KYC Onboarding — Full Execution & Installation Guide

Full Execution & Installation Guide (Beginner Friendly)

This doc is for someone brand new. You will run the full stack:
- MySQL + Redis + Kafka + Prometheus + Grafana + SonarQube (Docker)
- Spring Boot backend
- Angular frontend
- Validate: Swagger, Kafka events, Prometheus scrape, Grafana dashboards

---

0) What you need to install (one-time)

Windows / Mac / Linux
1. **Git**
2. **Docker Desktop** (includes Docker Compose)
3. **Java 17** (JDK)
4. **Maven** (or use IDE Maven)
5. **Node.js 20+** and **npm**
6. (Optional) **Sonar Scanner** (`sonar-scanner`)

---

1) Get the code ```bash git clone cd kyc-onboarding-app ```

---

2) Start all infrastructure using Docker From the repo root: ```bash docker compose up -d ```

This starts:
- MySQL (3306)
- Redis (6379)
- Kafka (9092) + Zookeeper (2181)
- Prometheus (9090)
- Grafana (3000) with **auto-provisioned dashboard + Prometheus datasource**
- SonarQube (9000)

Check containers:

```
bash
docker ps
```

---

3) Run the backend (Spring Boot)

Option A: Run from terminal

```
bash
cd backend
mvn spring-boot:run
```

Backend URLs:
- API: `http://localhost:8080`
- Swagger UI: `http://localhost:8080/swagger-ui.html`
- Prometheus metrics: `http://localhost:8080/actuator/prometheus`

Option B: Run from IDE (IntelliJ/Eclipse)
Import `backend` as Maven project and run the main Spring Boot class.

---

4) Run the frontend (Angular) ```bash cd frontend npm install npm start ```

Frontend URL:
- `http://localhost:4200`

---

5) Create users + login (Postman) Import: - `postman/KYC-Onboarding.postman_collection.json`

Set:
- `baseUrl = http://localhost:8080`

Run:
1. `Auth -> Register`
2. `Auth -> Login (JWT)` (token auto-saved)
3. `Cases -> Create Case`
4. `Cases -> Save Step 1`
5. `Documents -> Upload Document`

---

6) Kafka testing (events)

Produce an event via API

```
bash
curl -X POST http://localhost:8080/api/streams/case-event ^
  -H "Content-Type: application/json" ^
  -d @docs/kafka-samples/case-event.json
```

Validate Kafka topic has data

```
bash
docker exec -it kafka kafka-console-consumer   --bootstrap-server localhost:9092   --topic kyc.case.
```

---

7) Observability validation (Prometheus + Grafana)

Prometheus
Open:
- `http://localhost:9090`

Try query:
- `rate(http_server_requests_seconds_count[1m])`

Grafana
Open:
- `http://localhost:3000`
Login:
- user: `admin`
- pass: `admin`

You should see:
- Folder: **KYC**
- Dashboard auto-imported: **KYC Backend - Metrics**

If dashboard is empty, generate traffic:
- open Swagger UI and call APIs, or use Postman calls.

---

8) Audit export validation (Admin only) - CSV: `GET http://localhost:8080/api/admin/audit-export/csv` - Parquet: `GET http://localhost:8080/api/admin/audit-export/parquet`

---

9) Tests + Coverage

Backend

```
bash
cd backend
mvn test
```

Coverage report:
- `backend/target/site/jacoco/index.html`

Frontend

```
bash
cd frontend
npm run test:coverage
```

Coverage file:
- `frontend/coverage/lcov.info`

---

10) SonarQube (optional) Start already via compose, open: - `http://localhost:9000`

Then run scanner from repo root:

```
bash
sonar-scanner -Dsonar.host.url=http://localhost:9000 -Dsonar.login=<TOKEN>
```

---

11) Stop everything ```bash docker compose down ```