ProShop E-Commerce Platform

Project Overview

ProShop is a **full-stack MERN (MongoDB, Express.js, React, Node.js) e-commerce application** with a comprehensive **DevOps pipeline, cloud-native infrastructure, and enterprise-grade monitoring stack**. This project demonstrates modern software development practices, combining application development with enterprise-level DevOps methodologies, containerization, orchestration, cloud deployment strategies, and **comprehensive observability using Prometheus and Grafana**.

Application Stack

Frontend

- **React.js** with Redux Toolkit for state management
- **React Router** for client-side routing
- **React Bootstrap** for responsive UI components
- **RTK Query** for efficient data fetching and caching
- **PayPal SDK** integration for payment processing

Backend

- **Node.js** with Express.js framework
- **MongoDB** with Mongoose ODM for data modeling
- **JWT (JSON Web Tokens)** for authentication
- **bcrypt.js** for password hashing
- **Multer** for file upload handling
- **Cookie-based session management**
- **prom-client** for Prometheus metrics exposition

Core Features

- Full shopping cart functionality with persistent storage
- User authentication and authorization (JWT-based)
- Product catalog with search, pagination, and filtering
- Product reviews and rating system
- Admin dashboard for product, user, and order management

- PayPal payment integration
- Order tracking and delivery status management
- Image upload and management system

DevOps Technology Stack

1. Containerization

- **Docker**: Multi-stage Dockerfile for optimized production builds
- **Docker Compose**: Local development orchestration for MongoDB and application services
- Layer caching optimization for faster builds
- Production-ready container images with minimal attack surface

2. Container Orchestration

- **Kubernetes**: Production-grade container orchestration
- **StatefulSets**: For MongoDB and Prometheus with persistent storage
- **Deployments**: For stateless application components
- **Services**: ClusterIP for internal communication, NodePort for external access
- **ConfigMaps & Secrets**: Secure configuration and sensitive data management
- **Persistent Volume Claims**: For database and metrics data persistence
- **DaemonSets**: For node-level monitoring (node-exporter)

3. Infrastructure as Code (IaC)

- **Terraform**: Complete AWS infrastructure provisioning
  - **AWS EKS (Elastic Kubernetes Service)** cluster creation
  - **VPC** with public and private subnets across multiple availability zones
  - **NAT Gateways** for secure outbound internet access
  - **Security Groups** with least-privilege network policies
  - **IAM Roles and Policies** for service accounts (IRSA)
  - **S3 backend** for Terraform state management
  - **DynamoDB** for state locking
  - **AWS Load Balancer Controller** via Helm
  - **EBS volumes** for Prometheus and Grafana persistent storage

4. CI/CD Pipeline

- **GitLab CI/CD**: Comprehensive pipeline automation
  - **Multi-stage pipeline**: Install, Test, Build, Security Scan, Push, Deploy, Notify
  - **Dependency caching**: For faster build times

    - **Parallel job execution**: Frontend and backend testing
    - **Conditional execution**: Feature branch vs. main branch workflows
    - **Artifact management**: Docker image storage and versioning
    - **Automated deployments**: GitOps-style Kubernetes manifest updates

5. Security Scanning

- **Trivy**: Container vulnerability scanning
    - Image scanning for CVEs (HIGH and CRITICAL severity)
    - Filesystem scanning for dependency vulnerabilities
    - JSON and text report generation
    - Integration into CI pipeline with non-blocking warnings
- **npm audit**: Node.js dependency vulnerability scanning for both frontend and backend

6. Configuration Management

- **Ansible**: Automated deployment and infrastructure management
    - Cross-platform support (Debian/Ubuntu, RHEL/CentOS/Fedora)
    - Docker installation and configuration
    - Docker Compose stack deployment
    - Idempotent playbook design
    - Environment-specific variable management

7. Monitoring & Observability Stack

Prometheus - Metrics Collection & Storage

- **Time-series database** for metrics storage
- **Pull-based model** for scraping metrics from targets
- **Service Discovery**: Automatic discovery of Kubernetes services
- **PromQL**: Powerful query language for metrics analysis
- **Alertmanager**: Alert routing and notification management

Grafana - Visualization & Dashboards

- **Multi-datasource support**: Prometheus
- **Pre-built dashboards**: Kubernetes cluster, Node exporter, MongoDB
- **Custom dashboards**: Application-specific metrics visualization
- **Alerting**: Visual alert configuration and management
- **Dashboard provisioning**: GitOps-managed dashboard definitions