

# **Bahmni HIS + AI + WhatsApp + GitHub Agent**

## **95-Step Realistic Roadmap with DevOps & Governance**

### **Phase 0 — Repo Setup & Design (Steps 1–5)**

Create a GitHub repository with initial README and roadmap.  
Add .gitignore and Docker Compose skeleton for Bahmni services.  
Prepare CI/CD skeleton with GitHub Actions workflow file.  
Define staging and production branches with branch protection rules.  
Document architecture and security guidelines in the repo.

### **Phase 0.1 — Infrastructure Planning (Steps 6–10)**

Select DigitalOcean as the cloud provider for deployment.  
Provision a droplet with 4GB RAM, 2 vCPUs, and 80GB SSD.  
Estimate patient load and size resources accordingly.  
Plan future scaling with additional droplets or managed DB.  
Document infrastructure design and sizing decisions.

### **Phase 1 — Infrastructure & Base Setup (Steps 11–20)**

Configure swap file and firewall (UFW).  
Create a non-root user with sudo privileges.  
Configure SSH access using password initially, with plan for keys.  
Install fail2ban for brute force protection.  
Update system packages and apply security patches.  
Install Docker CE on the server.  
Install Docker Compose plugin.  
Verify Docker and Compose installation with test containers.  
Set up persistent storage directories for Bahmni data.  
Reserve volumes for DB and logs.

## **Phase 2 — ChatGPT Agent Mode Integration (Steps 21–25)**

Enable ChatGPT Agent Mode in your OpenAI account.

Connect Agent Mode with your GitHub repository.

Allow Agent Mode to manage Docker Compose configs for Bahmni.

Use Agent Mode to automate module setup (OpenMRS, OpenELIS, Odoo ERP).

Monitor and validate Agent Mode outputs (human-in-the-loop supervision).

## **Phase 3 — Bahmni HIS Core Modules (Steps 26–37)**

Clone Bahmni Docker Compose setup from official repo.

Create a .env file with service credentials and passwords.

Deploy Bahmni services with Docker Compose.

Verify OpenMRS (Bahmni EMR) is accessible via browser.

Configure OpenMRS patient registration module.

Enable Bahmni Appointments module.

Configure OpenELIS laboratory system integration.

Enable Odoo ERP module for billing and inventory.

Configure pharmacy workflows in Odoo ERP.

Load initial medical dictionaries (ICD-10, LOINC, drug lists).

Set up role-based access (doctor, nurse, lab tech, admin, finance).

Add Docker healthchecks for all services.

## **Phase 3.1 — Database Management (Steps 38–42)**

Deploy MySQL for OpenMRS EMR database.

Deploy PostgreSQL for Odoo ERP database.

Deploy PostgreSQL or MySQL for OpenELIS laboratory system.

Configure database connection pooling for performance.

Set up nightly automated dumps for all databases.

## **Phase 4 — Clinical Workflows & Forms (Steps 43–52)**

Configure patient demographics and identifiers.

Set up outpatient encounter forms.

Configure inpatient admission workflows.

Enable diagnosis capture in encounter forms.  
Add clinical observations to Bahmni forms.  
Configure lab order entry from EMR to OpenELIS.  
Enable results return from OpenELIS to Bahmni EMR.  
Configure pharmacy prescription forms.  
Set up billing workflows for outpatient visits.  
Enable financial invoicing workflows for inpatient care.

## **Phase 5 — AI Integration in Bahmni (Steps 53–63)**

Deploy OpenAI connector inside n8n.  
Configure secure API keys for ChatGPT.  
Connect Bahmni form submissions to AI via middleware API.  
Enable AI suggestions for preliminary diagnoses.  
Enable AI recommendations for lab tests.  
Configure AI prescription support (drug, dose, allergy check).  
Add AI summarization for clinical encounter notes.  
Configure AI-driven triage forms for intake.  
Enable audit logging of all AI outputs.  
Validate AI suggestions with clinician review.  
Add caching to reduce repeated API costs.

## **Phase 6 — n8n Agent Automation (Steps 64–74)**

Deploy n8n as a Docker service on the same droplet.  
Configure persistent volumes for n8n workflows.  
Secure n8n with HTTPS via reverse proxy (Nginx/Traefik).  
Create webhook endpoints for Bahmni → n8n.  
Build automation for patient appointment reminders.  
Automate follow-up reminders via n8n scheduler.  
Configure GitHub integration node in n8n.  
Build workflow for agent to open GitHub PRs.  
Enable workflow for automatic Docker redeploy on updates.  
Document all workflows for hospital IT staff.  
Store all API keys and tokens in Docker secrets or Vault.

## **Phase 7 — WhatsApp Integration (Steps 75–84)**

Create Meta Business account for WhatsApp API.  
Verify business number with Meta Cloud API.  
Configure permanent system access tokens.  
Deploy WhatsApp integration node in n8n.  
Submit notification templates for Meta approval.  
Connect Bahmni appointment data to WhatsApp notifications.  
Enable lab result notifications to patients via WhatsApp.  
Configure clinician alert messages for critical cases.  
Build WhatsApp chatbot for basic triage Q&A.;  
Set up fallback SMS or email if WhatsApp fails.

## **Phase 8 — GitHub + Agent Coding (Steps 85–90)**

Use branch protection to require human approval for merges.  
Configure GitHub Actions CI/CD with staging and production environments.  
Run automated tests for Bahmni containers and DB migrations.  
Enable n8n workflows for automated PR creation.  
Allow ChatGPT agent to suggest code changes via GitHub PRs.  
Automate Docker redeployment from GitHub main branch.

## **Phase 9 — Reporting & Analytics (Steps 91–93)**

Configure Bahmni dashboards for patient statistics.  
Build lab utilization and turnaround reports.  
Enable billing and finance dashboards in Odoo ERP.

## **Phase 10 — Governance, Training & Monitoring (Steps 94–95)**

Define data privacy, retention, and access policies for hospital compliance.  
Train staff on HIS usage, escalation, and incident handling.