PRANAV ANKIT RAJBHANDARI

Kathmandu, Nepal
Email: pranav.rjb79@gmail.com
GitHub: github.com/pranavrjb
LinkedIn: www.linkedin.com/in/pranavrajbhandari
Portfolio: www.pranavankitrajbhandari.com.np/

PROFESSIONAL SUMMARY

DevOps enthusiast with hands-on experience in CI/CD, containerization, and deploying applications using modern DevOps tools. Fast learner with strong debugging skills, eager to join a DevOps internship or junior level role to apply and grow technical skills in real-world projects.

TECHNICAL SKILLS

DevOps & Tools: Linux, Docker, Docker Compose, Jenkins, SonarQube, Ansible, Kubernetes, AWS, Terraform, Prometheus, Grafana, Trivy

Programming & Scripting: Python, JavaScript, Bash, YAML, Groovy, HTML, CSS

Version Control: Git, GitHub, Bitbucket

Core Skills: CI/CD Pipelines, Containerization, Monitoring, Automation

---

PROJECTS

Dockerized MERN Online Booking Platform

A full-scale online booking platform was designed and containerized using the MERN stack (MongoDB, Express.js, React, Node.js), ensuring consistent, portable, and easily deployable environments across development and production stages.

- Containerized both the React frontend and Node/Express backend using separate Dockerfiles to ensure consistent and isolated environments.
- Orchestrated the entire MERN stack using Docker Compose, including backend, frontend, and MongoDB services with shared networking and persistent volumes.
- Stored and managed Docker images in Harbor Registry for secure storage, versioning, and easy promotion across environments.

- Enabled a portable, scalable, and fully containerized setup that simplifies development, deployment, and environment management.

GitHub Repository: https://github.com/pranavrjb/docker-task

CI/CD Pipeline for Django TO-DO Application (Jenkins)

A complete CI/CD pipeline was developed for a Django TO-DO application using Jenkins, Docker, and Ansible. The application was containerized using both a Docker file and a docker-compose.yml file to efficiently define the Django service and its dependencies.

- Containerized the application with Dockerfile and Docker Compose to make the setup consistent and easy to run anywhere.
- Set up Jenkins to automatically pull the latest code from GitHub, build the Docker image, and push it to Docker Hub.
- Used Ansible for automated deployment, where Jenkins triggered a playbook on a remote server to pull the new image and start the updated container.
- Simplified the whole workflow from code changes to deployment making the release process faster, smoother, and more reliable.

GitHub Repository: https://github.com/pranavrjb/Django-Note-App

---

Automated CI/CD Pipeline for MERN Online Booking Platform

A fully automated CI/CD pipeline was implemented for an online booking platform built with the MERN stack (MongoDB, Express.js, React.js, Node.js), enhancing the development and deployment process to be faster, consistent, and more reliable.

- **Pipeline Automation:** Built the entire pipeline using Jenkins, Docker, Ansible, and GitHub, so every code change automatically triggered build, test, and deployment stages.
- **Containerization:** Dockerized both frontend (React) and backend (Node/Express) services, allowing each to run in isolated, consistent environments. Images were pushed to Harbor Registry, making deployment versioned and portable.
- **Monitoring & Security:** Integrated Prometheus & Grafana to monitor app performance and Trivy for automatic vulnerability scanning of Docker images.
- **Automated Deployment:** Used Ansible to deploy the application on Vagrant VMs automatically. The pipeline also sent notifications for build success or failure, ensuring smooth, reliable releases.

This setup simplified multi-service orchestration, improved deployment reliability, and ensured the application could run consistently across development, staging, and production environments.

GitHub Repository: https://github.com/pranavrjb/Medpulse

Java Expense Management Application – Deployment using Docker & AWS EC2

A fully containerized deployment workflow was created for a Java-based Expense Management application built with Spring Boot, Thymeleaf, and MySQL. The application was optimized using a multi-stage Dockerfile and deployed alongside its database using Docker Compose. The complete setup was hosted on an AWS EC2 instance, enabling fast, consistent, and scalable deployments with reduced image size and improved reliability.

- Developed and containerized a Java-based Expense Management application using a **multi-stage Dockerfile** to optimize image size and improve build efficiency.
- Created a **Docker Compose setup** to orchestrate the application container along with its dependent services.
- Deployed the application on an **AWS EC2 Linux instance**, configuring security groups, Docker runtime, and environment variables for production use.
- Automated build and deployment workflow using multi-stage Docker builds (builder and runtime stages), ensuring faster deployments and minimal image footprint.
- Ensured smooth application accessibility by configuring port mapping, system services, and persistent storage as needed.

GitHub Repository: https://github.com/pranavrjb/Expenses-app

HOBBIES

Traveling | Watching Movies | Listening to Music | Playing Cricket | Playing Mobile Games | Attending Community Events.

EDUCATION

Bachelor's in Information Management (BIM)
Tribhuvan University, Kathmandu, Nepal
Expected Graduation: 2026

CERTIFICATES

Development and Operations (DevOps) Training
March 3 to July 16, 2025
TechAxis Nepal.

DevOps Bootcamp by TWS
November 23, 2025

AWS DevOps and Agentic AI Masterclass by TWS
November 24, 2025

REFERENCE

TechAxis Nepal
Kumaripati Lalitpur, Nepal.
Contact No: 01-5437592

Mr. Suryaraj Timsina
Sr. DevOps Engineer at Quickfox Consulting.
Instructor at TechAxis
Email: timsinasuryaraj@hotmail.com
Contact No: 9843723188