DEV OPPS

DEPLOYMENT MODEL

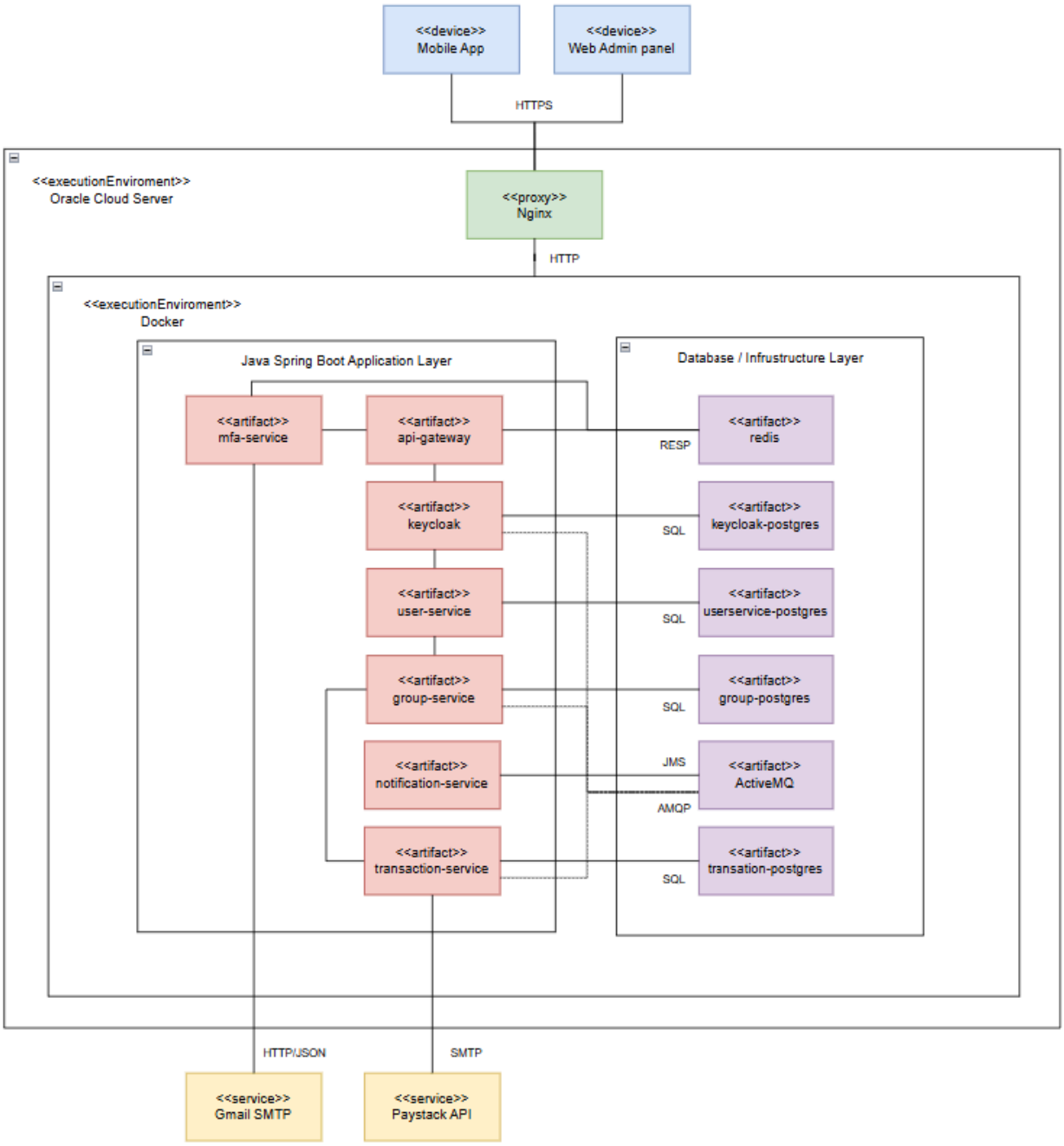
DEMO 3

STOCKFELLOW

BRIGHT BYTE ENTERPRISES

TABLE OF CONTENTS

BACKEND DEPLOYMENT DIAGRAM



DEPLOYMENT ENVIRONMENT

The StockFellow fintech backend will be deployed as a containerized Java Spring Boot microservices architecture on Oracle Cloud's free tier ARM compute instance. The system uses Docker Compose for orchestration and is cost-effective with very capable hardware.

Target Platform

- **Cloud Provider**: Oracle Cloud Infrastructure (OCI)
- **Instance Type**: ARM-based Ampere A1 Compute (Always Free)
- **Specifications:** 24GB RAM, 4 OCPUs, 200GB Storage
- **Operating System**: Ubuntu 22.04 LTS
- **Container Runtime**: Docker with Docker Compose

Reasons For Oracle Selection

- **Low/No cost** with generous Always Free services
- **Sufficient resources** for all microservices and databases
- **ARM architecture** provides excellent performance per core

Deployment Pattern

The system implements a **single-server containerized microservices architecture** with:

- **NGINX Reverse Proxy**: SSL termination and load balancing
- **Java Spring Boot Services**: Business logic microservices
- **PostgreSQL Databases**: Relational data storage
- **Redis Cache**: Session management and caching

SERVICE ARCHITECTURE OVERVIEW

Client Access Layer

- **Mobile Apps**: React Native/Flutter applications
- **Web Admin Panel**: Browser-based administration
- **External Access**: HTTPS through domain name or public IP

Proxy Layer

- **NGINX**: Port 80/443 (SSL termination, reverse proxy)

Application Layer (Java Spring Boot)

- **API Gateway**: Port 3000 (Request routing, authentication)
- **User Service**: Port 4000 (User management, profiles)
- **Group Service**: Port 4040 (Investment groups, communities)
- **Transaction Service**: Port 4080 (Payment processing, Paystack integration)
- **MFA Service**: Port 8087 (Multi-factor authentication)
- **Notification Service**: Port 4050 (Push notifications, messaging)
- **Keycloak**: Port 8180 (Identity and access management)

Database Layer

- **Keycloak PostgreSQL**: Port 5432 (Identity data)
- **User Service PostgreSQL**: Port 5431 (User profiles, accounts)
- **Notification PostgreSQL**: Port 5440 (Notification history)
- **Redis Cache**: Port 6379 (Sessions, cache)
- **ActiveMQ**: Port 61616 (Message broker)

External Services

- **Gmail SMTP**: Email delivery for MFA
- **Paystack API**: Payment processing

CONTAINER ARCHITECTURE

Application Containers

Service	Technology	Port	Purpose
nginx-proxy	NGINX	80, 433	Reverse proxy, SSL termination
api-gateway	Spring Boot	3000	API routing, authentication
user-service	Spring Boot	4000	User management
group-service	Spring Boot	4040	Group operations
transaction-service	Spring Boot	4080	Payment processing
mfa-service	Spring Boot	8087	Multi-factor authentication
notification-service	Spring Boot	4050	Notifications
keycloak	Keycloak	8180	Identity management

Database Containers

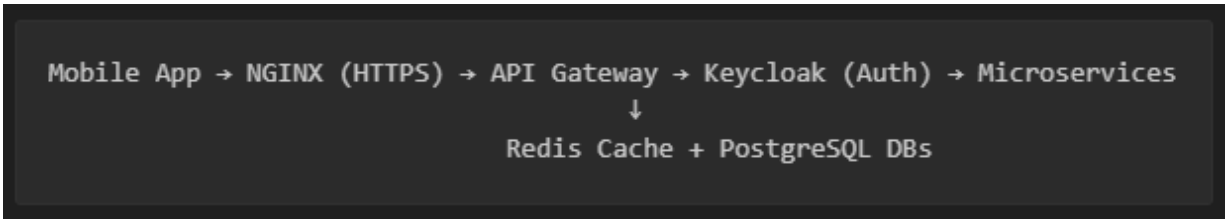
Service	Technology	Port	Purpose
keycloak-postgres	PostgreSQL 15	5432	Keycloak configuration + data
user-postgres	PostgreSQL 15	5431	User data
notification-postgres	PostgreSQL 15	5440	Notification data
group-postgres	PostgreSQL 15	5433	Group data
redis	Redis 7	6379	Cache and sessions
active-mq	ActiveMQ Artemis	61616	Message brokering

NETWORK ARCHITECTURE

Security Model

- **External Access:** Only NGINX (ports 80, 443) exposed to internet
- **Internal Network:** All services communicate via Docker bridge network
- **SSL/TLS:** NGINX handles SSL termination
- **Authentication:** JWT tokens via Keycloak for all API access

Communication Flow



DEPLOYMENT PROCESS

Configuration

- **Environment Variables:** All configuration via .env file
- **Secrets:** Database passwords, API keys in environment variables
- **External URLs:** Domain-based URLs for production
- **JVM Settings:** Optimized for 24GB RAM allocation

Workflow

- **CI Pipeline:** Runs all tests and linting
- **CD Pipeline:** Deploys generated Docker artifacts to servers Docker instance

RESOURCE ALLOCATION

Memory Distribution

- **Java Services:** 10GB (6 services × ~1.5GB each)
- **Keycloak:** 2GB
- **PostgreSQL:** 1.5GB (3 instances)
- **Redis + ActiveMQ:** 0.5GB
- **NGINX:** 0.1GB
- **System/OS:** 6GB
- **Buffer:** 4GB

MONITORING AND MAINTENANCE

Health Monitoring

- **Health Endpoints:** All Spring Boot services expose /actuator/health
- **Database Health:** pg_isready for PostgreSQL instances
- **System Monitoring:** monitor.sh script for resource usage

Logging

- **Application Logs:** Docker container logs
- **Access Logs:** NGINX request logging
- **Error Tracking:** Centralized via Docker logs

SECURITY CONSIDERATIONS

Network Security

- **Firewall:** Only ports 22, 80, 443 open to internet
- **Internal Communication:** Services isolated in Docker network
- **SSL/TLS:** All external communication encrypted

Application Security

- **OAuth 2.0:** Authentication via Keycloak
- **JWT Tokens:** Secure service-to-service communication
- **Database Security:** Isolated databases per service
- **Secret Management:** Environment variable based
- **Tokenized Payment Details:** via Paystack Authorizations