# Azure Cloud Operations & Monitoring Hub — Project Plan

## 1) Executive Summary

The Azure Cloud Operations & Monitoring Hub is a simulated enterprise-grade environment built to demonstrate Azure administration, automation, and governance skills. This project plan outlines the delivery approach, phases, objectives, deliverables, and timelines for building, deploying, and operating the solution.

## 2) Project Objectives

- Establish a secure, automated, and observable Azure environment.  
- Apply Infrastructure as Code (IaC) and CI/CD automation.  
- Integrate monitoring, alerting, and self-healing automation.  
- Demonstrate cost management and sustainability principles.  
- Document a full enterprise-style operations workflow.

## 3) Scope

In Scope:  
- Web application and backend API hosted on Azure App Service.  
- Infrastructure automation via Bicep or Terraform.  
- GitHub Actions CI/CD for deployment automation.  
- Azure Front Door, Key Vault, and Log Analytics integration.  
- Monitoring, alerting, and AIOps automation with Logic Apps.  
- Cost management dashboards and lifecycle optimization.  
  
Out of Scope:  
- Non-Azure platforms (AWS/GCP).  
- Multi-tenant SaaS scaling or customer data migration.

## 4) Approach & Methodology

The project follows a phased approach, where each phase builds on the previous one. Initial deployment will occur through the Azure Portal for validation, followed by Infrastructure as Code for repeatability. Continuous integration and delivery will be implemented via GitHub Actions. Monitoring and automation will be layered last to create a complete operations ecosystem.

## 5) Phase Plans

### Phase 1 — Core Infrastructure & App

Objective: Establish the initial environment and deploy a minimal web application.  
  
Tasks:  
- Create Resource Group and baseline resources (Storage, App Service, Front Door, Key Vault).  
- Deploy sample app manually using VS Code.  
- Verify Front Door routing and HTTPS configuration.  
  
Deliverables:  
- Live app accessible via Front Door.  
- Verified secure endpoint (HTTPS enforced).  
- Manual deployment documentation.

### Phase 2 — Infrastructure as Code (IaC) + CI/CD

Objective: Automate environment deployment and establish DevOps pipeline.  
  
Tasks:  
- Develop Bicep/Terraform templates for all Azure resources.  
- Validate deployment with Azure CLI (what-if, plan, apply).  
- Set up GitHub Actions for infrastructure and app deployment.  
- Automate CDN purge after each deployment.  
  
Deliverables:  
- IaC templates in /infra directory.  
- GitHub Actions pipelines for infra and app.  
- Automated deployments with logs and validation.

### Phase 3 — Monitoring & Observability

Objective: Implement monitoring and logging to track system performance.  
  
Tasks:  
- Configure Log Analytics and Application Insights.  
- Create alert rules for availability, 5xx errors, CPU, and latency.  
- Build Azure Monitor dashboards and Workbooks.  
- Enable diagnostic settings on key resources.  
  
Deliverables:  
- Log Analytics workspace and linked insights.  
- Active alerting system.  
- Operational dashboard and monitoring workbook.

### Phase 4 — Automation & AIOps

Objective: Introduce automation for self-healing and intelligent operations.  
  
Tasks:  
- Create Logic App triggered by alerts to perform remediation actions.  
- Automate restart, scaling, or notifications based on thresholds.  
- (Optional) Add Azure OpenAI Function to summarise incidents or generate reports.  
  
Deliverables:  
- Logic App automation flow.  
- Self-healing and notification pipeline.  
- (Optional) AI-driven incident summary feature.

### Phase 5 — Cost & Sustainability

Objective: Integrate cost management and sustainability policies.  
  
Tasks:  
- Implement Azure Budgets and cost alerts.  
- Apply lifecycle rules for Storage (Hot → Cool → Archive).  
- Schedule shutdowns for non-production environments.  
- Develop cost and energy-efficiency dashboard.  
  
Deliverables:  
- Cost budget with alerts.  
- Lifecycle management policy.  
- Sustainability metrics dashboard.

## 6) Timeline Overview

Estimated Duration: 4–6 weeks (adjustable based on testing and learning pace)  
  
Phase 1: 4–5 days  
Phase 2: 7–10 days  
Phase 3: 5–7 days  
Phase 4: 5–7 days  
Phase 5: 3–5 days

## 7) Deliverables Summary

- Deployed Azure environment with automated operations.  
- Infrastructure as Code templates.  
- GitHub Actions pipelines.  
- Monitoring dashboards and automation workflows.  
- Documentation (Design + Plan).

## 8) Roles & Responsibilities

Project Owner: Azz (Azure Admin & Developer)  
Responsibilities:  
- Design architecture and IaC templates.  
- Configure Azure resources and automation.  
- Maintain documentation and testing.  
- Manage CI/CD pipelines and cost control.

## 9) Risks & Mitigation

- Risk: Cost overrun → Mitigation: Budgets and cleanup schedules.  
- Risk: Misconfigurations → Mitigation: IaC validation and peer review.  
- Risk: Alert fatigue → Mitigation: Optimize alert thresholds.  
- Risk: Security gaps → Mitigation: RBAC, Key Vault, and HTTPS enforcement.

## 10) Success Metrics

- 100% deployment automation (no manual clicks in production).  
- System uptime > 99.5%.  
- Alerts trigger actions within 2 minutes.  
- Monthly cost optimization > 15% savings through automation.  
- All documentation and dashboards complete.