

CAPSTONE PROJECT

# Utility Billing System

# Project Overview

Utility Billing System uses a microservices architecture to manage end-to-end utility billing operations such as consumer management, tariff configuration, meter readings, bill generation, payments, and notifications.

The system supports multiple user roles with strict role-based access control, ensuring that each user interacts only with the functionalities relevant to their responsibilities. This application is designed to resemble real-world billing systems used by electricity, water, or gas utility providers.

## Core Modules

- Auth Service: User registration, login, password management, JWT generation
- Consumer Service: Consumer profile and connection management
- Utility Service: Utility types and tariff plan management
- Meter Service: Meter reading submission and consumption tracking
- Billing Service: Bill generation, bill status management
- Payment Service: Payment processing and bill settlement
- Notification Service: Email notifications for user approval and bill overdue events

# Roles & Responsibilities

## ADMIN

Manages users, approves consumers, configures utilities and tariffs, and oversees system operations.

## BILLING OFFICER

Generates bills based on meter readings, manages billing cycles, and tracks bill statuses.

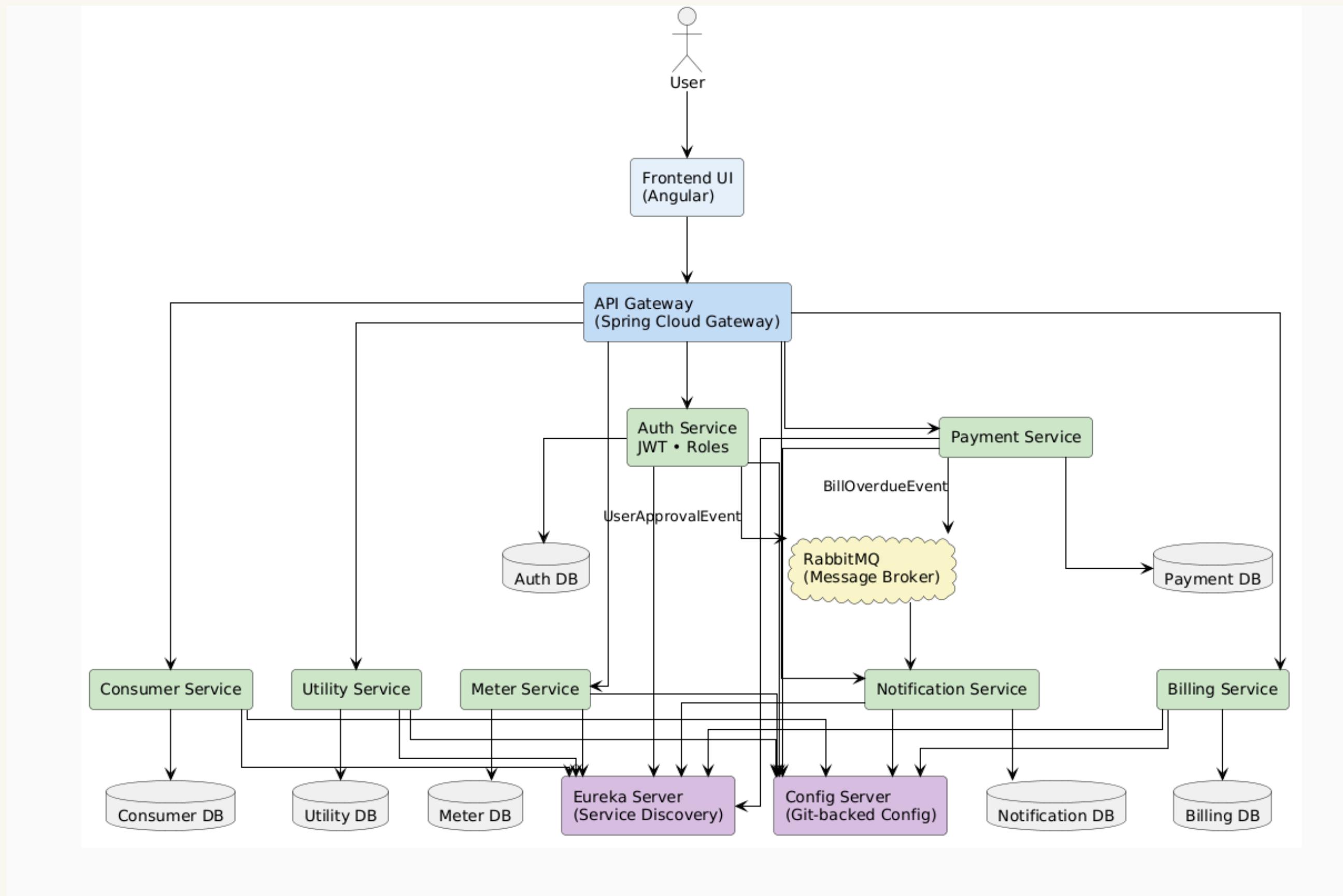
## ACCOUNTS OFFICER

Handles payments, verifies transactions, and sends overdue reminders.

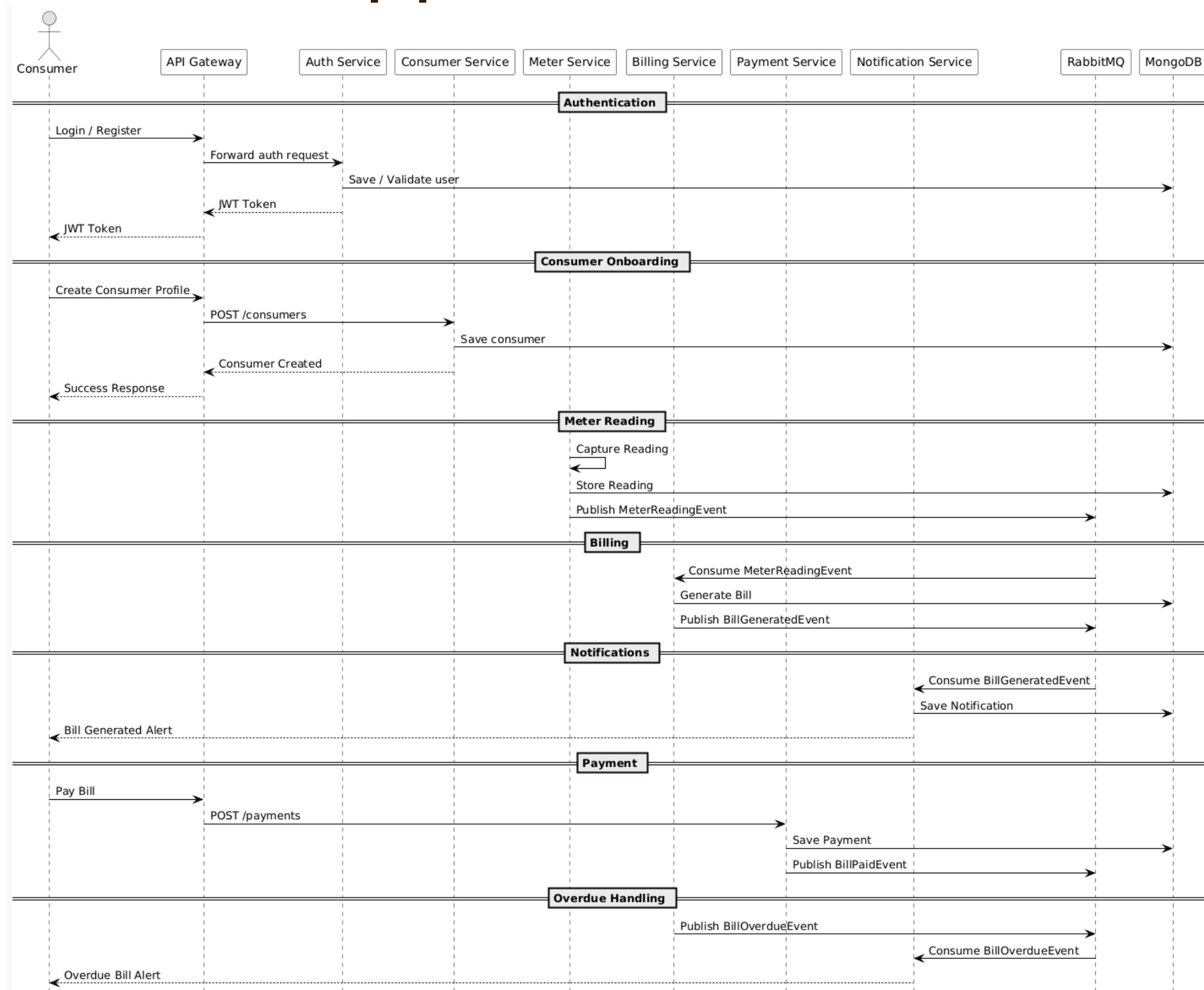
## CONSUMER

Views bills, manages personal utility connections, makes payments, and receives notifications.

# System Architecture



# Application Flow



# SonarQube Coverage Report

The screenshot shows the SonarQube cloud interface for the project `utility-billing-system`. The main summary indicates a **Passed** status. The dashboard displays various metrics across four categories: Security, Reliability, Maintainability, and Duplications.

**Security:** 0 Open issues (Grade A)

**Reliability:** 1 Open issue (Grade D)

**Maintainability:** 0 Open issues (Grade A)

**Duplications:** 2.4% (Grade C)

**Coverage:** 92.1% (Grade B)

**Accepted Issues:** 0

**Information:** 0

**Administration:** 0

**Overall Code:** 606 Lines to cover

**Security Hotspots:** 0

# Docker Containers

The screenshot shows the Docker Desktop interface with the following details:

- Top Bar:** docker desktop PERSONAL, Search bar, Ctrl+K, Help, Notifications (1), Settings, Grid View, Sign in, Minimize, Maximize, Close.
- Sidebar:** Containers (selected), Images, Volumes, Builds, Docker Hub, Docker Scout, Extensions.
- Container Overview:** Container CPU usage (5.26% / 1200% (12 CPUs available)), Container memory usage (3.76GB / 7.43GB), Show charts.
- Search and Filter:** Search bar, Filter icon, Only show running containers.
- Table of Containers:** A grid showing 13 items (running containers).

	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	billing-service	41d0c0d221f2	backend-bi	8086:8086 ↗	0.16%	3 hours ago	<input type="checkbox"/> ⋮ ⚡
<input type="checkbox"/>	api-gateway	1eb641b640a0	backend-ap	9999:9999 ↗	0.14%	3 hours ago	<input type="checkbox"/> ⋮ ⚡
<input type="checkbox"/>	consumer-service	bb8cafa95237	backend-cc	8081:8081 ↗	0.22%	3 hours ago	<input type="checkbox"/> ⋮ ⚡
<input type="checkbox"/>	meter-service	c62d76960196	backend-m	8085:8085 ↗	0.18%	3 hours ago	<input type="checkbox"/> ⋮ ⚡
<input type="checkbox"/>	payment-service	509bff7fd395	backend-pa	8087:8087 ↗	0.16%	3 hours ago	<input type="checkbox"/> ⋮ ⚡
<input type="checkbox"/>	auth-service	ca6a3a41fa55	backend-ai	9000:9000 ↗	0.17%	3 hours ago	<input type="checkbox"/> ⋮ ⚡
<input type="checkbox"/>	utility-service	91c1f2fa2d4d	backend-ut	8083:8083 ↗	0.16%	3 hours ago	<input type="checkbox"/> ⋮ ⚡
<input type="checkbox"/>	eureka-server	1bb8a76f86e4	backend-el	8761:8761 ↗	1.17%	3 hours ago	<input type="checkbox"/> ⋮ ⚡
<input type="checkbox"/>	config-server	9cb37b06b622	backend-cr	8888:8888 ↗	0.25%	3 hours ago	<input type="checkbox"/> ⋮ ⚡
- Bottom Right:** Showing 13 items

# Jenkins Pipeline

Jenkins / Utility-billing-system #1

```
[INFO] --- spring-boot:3.2.0:repackage (default) @ notification-service ---
[INFO] Replacing main artifact C:\ProgramData\Jenkins\.jenkins\workspace\Utility-billing-system\backend\notification-service\target\notification-service-1.0.0.jar with repackaged archive, adding nested dependencies in BOOT-INF/.
[INFO] The original artifact has been renamed to C:\ProgramData\Jenkins\.jenkins\workspace\Utility-billing-system\backend\notification-service\target\notification-service-1.0.0.jar.original
[INFO] -----
[INFO] Reactor Summary for utility-billing-system 1.0.0:
[INFO]
[INFO] utility-billing-system ..... SUCCESS [ 0.449 s]
[INFO] config-server ..... SUCCESS [01:02 min]
[INFO] eureka-server ..... SUCCESS [ 14.277 s]
[INFO] api-gateway ..... SUCCESS [ 16.307 s]
[INFO] auth-service ..... SUCCESS [ 10.762 s]
[INFO] consumer-service ..... SUCCESS [ 6.648 s]
[INFO] utility-service ..... SUCCESS [ 6.039 s]
[INFO] meter-service ..... SUCCESS [ 7.886 s]
[INFO] billing-service ..... SUCCESS [ 14.074 s]
[INFO] payment-service ..... SUCCESS [ 12.325 s]
[INFO] notification-service ..... SUCCESS [ 12.273 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 02:48 min
[INFO] Finished at: 2026-01-06T21:59:22+05:30
[INFO] -----
[Pipeline] }
[Pipeline] // dir
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
```

# Jenkins Pipeline

 Jenkins / Utility-billing-system #1

Container auth-service Starting  
Container payment-service Started  
Container notification-service Started  
Container utility-service Started  
Container consumer-service Started  
Container meter-service Started  
Container api-gateway Started  
Container billing-service Started  
Container auth-service Started  
[Pipeline] }  
[Pipeline] // dir  
[Pipeline] }  
[Pipeline] // withEnv  
[Pipeline] }  
[Pipeline] // stage  
[Pipeline] stage  
[Pipeline] { (Declarative: Post Actions)  
[Pipeline] echo  
Utility Billing System deployed successfully using Docker Compose!  
[Pipeline] }  
[Pipeline] // stage  
[Pipeline] }  
[Pipeline] // withEnv  
[Pipeline] }  
[Pipeline] // withEnv  
[Pipeline] }  
[Pipeline] // node  
[Pipeline] End of Pipeline

Finished: SUCCESS

# Swagger Documentation

## auth-controller

^

**PUT** /auth/users/{userId}/reject

∨

**PUT** /auth/users/{userId}/approve

∨

**PUT** /auth/change-password

∨

**POST** /auth/reset-password

∨

**POST** /auth/register

∨

**POST** /auth/login

∨

**POST** /auth/forgot-password

∨

**GET** /auth/users

∨

**GET** /auth/users/{userId}

∨

**GET** /auth/users/pending

∨

**GET** /auth/profile

∨

**GET** /auth/pending-status

∨

# Swagger Documentation

Servers

http://localhost:8081 - Generated server url ▾

connection-controller ^

PUT /connections/{connectionId} ▾

PUT /connections/requests/{requestId} ▾

POST /connections ▾

POST /connections/request ▾

GET /connections/{consumerId} ▾

GET /connections/requests/pending ▾

GET /connections/internal/{connectionId} ▾

GET /connections/internal/billing-view ▾

GET /connections/internal/all ▾

consumer-controller ^