Group 9: Tech Innovations

Online Bookstore Microservices

Docker Microservices - Building scalable applications with ease

PRESENTED TO
Prof. Pedram Habibi

PRESENTED BY
Group 9 - NAA

Our Team

Nicholas Ilechie
(Group Leader)

Jiyad Mohammed Arif
Shaikh

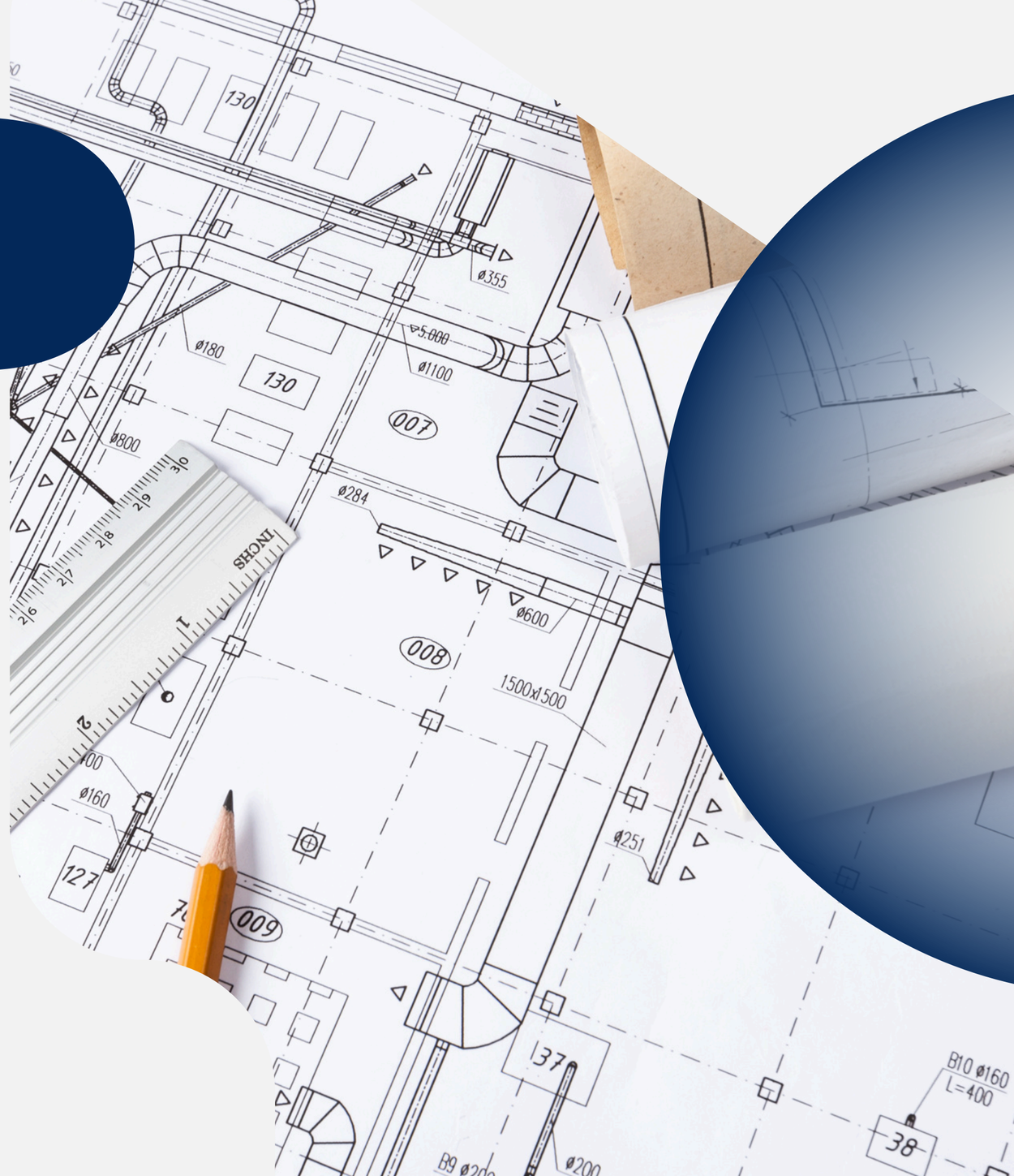
Nirajbhai Ranchhodbhai
Limbasiya

Helly Rajeshbhai Patel

AGENDA

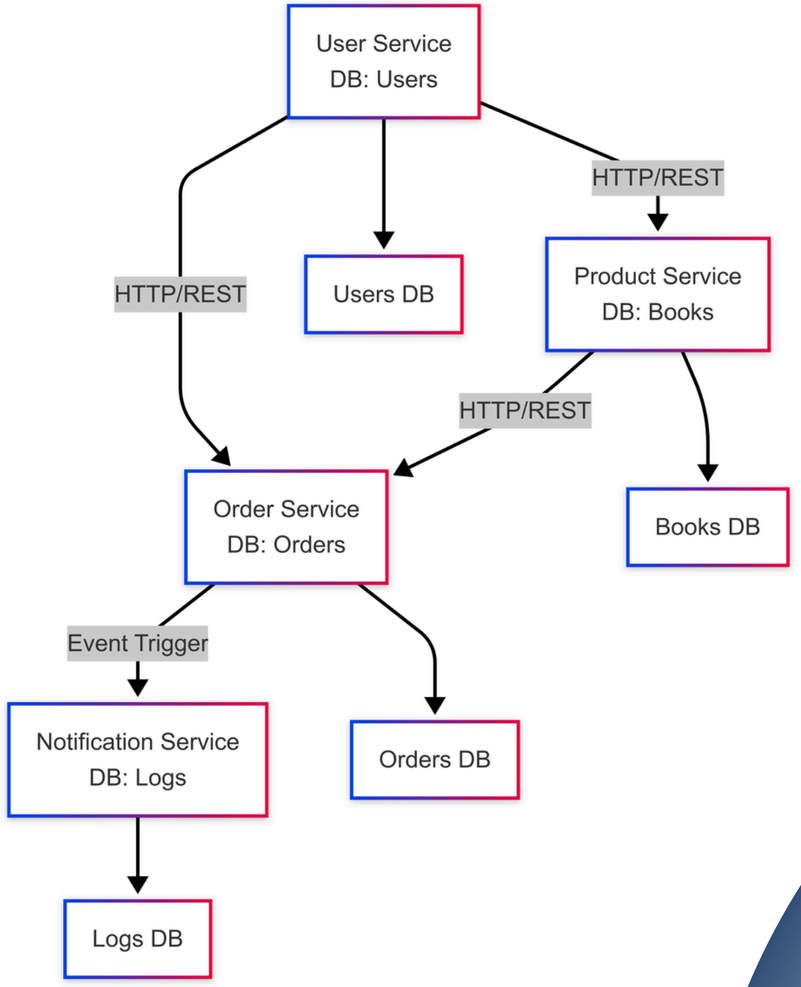
Project Overview

- **Goal:** Microservices-based online bookstore
- **Services:** User (registration, profiles), Product (book catalog)
- **Tech Stack:** Node.js, Docker, Kubernetes (Minikube), GitHub Actions, Prometheus, Grafana, Postman
- **Development Phases:** Design/Dockerization, Kubernetes, Testing/CI-CD, Presentation



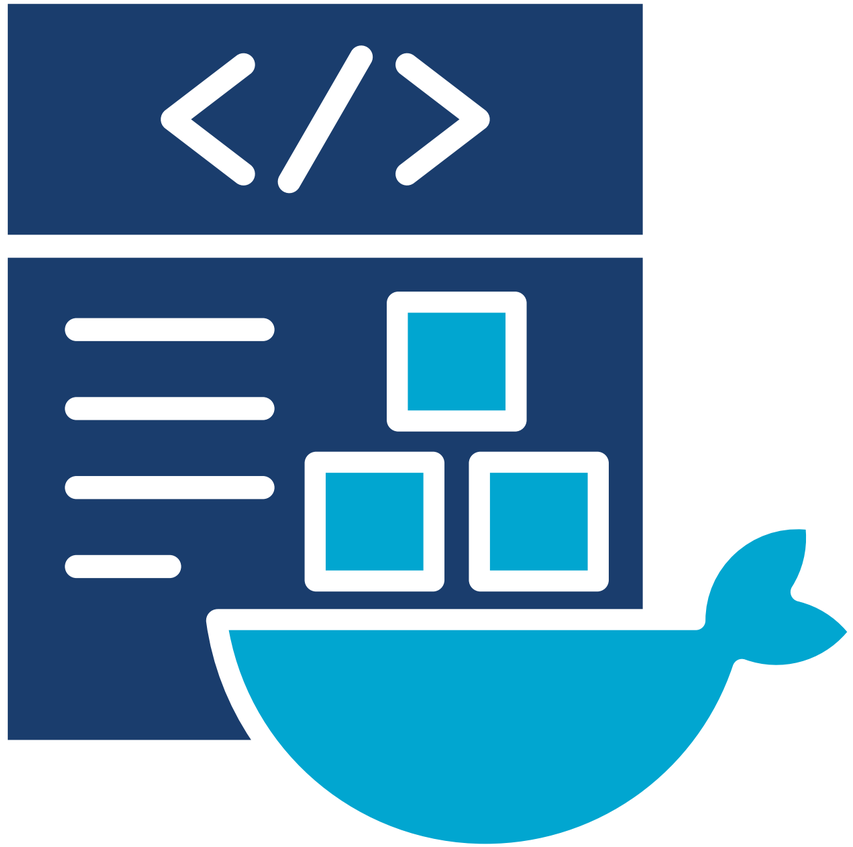
Architecture and Design

- **Modular Services:** User (POST /users/register, GET /users), Product (GET /books)
- **RESTful APIs:** HTTP communication via Kubernetes DNS
- **Persistence:** In-memory storage per service
- **Documentation:** README files, architecture diagram



Docker Implementation

- **Dockerfiles:** Lightweight Node.js images
- **Docker Compose:** Local testing with docker-compose.yml
- **Security:** ConfigMaps/Secrets for sensitive data



Kubernetes Implementation

- **Manifests:** Deployments, Services, ConfigMaps, Secrets in bookstore namespace
- **Auto-scaling:** HPA on User Service (70% CPU threshold, 2-5 replicas)
- **Networking:** ClusterIP for inter-service communication
- **Monitoring:** Prometheus/Grafana for CPU/memory metrics

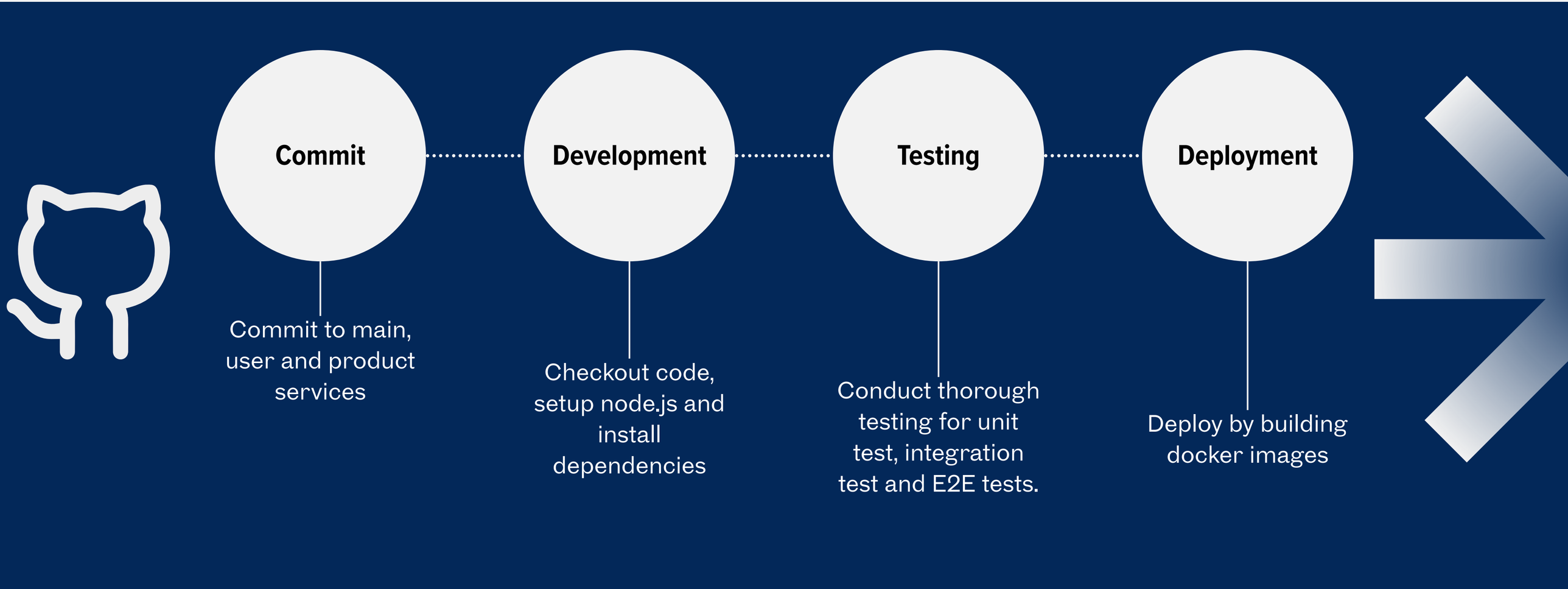
API Testing with Postman

- **Tool:** Postman for RESTful API validation
- **Endpoints Tested:**
  - **User:** POST /users/register, GET /users
  - **Product:** GET /books
- **Results:** 201 Created, 200 OK responses

Testing Strategy

- **Unit Tests:** userController.test.js, productController.test.js
- **Integration Tests:** user-product.test.js
- **End-to-End Tests:** bookstore.test.js
- **Status:** All tests passing successfully

CI/CD Pipeline



Challenges Encountered & Lessons Learned

Lessons Learned

- Microservices Architecture: Modular design, RESTful APIs, SOLID principles
- Containerization & Orchestration: Docker containers, Kubernetes deployment, HPA scaling
- Testing Methodologies: Unit, integration, E2E testing, API validation with Postman
- DevOps Practices: CI/CD automation with GitHub Actions
- Debugging Skills: kubectl logs, describe pod commands for troubleshooting

Challenges Encountered

- Testing Issues: Fixed array reset problems in User Service tests
- CI/CD Pipeline: Removed problematic Minikube step in GitHub Actions
- Kubernetes Deployment: Resolved CreateContainerConfigError issues

Thank You

Questions & Discussion

View Project on GitHub