RESORT CRM

End-to-End Application Development & Deployment

Start Slide

Dev Ops Internal Presentation

resort-crm-by-dk.vercel.app

Project Overview

Resort CRM is a full-stack web application designed to manage resort-related operations using a scalable backend and a modern frontend.

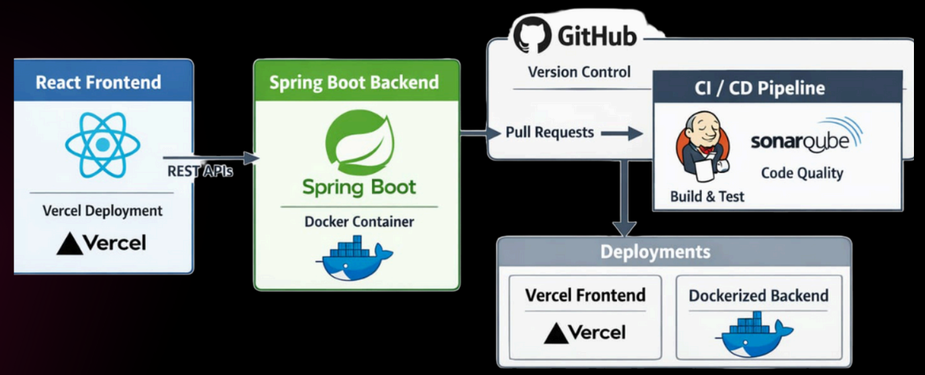
Objectives:

- Modular backend architecture
- Responsive frontend
- DevOps best practices
- Code quality and automation

System Architecture

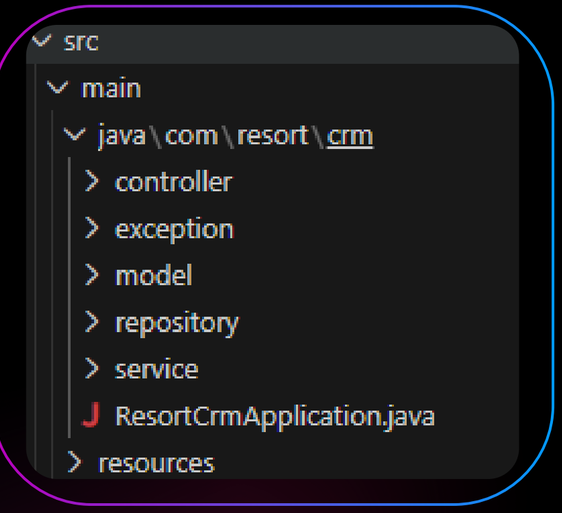
Architecture Flow:

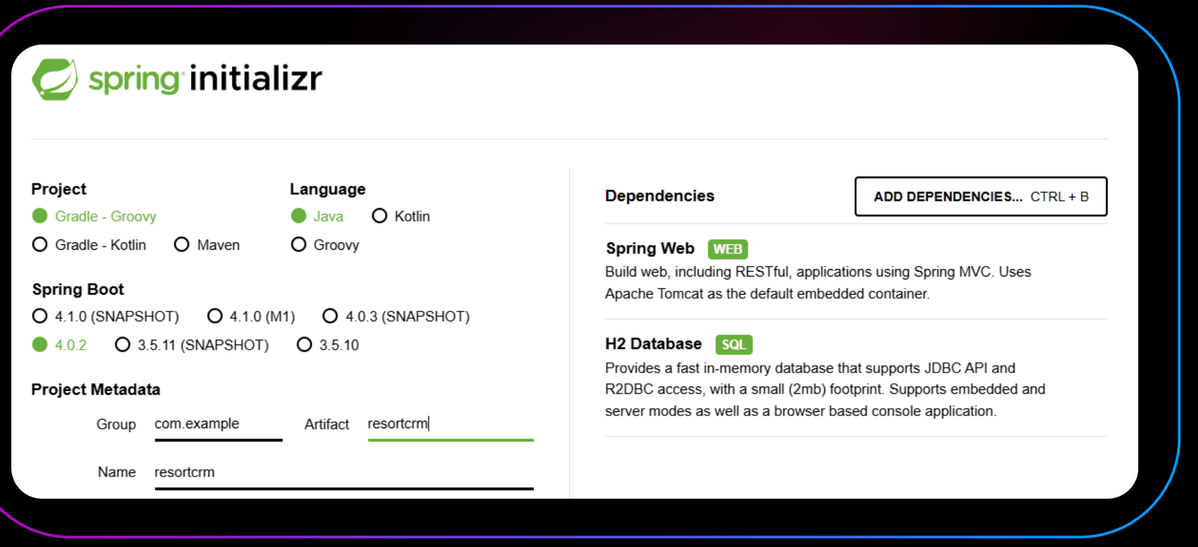
- React frontend communicates with Spring Boot backend via REST APIs
- GitHub used for version control
- Pull Request–based CI pipeline
- SonarQube for code quality checks
- Backend dockerized locally
- Frontend deployed on Vercel



Backend Development

- Project initialized using Spring Initializr
- Maven-based Spring Boot application
- Layered architecture:
- Controller
- Service
- Repository
- Entity
- REST APIs using JSON

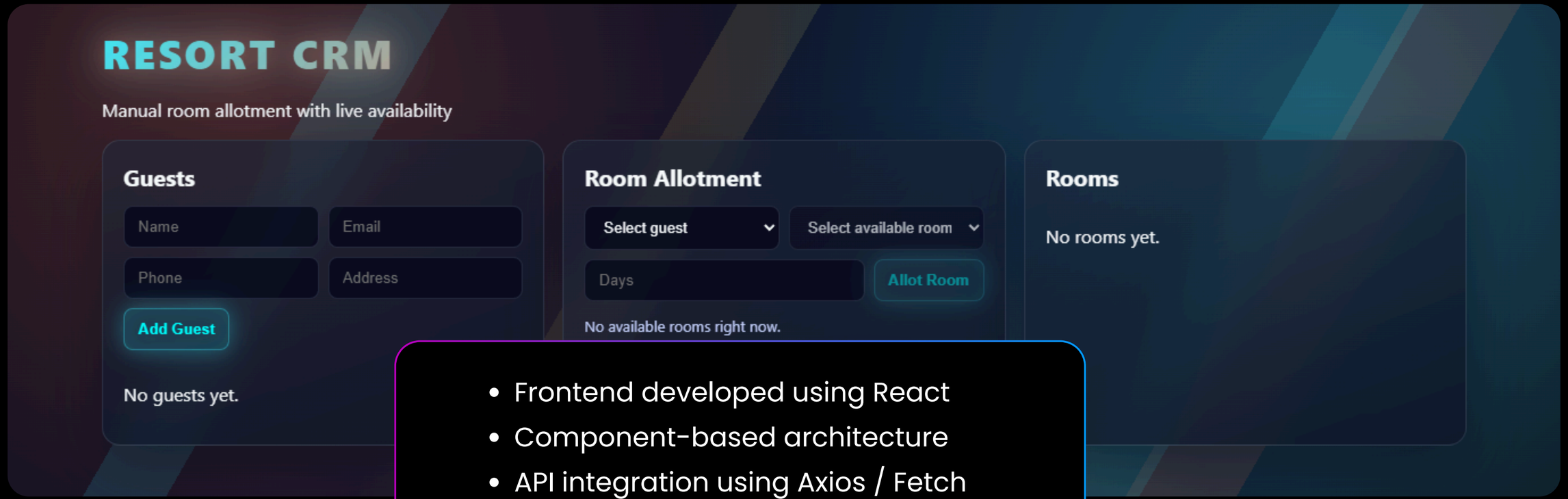




Backend Execution

- Application executed using: mvn spring-boot:run
- Backend accessible at: http://localhost:8080
- APIs tested using browser
- For Example : /api/rooms

Frontend Development

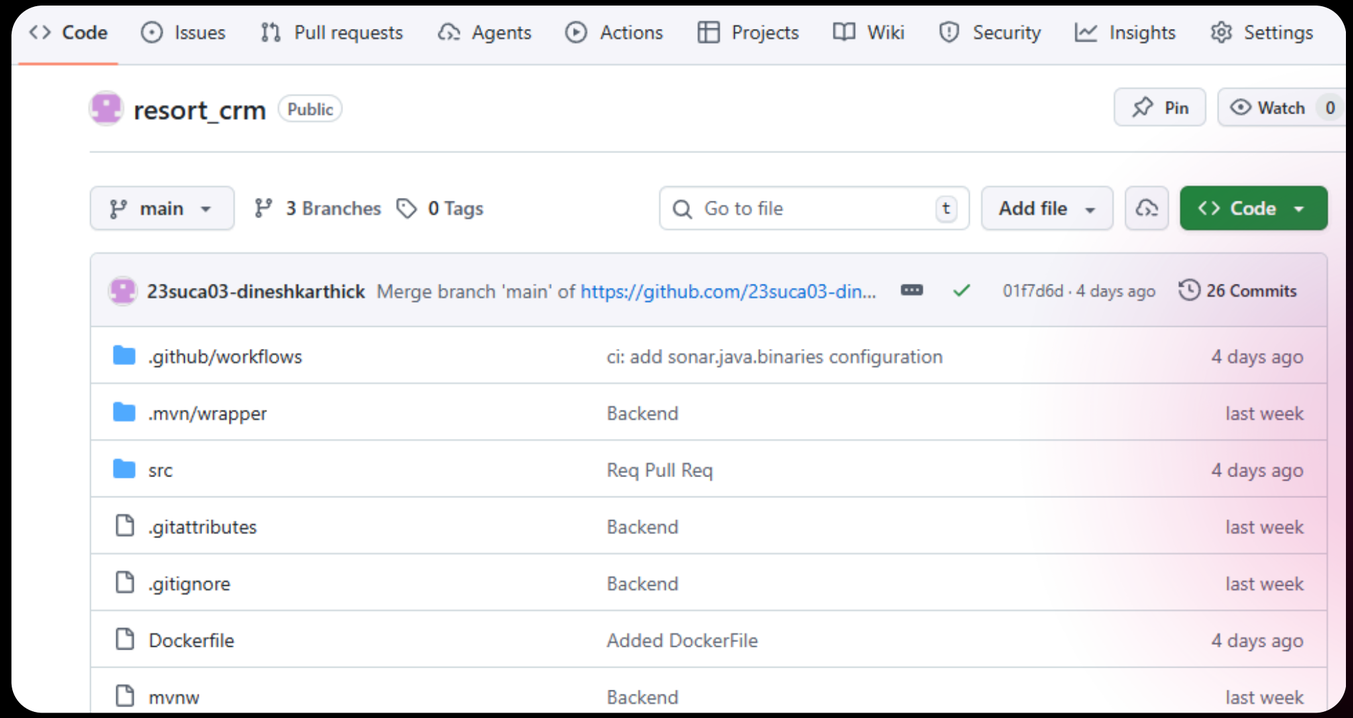


- Frontend developed using React
- Component-based architecture
- API integration using Axios / Fetch
- Local execution: npm start
- Runs on: http://localhost:3000

resort-crm-by-dk.vercel.app

Version Control & CI Pipeline

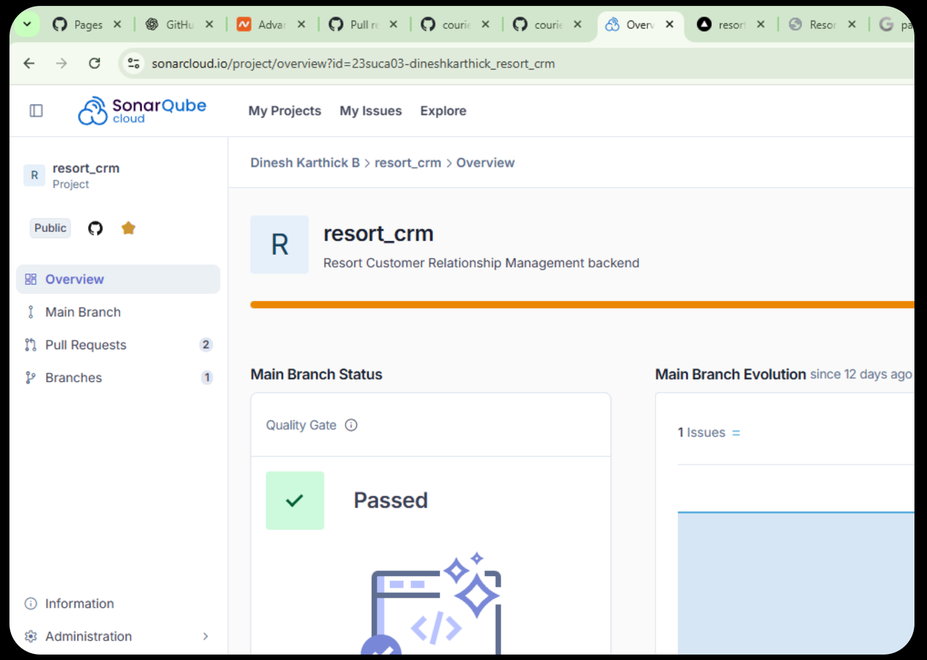
- Git used for source control
- GitHub repository with branch-based workflow
- Pull Requests used for merging changes
- CI pipeline triggered automatically on PR



resort-crm-by-dk.vercel.app

CI & SonarQube Integration

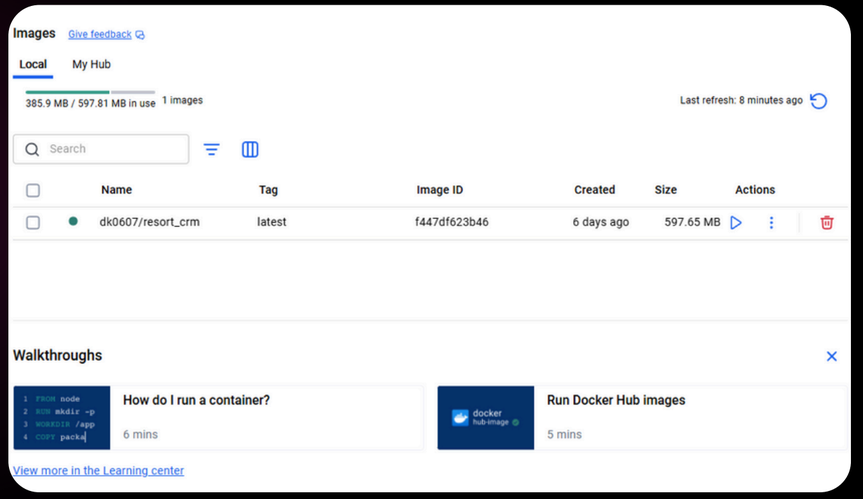
- SonarQube integrated into CI pipeline
- Static code analysis performed during PR
- Quality Gate must pass before merge
- Checks for:
  - Bugs
  - Code smells
  - Security issues



resort-crm-by-dk.vercel.app

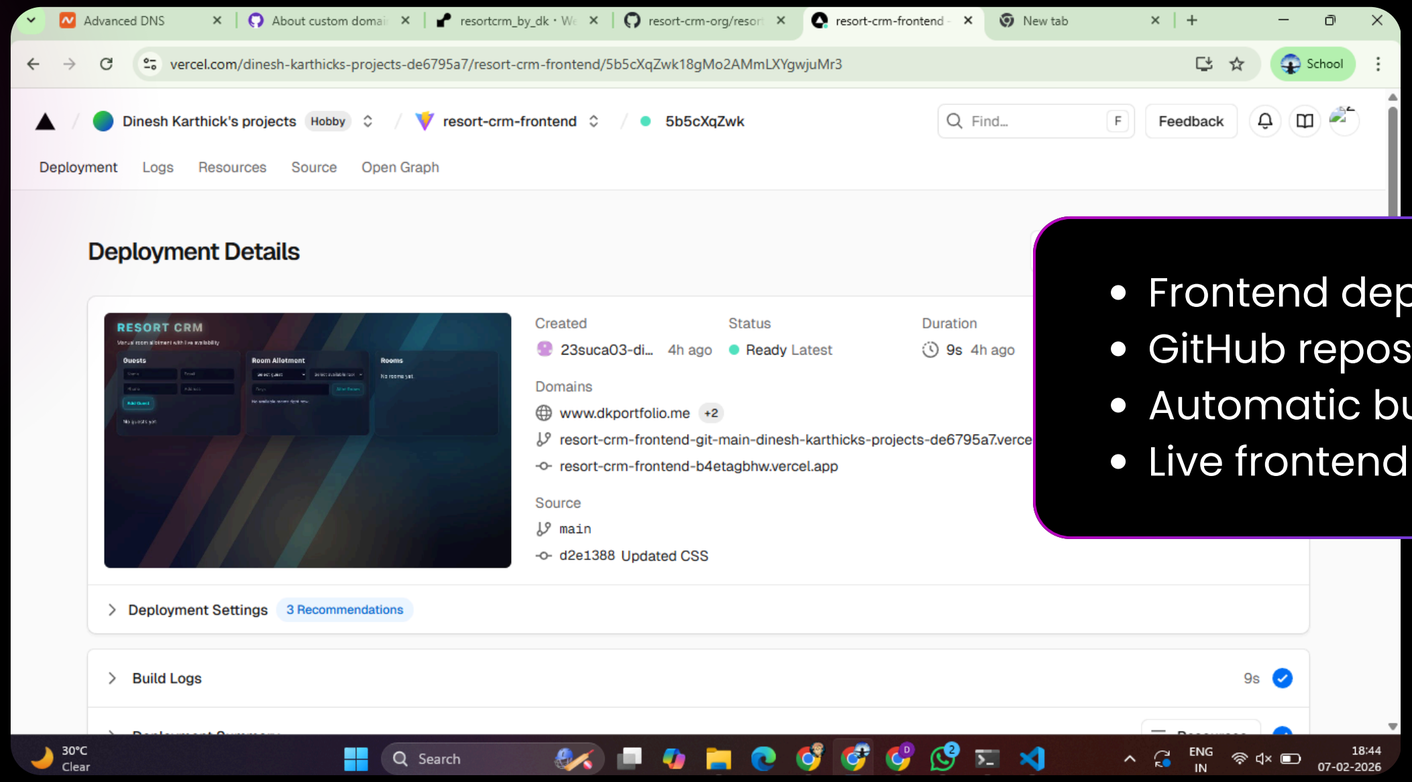
Backend Dockerization

- Spring Boot backend dockerized for local testing
- Dockerfile created using OpenJDK base image
- Backend verified inside Docker container
- Docker commands used:
  - docker build -t resort_crm_backend .
  - docker run -p 8080:8080 resort_crm_backend



resort-crm-by-dk.vercel.app

Frontend Deployment

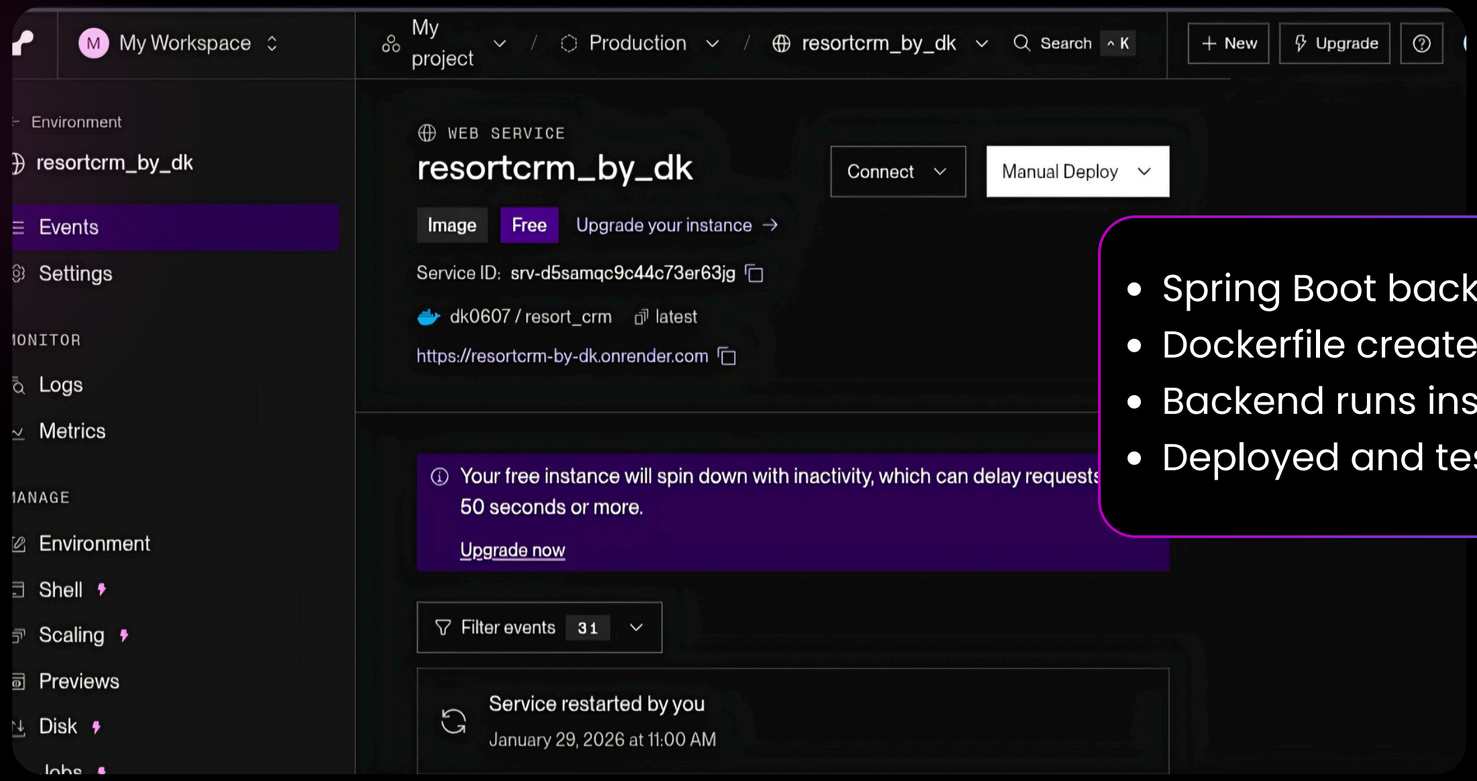


- Frontend deployed using Vercel
- GitHub repository connected to Vercel
- Automatic build and deployment
- Live frontend accessible via Vercel URL

resort-crm-by-dk.vercel.app

Resort CRM Docker Versal Render

Backend Deployment (Optional)



- Spring Boot backend containerized using Docker
- Dockerfile created with OpenJDK base image
- Backend runs inside Docker on port 8080
- Deployed and tested in local Docker environment

resort-crm-by-dk.vercel.app

Issues Faced & Solutions

Issues Faced

- Docker Desktop startup issues
- Port conflicts (8080 already in use)
- CORS errors between frontend & backend
- SonarQube quality gate failures

Solutions Implemented

- Enabled virtualization & restarted Docker services
- Resolved port conflicts by freeing/changing ports
- Configured CORS settings in Spring Boot
- Refactored code to meet SonarQube quality standards

Thank You

FOR YOUR ATTENTION