# eGroceryMart System Architecture
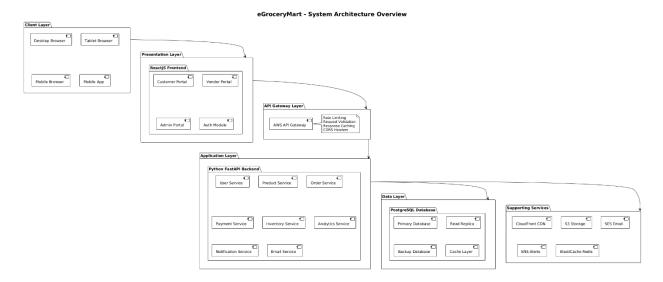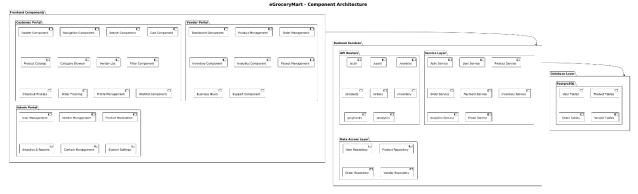
## Table of Contents

# Overview

This document outlines the complete system architecture for the eGroceryMart multi-vendor eCommerce platform, including frontend, backend, database, and AWS deployment strategy.

## 1. System Architecture Overview



eGroceryMart - System Architecture Overview

## 2. Component Architecture
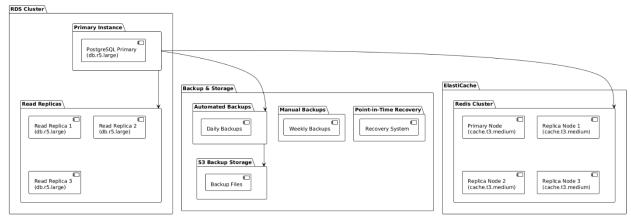


eGroceryMart - Component Architecture

# 3. Database Layer (PostgreSQL)

*Database Architecture*

**eGroceryMart - Database Deployment Architecture**

RDS Cluster

Primary Instance

PostgreSQL Primary
(db.r5.large)

Read Replicas

Read Replica 1
(db.r5.large)

Read Replica 2
(db.r5.large)

Read Replica 3
(db.r5.large)

Backup & Storage

Automated Backups

Daily Backups

Manual Backups

Weekly Backups

Point-in-Time Recovery

Recovery System

S3 Backup Storage

Backup Files

ElastiCache

Redis Cluster

Primary Node
(cache.t3.medium)

Replica Node 1
(cache.t3.medium)

Replica Node 2
(cache.t3.medium)

Replica Node 3
(cache.t3.medium)

# 4. AWS Deployment Architecture

## 4.1 High-Level AWS Infrastructure

**eGroceryMart - AWS Deployment Architecture**

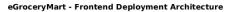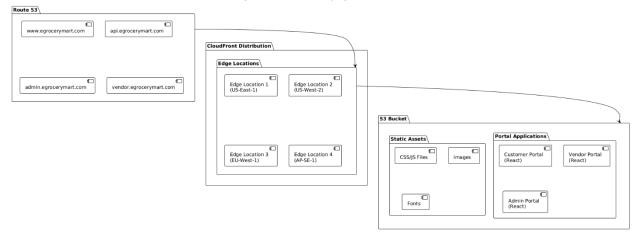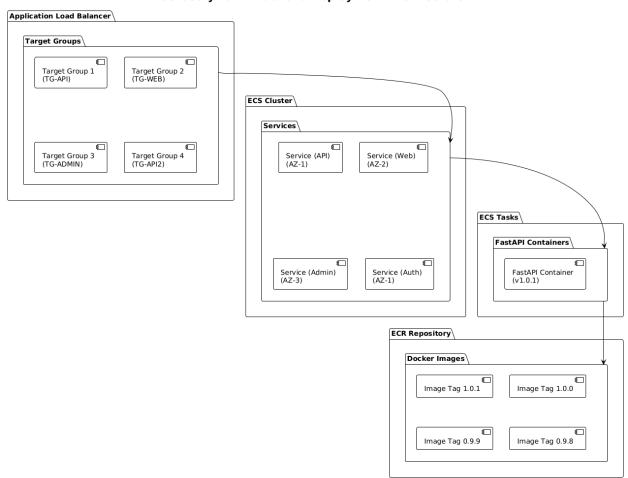# 4.2 Detailed AWS Services Architecture
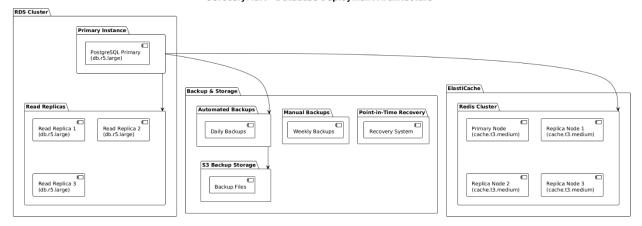
## Frontend Deployment (ReactJS)

**eGroceryMart - Frontend Deployment Architecture**

**Route 53**
- www.egrocerymart.com
- api.egrocerymart.com
- admin.egrocerymart.com
- vendor.egrocerymart.com

**CloudFront Distribution**

**Edge Locations**
- Edge Location 1 (US-East-1)
- Edge Location 2 (US-West-2)
- Edge Location 3 (EU-West-1)
- Edge Location 4 (AP-SE-1)

**S3 Bucket**

**Static Assets**
- CSS/JS Files
- Images
- Fonts

**Portal Applications**
- Customer Portal (React)
- Vendor Portal (React)
- Admin Portal (React)

## Backend Deployment (FastAPI)

**eGroceryMart - Backend Deployment Architecture**

**Application Load Balancer**

**Target Groups**
- Target Group 1 (TG-API)
- Target Group 2 (TG-WEB)
- Target Group 3 (TG-ADMIN)
- Target Group 4 (TG-API2)

**ECS Cluster**

**Services**
- Service (API) (AZ-1)
- Service (Web) (AZ-2)
- Service (Admin) (AZ-3)
- Service (Auth) (AZ-1)

**ECS Tasks**

**FastAPI Containers**
- FastAPI Container (v1.0.1)

**ECR Repository**

**Docker Images**
- Image Tag 1.0.1
- Image Tag 1.0.0
- Image Tag 0.9.9
- Image Tag 0.9.8

*Database Deployment (PostgreSQL)*

**eGroceryMart - Database Deployment Architecture**



# 5. AWS Services Breakdown

1. Compute Services

- **ECS (Elastic Container Service)**: Container orchestration for FastAPI
- **Lambda**: Serverless functions for specific tasks

2. Storage Services

- **S3**: Static website hosting, file storage, backups
- **CloudFront**: CDN for global content delivery

3. Database Services

- **RDS PostgreSQL**: Primary database with read replicas
- **ElastiCache**: Redis for session management and caching

4. Networking Services

- **VPC**: Virtual private cloud for network isolation
- **ALB**: Application Load Balancer for traffic distribution
- **API Gateway**: API management and security
- **Route 53**: DNS management and health checks

5. Security Services

- **IAM**: Identity and access management
- **WAF**: Web Application Firewall
- **CloudTrail**: API call logging
- **KMS**: Key management for encryption

6. Monitoring Services

- **CloudWatch**: Monitoring and logging

- **SNS**: Notifications and alerts
- **X-Ray**: Distributed tracing

7. Integration Services

- **SES**: Email service for notifications
- **SQS**: Message queuing for async processing
- **EventBridge**: Event-driven architecture

# 6. Security Architecture

Network Security
- **VPC with private subnets** for backend services
- **Security groups** for traffic control
- **NACLs** for subnet-level security

Application Security
- **HTTPS/TLS** encryption for all communications
- **JWT tokens** for authentication
- **API Gateway** for rate limiting and validation
- **WAF** for web application protection

Data Security
- **Encryption at rest** for all data
- **Encryption in transit** for all communications
- **IAM roles** for service permissions
- **Database encryption** for sensitive data

# 7. Scalability Strategy

Horizontal Scaling
- **Auto Scaling Groups** for ECS services
- **Read replicas** for database scaling
- **CDN** for global content delivery
- **Load balancers** for traffic distribution

Caching Strategy
- **CloudFront** for static content

- **ElastiCache Redis** for session data
- **Application-level caching** for frequently accessed data
- **Database query caching** for performance

## 8. Monitoring and Alerting

Infrastructure Monitoring
- **CloudWatch metrics** for all AWS services
- **Log aggregation** with CloudWatch Logs

Application Monitoring
- **Error tracking** and alerting

Alerting Strategy
- **SNS topics** for different alert types
- **Email notifications** for critical issues
- **SMS alerts** for urgent problems

## 9. Disaster Recovery

Backup Strategy
- **Automated daily backups** for RDS
- **S3 versioning** for file protection
- **Point-in-time recovery** capabilities

Recovery Strategy
- **Cross-region replication** for critical services
- **Automated failover** procedures
- **Recovery time objectives** (RTO) and recovery point objectives (RPO)