

DevOps CI/CD Pipeline for ABC Retail Application

A complete DevOps implementation transforming ABC's traditional retail application into a modern, cloud-native solution with continuous integration and deployment.



by **Vaibhav oza**



Project Setup & Goals



Objective

Build a complete DevOps pipeline from code push to deployment and monitoring.



Business Goal

Transform ABC's traditional retail app into a modern, CI/CD-enabled cloud-native application.



Initial Setup

Java, Maven, Git, Docker, Jenkins, Ansible installed on WSL2 Ubuntu.

GitHub Integration



Code Repository

Project source code pushed to GitHub repository: [devOPS_Project1](#).



Documentation

Comprehensive screenshots and documentation folders created as supporting evidence.



Project Guides

Readme and Markdown files generated to explain project details and usage.



CI/CD Pipeline with Jenkins



Clone Repository

Pull latest code from GitHub.



Build WAR

Compile application using Maven.



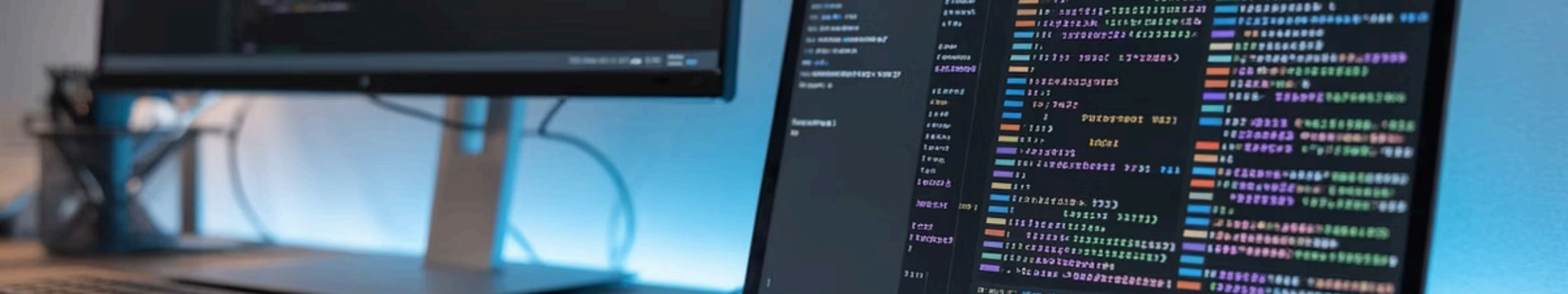
Build Docker Image

Create abc-retail-app container image.



Deploy with Ansible

Trigger playbook to launch container.



Docker Integration

Create Dockerfile

Use Tomcat 9 base image and copy WAR into webapps/ROOT.war.

Build Image

Run `docker build -t abc-retail-app:latest` command.

Run Container

Launch container on port 9090 and verify accessibility.

Ansible Automation

Create Playbook
Write run_docker.yml to automate container deployment.

Verify Deployment
Confirm container is running properly.

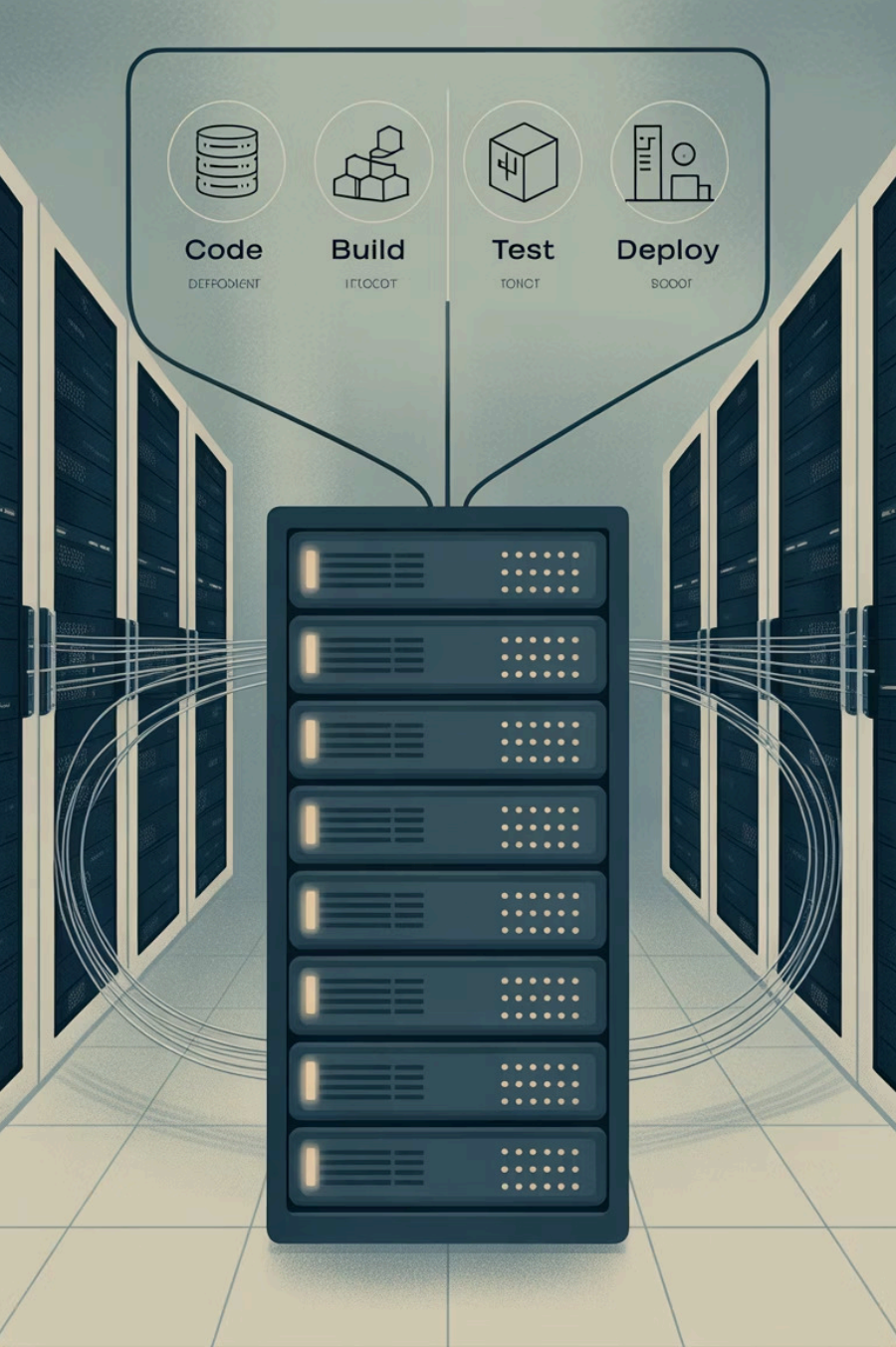


Jenkins Integration

Connect Ansible with Jenkins pipeline.

Execute Playbook

Run automation to launch container.



Kubernetes Deployment



Minikube Setup

Start Minikube using Docker driver.



YAML Configuration

Create deployment.yaml and service.yaml files.



Service Exposure

Access app via minikube service retail-service.

Monitoring with Prometheus

- Installed on port 9091
- Node Exporter on port 9100
- Configured prometheus.yml

Monitoring with Grafana

Grafana is used to visualize and analyze the metrics collected by Prometheus for the ABC Retail application.

It provides interactive dashboards that display real-time data on application performance, resource usage, and system health.

Configured to connect to the Prometheus data source, enabling team members to monitor and troubleshoot effectively.



Documentation & Evidence

Comprehensive documentation includes screenshots, markdown guides, and organized project structure with dedicated folders for each component available in the github repository.

Final Outcome

100%

Automation

Full CI/CD pipeline from code to deployment

24/7

Monitoring

Real-time metrics via Grafana dashboards

3x

Scalability

Improved application scaling with Kubernetes

