Hospitality App

Course Name: DevOps Foundation

Institution Name: Medicaps University – Datagami Skill Based Course

Student Name(s) & Enrolment Number(s):

Sr no	Student Name	Enrolment Number
1	Sumit Nigade	EN22CS301999
2	Vedang Rajoriya	EN22CS3011066
3	Vandana Patel	EN22CS3011057
4	Suryapratap Sisodiya	EN22CS3011004

Group Name: Group 08D9

Project Number:DO-08

Industry Mentor Name: Mr. Vaibhav

University Mentor Name: Dr. Ritesh Joshi

Academic Year:2025-26

Problem Statement & Objectives

1. Problem Statement

Many traditional applications are deployed using manual or semi-automated processes, which often lead to configuration inconsistencies, longer deployment times, and service downtime during updates. Such issues are critical in hospitality applications where service availability directly impacts customer satisfaction and business operations.

The problem addressed in this project is to implement a cloud-native deployment solution that automates the build and deployment process while ensuring high availability. The solution aims to eliminate downtime during updates and provide a scalable deployment architecture suitable for real-world hospitality platforms.

2. Project Objectives

The key objectives of this project are:

- To design and develop a modern hospitality web application
- To implement containerization using Docker for consistent deployment
- To automate build and deployment processes using CI/CD pipelines
- To deploy and manage containers using Kubernetes
- To host the application on AWS EC2 cloud infrastructure
- To achieve zero-downtime deployment using rolling update strategies
- To gain hands-on experience with industry-standard DevOps tools

3. Scope of the Project

The scope of this project includes frontend development, containerization, CI/CD automation, and cloud deployment. The application focuses on user interface, booking flow, and admin management features. Backend services such as authentication and database management are handled using Supabase.This project does not cover advanced payment gateway integration or large-scale load testing. However, the architecture is designed in a way that supports future scalability and enhancements.

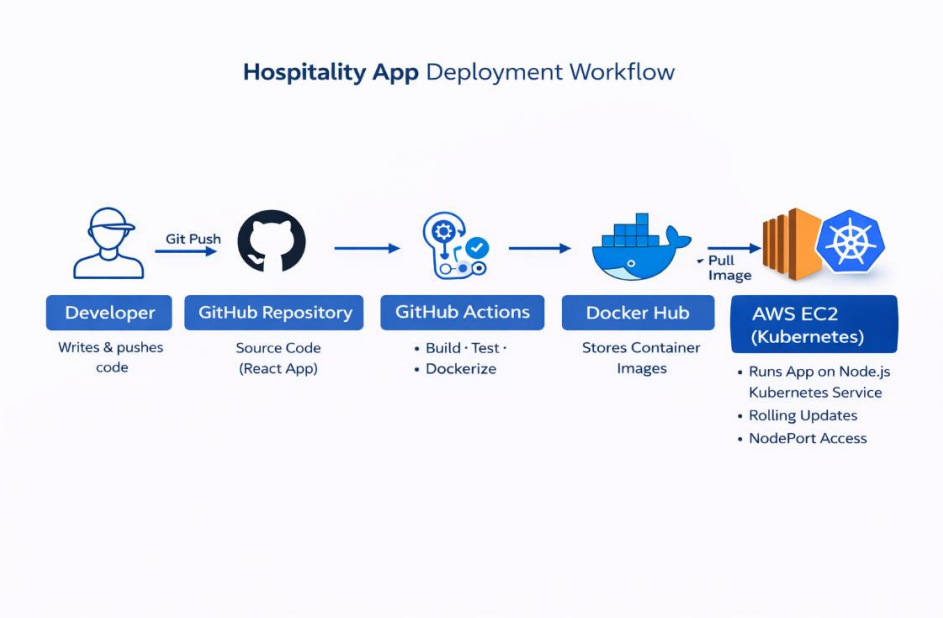
Proposed Solution

1. Key features

- User registration and login functionality
- Room browsing and booking interface
- Admin dashboard for managing room listings
- Responsive and mobile-friendly UI
- Containerized deployment using Docker
- Automated CI/CD pipeline using GitHub Actions
- Kubernetes-based orchestration
- Zero-downtime rolling updates
- Cloud deployment on AWS EC2

2. Overall Architecture / Workflow

The proposed system follows a cloud-native microservice-style architecture. The developer pushes source code to a GitHub repository, which triggers a CI/CD pipeline using GitHub Actions. The pipeline automatically installs dependencies, runs tests, and builds a Docker image. The image is pushed to Docker Hub for versioned storage.Kubernetes pulls the Docker image from Docker Hub and deploys it on an AWS EC2 instance. Rolling updates ensure that new versions of the application are deployed without affecting current users. The application is exposed to users through a Kubernetes service, making it accessible via a browser using the public IP address of the EC2 instance.

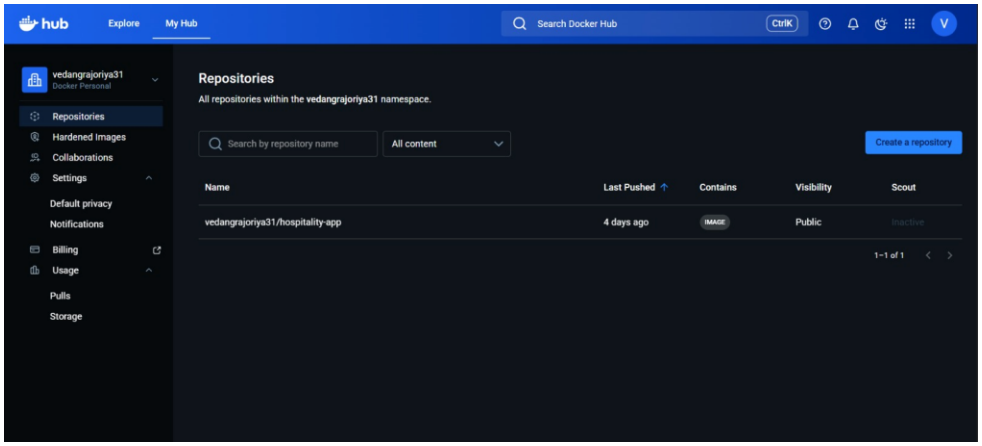


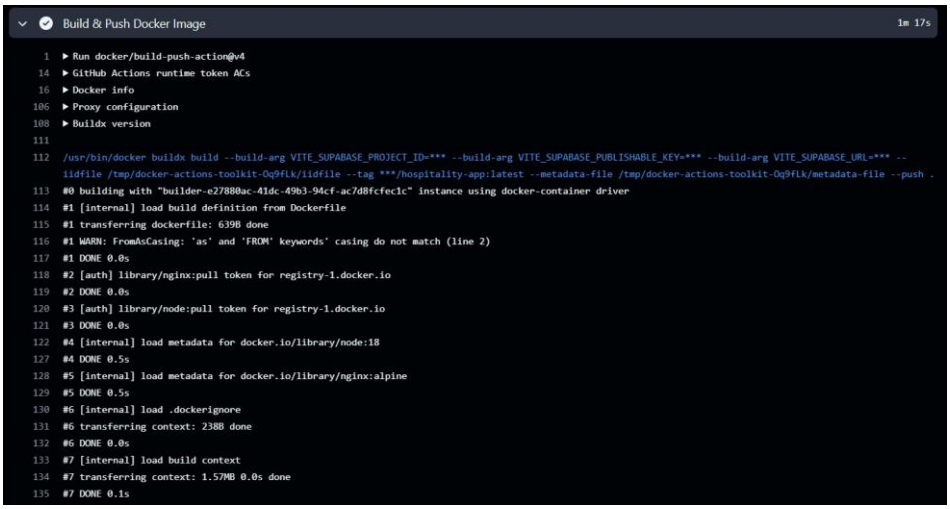
3. Tools & Technologies Used

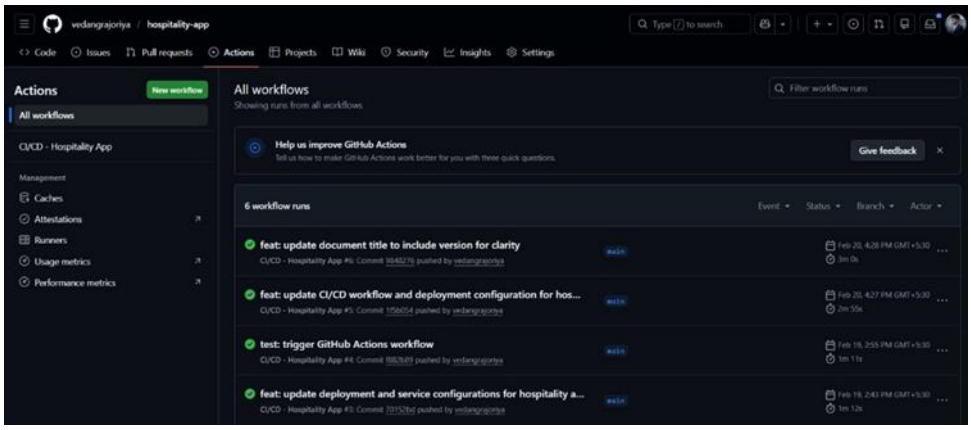
- **Frontend:** React (Vite + TypeScript), Tailwind CSS
- **Backend Services:** Supabase (Authentication & Database)
- **Containerization:** Docker
- **CI/CD:** GitHub Actions
- **Container Registry:** Docker Hub
- **Orchestration:** Kubernetes
- **Cloud Platform:** AWS EC2
- **Web Server:** Ubuntu
- **Version Control:** Git

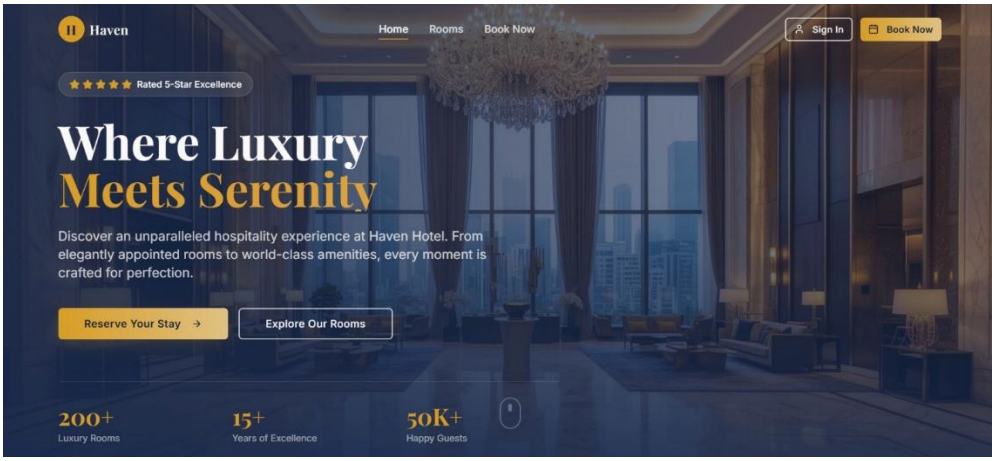
Results & Output

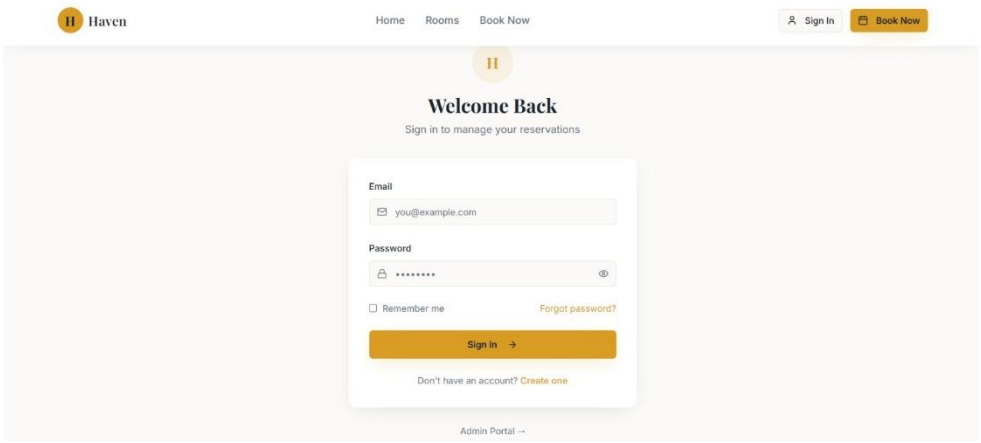
1. *Screenshots / outputs*

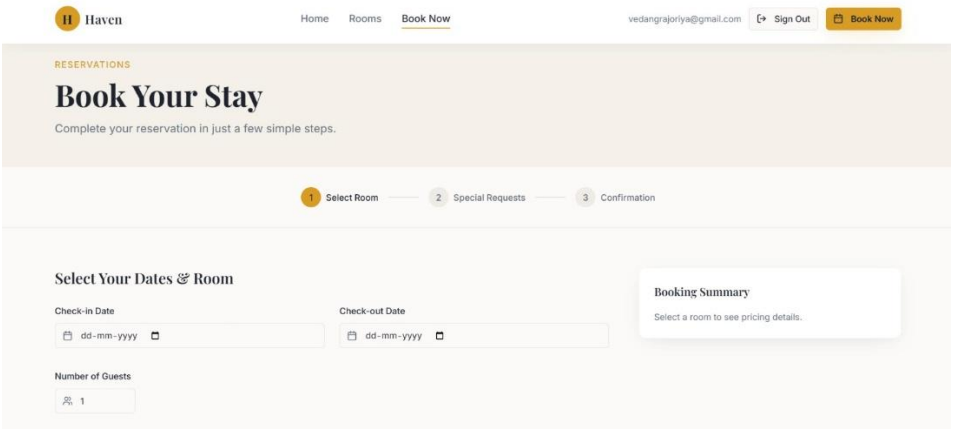



Build & Push Docker Image
1m 17s
▶ Run docker/build-push-action@v4
▶ GitHub Actions runtime token ACs
▶ Docker info
▶ Proxy configuration
▶ Buildx version
/usr/bin/docker buildx build --build-arg VITE_SUPABASE_PROJECT_ID=*** --build-arg VITE_SUPABASE_PUBLISHABLE_KEY=*** --build-arg VITE_SUPABASE_URL=*** --iidfile /tmp/docker-actions-toolkit-Oq9fLk/iidfile --tag ***/hospitality-app:latest --metadata-file /tmp/docker-actions-toolkit-Oq9fLk/metadata-file --push .
#0 building with "builder-e27880ac-41dc-49b3-94cf-ac7d8fcfec1c" instance using docker-container driver
#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile: 639B done
#1 WARN: FromAsCasing: 'as' and 'FROM' keywords' casing do not match (line 2)
#1 DONE 0.0s
#2 [auth] library/nginx:pull token for registry-1.docker.io
#2 DONE 0.0s
#3 [auth] library/node:pull token for registry-1.docker.io
#3 DONE 0.0s
#4 [internal] load metadata for docker.io/library/node:18
#4 DONE 0.5s
#5 [internal] load metadata for docker.io/library/nginx:alpine
#5 DONE 0.5s
#6 [internal] load .dockerignore
#6 transferring context: 238B done
#6 DONE 0.0s
#7 [internal] load build context
#7 transferring context: 1.57MB 0.0s done
#7 DONE 0.1s


vedangrajoriya / hospitality-app
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Actions
New workflow
All workflows
CI/CD - Hospitality App
Management
Caches
Attestations
Runners
Usage metrics
Performance metrics
All workflows
Showing runs from all workflows
Help us improve GitHub Actions
Give feedback
6 workflow runs
Event
Status
Branch
Actor
feat: update document title to include version for clarity
feat: update CI/CD workflow and deployment configuration for hos...
test: trigger GitHub Actions workflow
feat: update deployment and service configurations for hospitality a...


Haven
Home
Rooms
Book Now
Sign In
Book Now
Rated 5-Star Excellence
Where Luxury
Meets Serenity
Discover an unparalleled hospitality experience at Haven Hotel. From elegantly appointed rooms to world-class amenities, every moment is crafted for perfection.
Reserve Your Stay
Explore Our Rooms
200+
Luxury Rooms
15+
Years of Excellence
50K+
Happy Guests


Haven
Home
Rooms
Book Now
Sign In
Book Now
Welcome Back
Sign in to manage your reservations
Email
you@example.com
Password
Remember me
Forgot password?
Sign In
Don't have an account? Create one
Admin Portal →


Haven
Home
Rooms
Book Now
vedangrajoriya@gmail.com
Sign Out
Book Now
RESERVATIONS
Book Your Stay
Complete your reservation in just a few simple steps.
1 Select Room
2 Special Requests
3 Confirmation
Select Your Dates & Room
Check-in Date
dd-mm-yyyy
Check-out Date
dd-mm-yyyy
Number of Guests
1
Booking Summary
Select a room to see pricing details.

2. *Key outcomes*

- Successfully deployed a cloud-native hospitality application
- Automated the entire deployment process using CI/CD
- Ensured high availability with zero-downtime updates
- Improved deployment reliability and scalability
- Gained hands-on experience with cloud and DevOps tools

Conclusion

This project successfully demonstrates the deployment of a hospitality web application using modern cloud and DevOps technologies. The application was developed with a focus on scalability, automation, and high availability. By using Docker for containerization, the application achieved consistency across different environments, reducing deployment-related issues.

The implementation of a CI/CD pipeline using GitHub Actions automated the build and deployment process, minimizing manual intervention and improving reliability. Kubernetes was used to manage container orchestration and implement rolling updates, ensuring zero downtime during application updates. This approach is especially important for hospitality applications where uninterrupted service is essential.

Deploying the application on AWS EC2 provided a reliable cloud infrastructure with global accessibility and scalability. Overall, this project helped in gaining practical experience with containerization, CI/CD automation, Kubernetes orchestration, and cloud deployment. The project reflects real-world DevOps practices and provides a strong foundation for building and deploying production-ready applications.

Future Scope & Enhancements

- Integrate a custom backend service using frameworks such as Django or FastAPI
- Deploy the application using managed Kubernetes services like AWS EKS
- Implement monitoring and logging tools such as Prometheus and Grafana
- Enhance security using role-based access control and Kubernetes secrets
- Add payment gateway integration for online bookings
- Introduce real-time notifications for booking confirmations
- Implement auto-scaling and load balancing to handle high traffic
- Develop a mobile application for improved accessibility
- Optimize performance through caching and database indexing
- Add analytics and reporting features for business insights