer

- RDS PostgreSQL (Multi-AZ)
- S3 Vector Stores

External Services

- Google AI
- HuggingFace API
- GitHub API
- Tavily Search

Network Topology & Data Flow

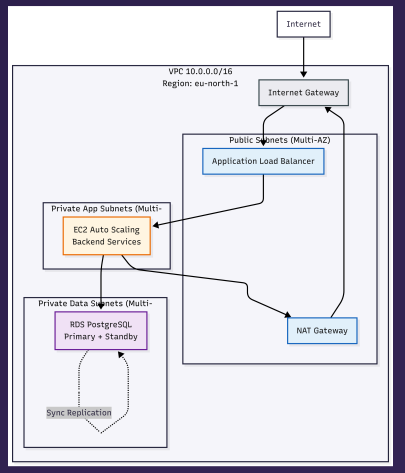


Figure 2: VPC Subnet Architecture

VPC Architecture

- **CIDR:** 10.0.0.0/16
- **Public Subnets:** ALB, NAT Gateway
- **Private App Subnets:** EC2 Backend
- **Private Data Subnets:** RDS PostgreSQL
- **Multi-AZ:** eu-north-1a, eu-north-1b

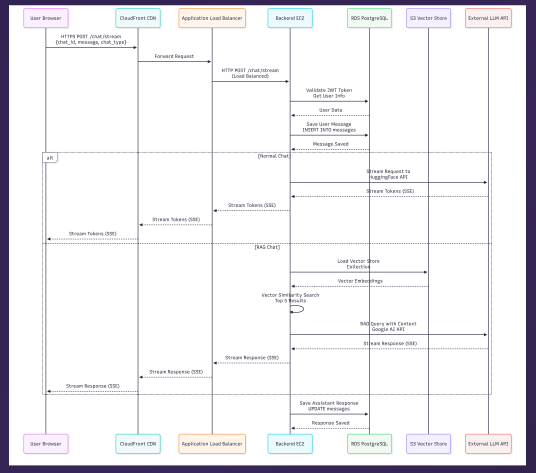


Figure 3: Chat Streaming Request Flow

Request Flow

User → CloudFront → ALB

Backend EC2 → RDS/S3 → External APIs

Detailed Design – Component Specifications

Frontend

Technology Stack
Next.js 15.5.7, React 18, TypeScript 5

Instance
t3.medium (2 vCPU, 4 GB)

CSS Framework
Tailwind CSS

Instances
2-5 instances

Backend

Framework
FastAPI 0.104+, Python 3.11

Instance
t3.large (2 vCPU, 8 GB)

Workers
5 workers per instance

Instances
2-8 instances

Database Layer

Database
PostgreSQL 15

RDS Instance
db.t3.medium Multi-AZ

Storage
100 GB gp3

Backup
7-day retention, Point-in-time

Vector Store

Engine
ChromaDB

Backend
S3 Standard

Embeddings
Google AI (1536-dim)

Technology Choices

RDS vs Self-Hosted

✓ Automated backups, Multi-AZ failover

EC2 vs ECS vs Lambda

✓ Predictable performance for RAG

S3 vs EBS

✓ 61% cost savings ($4.30/month)

ALB vs NLB

✓ Layer 7 routing, SSL termination

Cost Optimization 61%

S3 storage vs EBS-only

Reliability & Fault Tolerance

Failure Modes & Impact

- EC2 instance failure
- S3 vector store loss
- DDoS attack
- DB pool exhaustion
- DB transaction failure
- Network partition
- RDS primary failure
- External API failure
- Backend app crash
- Vector store corruption
- Authentication bypass

Recovery Mechanisms

Auto-recovery	5-15 min RTO
ASG replacement	5-10 min
RDS Multi-AZ failover	<60s RTO, 0 RPO
Circuit breaker	5 failures, 60s timeout

Multi-AZ Deployment

- All critical components deployed across multiple availability zones
- RDS synchronous replication for zero data loss
- Automatic health checks and instance replacement
- NAT Gateway redundancy across AZs

Service Level Objectives

Availability Target	99.9%
P95 Latency (Normal)	<2s
P95 Latency (RAG)	<5s
Recovery Time Objective	<15min

Scalability & Performance

Scaling Strategies

Horizontal Scaling (Primary)

- ✓ Auto Scaling Groups
- ✓ Stateless design (JWT)
- ✓ ALB load distribution

Vertical Scaling (Secondary)

- ✓ Instance type upgrades
- ✓ Reserved Instances

Capacity Planning

Baseline	**100 users** 2-4 instances
Peak	**1,000 users** 8 instances
Spike	**10x traffic** 3-5 min scale-out

Frontend ASG

Min / Max	**2 / 5**
Instance	**t3.medium**

Scale-out

- › CPU >70%
- › Memory >80%

Backend ASG

Min / Max	**2 / 8**
Instance	**t3.large**

Scale-out

- › CPU >75%
- › Queue >100

Scale-in

CPU <30% & Queue <20

Performance Optimization

Connection Pooling

5 connections/worker

Async Processing

Non-blocking I/O

CloudFront Caching

1yr immutable, 1hr HTML

Database Optimization

Indexes, connection limits

Scaling Triggers

- CPU utilization
- Memory usage
- HTTP Request count
- Queue depth

Security Design - Overview

Defense-in-Depth Layers

1. Network Perimeter: CloudFront, WAF, Shield
2. Application: WAF rules, JWT, input validation
3. Infrastructure: VPC segmentation, security groups
4. Data: AES-256 at rest, TLS 1.2+ in transit
5. Identity: IAM roles, Secrets Manager, audit logging

Encryption Strategy

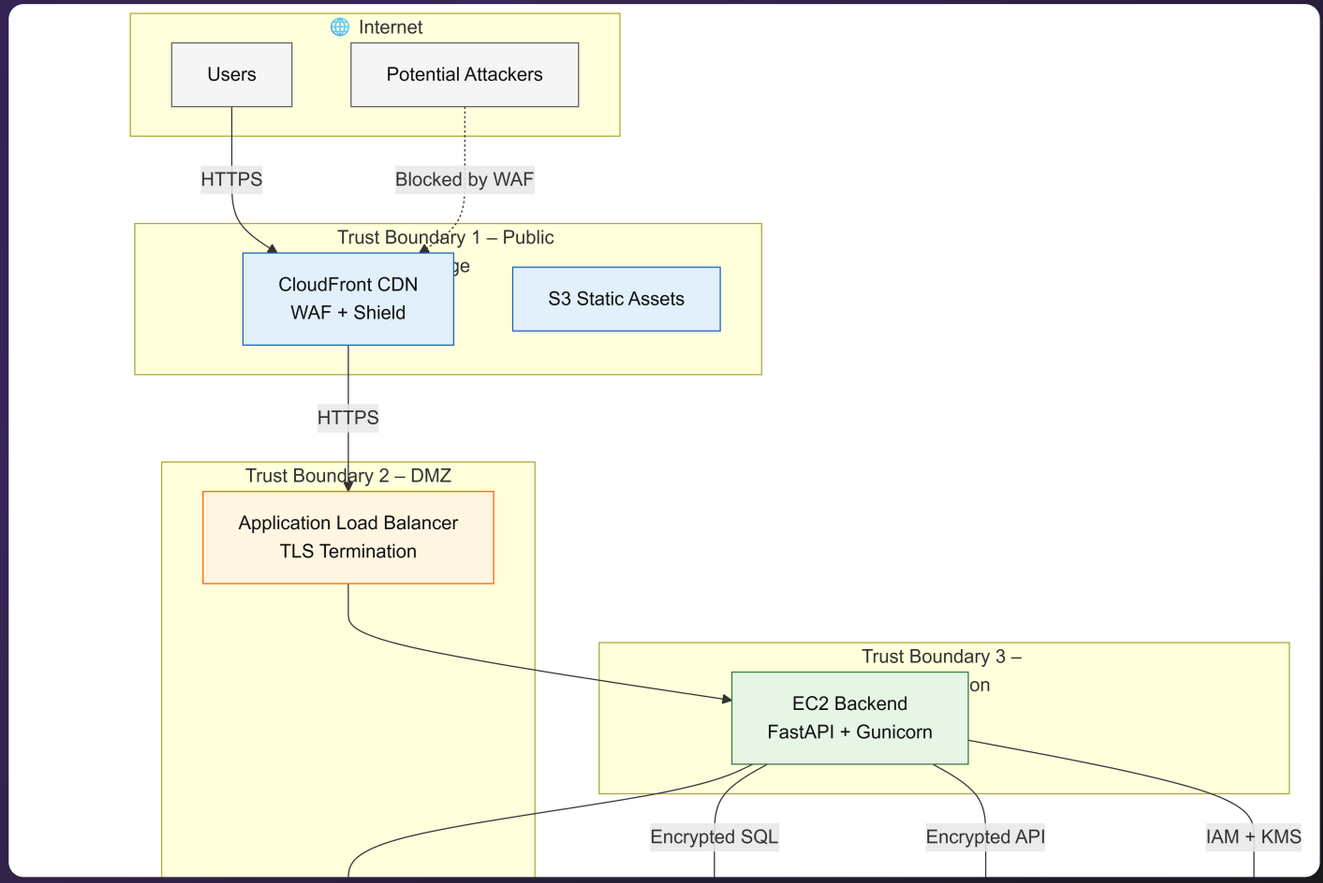
RDS	AES-256, TLS 1.2+
S3	SSE-KMS, HTTPS
EBS	KMS encryption
Secrets	KMS encryption

Authentication

JWT Token	256-bit secret
Expiration	30 days
Password Hashing	bcrypt (work 12)

Security Architecture & Trust Boundaries

Public DMZ Private



Security Design – Network Segmentation

VPC Architecture

10.0.0.0/16 | eu-north-1a | eu-north-1b

Public Subnets
- ALB (HTTPS 443)
- NAT Gateway

Private App Subnets
- EC2 Backend
- HTTP 8000 only

Private Data Subnets
- RDS PostgreSQL
- Port 5432 only

Security Groups
- Least privilege access
- Deny-all default
- Port-specific rules

Network ACLs
- Additional protection layer
- Stateless rules
- Subnet-level filtering

IAM & Access Control
- EC2 instance roles
- Least privilege policies
- No long-lived creds
- Audit logging enabled

Compliance Framework

GDPR Compliance
- Encryption
- Access controls
- Data retention
- EU data residency

SOC 2 Controls
- Security
- Monitoring
- Audit logging
- Controls

RDS Multi-AZ
- Synchronous replication
- <60s failover
- Zero data loss
- 7-day backups

Operational Plan - CI/CD Pipeline

CI/CD Pipeline Stages

1 Source Control
GitHub, PR reviews (2 approvals), branch protection

2 Build
Docker image, dependencies, linting

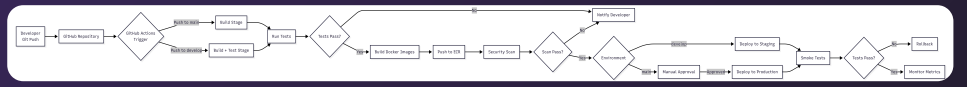
3 Test
Unit tests >70%, integration tests, security scans

4 Deploy Staging
Automatic (develop branch), smoke tests

5 Deploy Production
Manual approval, blue-green/canary, monitor 1h

6 Post-Deployment
Smoke tests, monitoring, rollback if error >5% or latency 2x

CI/CD Pipeline Flow



Blue-Green
Zero downtime

Canary
Gradual rollout

Rolling
Incremental update

Monitoring & Runbooks

CloudWatch Metrics & Alarms

CPU Utilization	>80%	5 min
Memory Usage	>85%	5 min
Error Rate	>5%	5 min
Response Time	>2x	10 min
DB Connections	>80%	max

Disaster Recovery

RDS	S3	Testing
7-day retention	Versioning enabled	Monthly drills
5-min RTO	Cross-region repl	Verify restores

Top Incidents & Resolutions

High Error Rate

- ✓ Check CloudWatch logs
- ✓ Verify external APIs
- ✓ Rollback if code issue
- ✓ Activate circuit breaker

Database Failure

- ✓ Verify Multi-AZ failover
- ✓ Check RDS metrics
- ✓ Restore from backup
- ✓ <60s automatic failover

Instance Unhealthy

- ✓ Check /health endpoint
- ✓ Restart service
- ✓ ASG auto-replace (5-10 min)
- ✓ Review application logs

DDoS Attack

- ✓ Check traffic spike
- ✓ Review Shield alerts
- ✓ WAF rate limiting
- ✓ Escalate to security team

Cost Estimate

Baseline Cost

$475.23 /month

100 users, 1,000 req/day

Optimized Cost

$325–350 /month

With optimization strategies

Cost Breakdown

EC2 Frontend	$67.64	EC2 Backend	$144.47
RDS (Multi-AZ)	$116.62	ALB	$22.00
CloudFront	$10.00	NAT Gateway	$72.00
S3 Storage	$5.00	Secrets Manager	$2.50
CloudWatch	$15.00	Data Transfer	$20.00

Optimization Strategies

Reserved Instances — $63/mo
1-year term, 30% savings

S3 Intelligent-Tiering — $1/mo
Automatic cost optimization

Right-Sizing — $50/mo
Monitor and adjust instances

NAT Gateway Opt — $36/mo
Single NAT, VPC endpoints

Cost Per User

Baseline

$4.75 /user/mo

Optimized

$3.25–3.50 /user/mo

Testing Strategy

Test Pyramid

70% Unit Tests
Individual functions/components

20% Integration Tests
API endpoints, DB operations

10% E2E Tests
Complete user workflows

Load Testing Scenarios

Baseline
100 users, <200ms

Peak
1,000 users, <2s

Spike
10x traffic, rapid scale

Stress
Exceed capacity, graceful degradation

Endurance Test
24-hour sustained load, no memory leaks

Chaos Engineering

EC2 Termination
ASG replacement 5-10 min

RDS Primary Failure
Multi-AZ failover <60s

External API Failure
Circuit breaker, graceful degradation

Network Partition
Partial availability, RDS failover

Security Testing

Penetration Testing
OWASP Top 10, SQL injection, XSS

Vulnerability Scanning
Dependencies, Docker images, infrastructure

Security Audits
Quarterly reviews, IAM audits, compliance

Pass Criteria

- Unit tests >70%
- Integration OK
- E2E paths OK

Metrics

- Latency <2s
- Error rate <1%
- Auto-scale <5 min

Frequency

- Unit tests daily
- Load tests weekly
- Security audits quarterly

Testing Philosophy: Comprehensive coverage, automated pipelines, continuous monitoring

Risk Register

R1: Data Breach
CRITICAL MED

Encryption, IAM roles, WAF, network segmentation, CloudTrail logging, security audits

R2: External API Failure
HIGH MED

Circuit breaker (5 fails/60s), fallback responses, monitoring, retry logic, graceful degradation

R3: Cost Overruns
MED MED

Reserved Instances (30% savings), AWS Budgets, Cost Explorer, right-sizing, monthly reviews

R4: Scalability Bottlenecks
HIGH LOW

Auto-scaling (2-8 instances), load testing, capacity planning, monitoring, performance optimization

R5: Single Region Deployment
MED LOW

Multi-AZ deployment, cross-region replication (optional), multi-region (future)

R6: Database Failure
CRITICAL LOW

RDS Multi-AZ (auto failover <60s), automated backups (7-day), point-in-time recovery (5-min RPO)

Key Mitigations

- **R1:** Defense-in-depth encryption & monitoring
- **R2:** Circuit breaker & fallback mechanisms
- **R3:** Reserved Instances & cost monitoring
- **R6:** Multi-AZ failover & automated backups

2	4	2
CRITICAL	HIGH/MED	LOW

Risk Monitoring

- Quarterly risk review meetings
- Update based on incidents & testing
- Critical risks escalate immediately

R7: Compliance
HIGH LOW

R8: Personnel
MED LOW

Conclusion & Future Work

Key Achievements

- **99.9% Availability**
  Target achieved
- **Multi-AZ Deployment**
  <60s RTO, 0 RPO
- **Defense-in-Depth Security**
  5 protection layers
- **Auto-Scaling Capability**
  2-8 backend instances
- **Production Monitoring**
  Comprehensive alerting
- **Cost Optimization**
  61% savings achieved

Current Limitations

- Single region deployment (eu-north-1)
- Reserved Instances pending
- Advanced monitoring (X-Ray) & caching (Redis) planned
- Budget constraints limit capacity
- External API dependencies

Ready to Scale

Production-ready architecture designed for growth, security, and operational excellence

99.9%	<60s
Availability	Failover

Future Enhancements

3-6 MONTHS

- Multi-region deployment
- AWS X-Ray integration
- Redis caching layer
- GuardDuty & Reserved Instances

6-12 MONTHS

- ECS Fargate migration
- Distributed vector database
- Advanced RAG techniques
- Read replicas for scaling

12+ MONTHS

- Microservices architecture
- ML pipeline integration
- Voice & mobile support
- Multi-cloud strategy

Team Reflection

Team Overview

Cloud Computing (SE-315) • BESE29-A

- Abdullah
- Muhammad Azfar
- Muhammad Affan
- Muhammad Hamza Haider
- Muhammad Muhad

Key Learnings

- Comprehensive cloud architecture design
- Balancing cost, performance & security
- Documentation & operational procedures
- Systematic risk management
- Practical application of course concepts

Individual Contributions

Abdullah

Architecture • Scalability • Security • Team Coordination

Muhammad Azfar

Design • Reliability • Risk Register

Muhammad Affan

Testing • Conclusion • Architecture Diagrams

M. Hamza Haider

Cost • Testing • Conclusion

Muhammad Muhad

Operational Plan • Cost

99.9%
Availability

5 Layers
Security

<60s
Failover

Thank You

Questions & Discussion

Cloud Computing SE-315

BESE29-A