# Azure Cloud Operations & Monitoring Hub — Final Design Document

## 1) Project Objective

This project simulates a real-world Azure production environment designed to demonstrate professional-level cloud operations, monitoring, security, automation, and governance. Unlike the Cloud Resume Challenge, this solution focuses on managing a complete operational system rather than hosting a personal website.

The environment showcases an administrator’s ability to design, deploy, secure, automate, and maintain cloud infrastructure. It represents how an Azure Administrator or Cloud Engineer operates a live system with scalability, observability, and cost efficiency.

## 2) Key Objectives

- Design a scalable, secure Azure infrastructure for a simulated enterprise app  
- Implement Infrastructure as Code (Bicep/Terraform) for automation and repeatability  
- Enable CI/CD using GitHub Actions  
- Integrate monitoring, alerting, and AIOps automation  
- Apply governance, role-based security, and cost management  
- Incorporate sustainability and green cloud principles

## 3) Project Scope

Core Components (MVP):  
- App: Web frontend + minimal API (e.g., visitor or status endpoint)  
- Infra: Azure Storage, App Service, Front Door, Key Vault, VNet, NSGs, Bastion  
- Security: Entra ID RBAC, Managed Identities, HTTPS enforcement  
- Monitoring: Log Analytics, Application Insights, Alerts, Workbooks  
- Automation: Logic App or Function for self-healing actions  
  
Advanced Components (Optional):  
- AI-Ops integration with Azure OpenAI (incident summaries or predictions)  
- Cost automation, lifecycle rules, and sustainability dashboards  
- Edge/CDN optimization and regional replication

## 4) Architecture Overview

Architecture Flow:  
GitHub (CI/CD) → Azure Front Door → App Service (API/Web) → Data Layer (Table, Cosmos, or SQL) → Monitoring → Automation & AIOps

The system includes Front Door for global routing, App Service for application hosting, a secured data layer, monitoring via Log Analytics and Insights, and a Logic App for automated operational responses.

## 5) Resource Inventory

Resource Group: rg-ops-hub-prod  
Front Door: afd-ops-prod  
App Service Plan: asp-ops-prod  
App Service: app-ops-web-prod  
Key Vault: kv-ops-prod  
Storage Account: stopsprod  
Data Store: sql-ops-prod or cosmos-ops-prod  
Virtual Network: vnet-ops-prod (snet-app, snet-data, snet-mgmt)  
Bastion Host: bastion-ops-prod  
Monitoring: laws-ops-prod, App Insights linked  
Automation: logic-ops-autoresponse / func-ops-autofix

## 6) Security & Access

- Managed Identity for secure app-to-service access  
- Key Vault for secrets; restricted to admin and MI  
- Network Security Groups for segmented access  
- Front Door WAF for OWASP rule protection  
- RBAC and principle of least privilege for access control  
- HTTPS enforced with HSTS via Front Door

## 7) CI/CD and Infrastructure as Code

- IaC via Bicep/Terraform stored in /infra  
- GitHub Actions pipeline for deployment and validation  
- Automatic CDN purge on successful deployment  
- Manual approval gate for production apply step

## 8) Observability & AIOps

- App Insights and Log Analytics for telemetry and logs  
- Alert Rules: availability, error rate, CPU, and cost anomalies  
- Logic App automation for alert response: notify → restart/scale → log incident  
- Optional Azure OpenAI Function for summarising incident logs

## 9) Cost & Sustainability

- Azure Budgets for cost alerting at 50/80/100% thresholds  
- Off-hours auto-shutdown schedule for non-prod systems  
- Storage lifecycle rules (Hot → Cool → Archive)  
- Regional placement for renewable energy preference  
- Cost workbook dashboard for transparency

## 10) Delivery Phases

Phase 1 — Core Infrastructure & App  
Phase 2 — Infrastructure as Code + CI/CD  
Phase 3 — Monitoring & Observability  
Phase 4 — Automation & AIOps  
Phase 5 — Cost & Sustainability

## 11) Acceptance Criteria

- All resources deployed via IaC with no manual drift  
- CI/CD pipelines functioning end-to-end  
- Secrets securely stored in Key Vault  
- Alerts trigger notifications and automated actions  
- Monitoring dashboard operational with key metrics visible

## 12) Risks & Mitigations

- Secret leakage → use Managed Identity + Key Vault  
- Cost overruns → budgets + lifecycle policies  
- Access misconfigurations → enforce RBAC reviews and least privilege

## 13) Deliverables

/infra → Bicep/Terraform templates  
/app → Web + API code  
.github/workflows → App + Infra pipelines  
README.md → Documentation, diagrams, and deployment notes  
Dashboard workbook → Azure Monitor JSON export

## 14) Push Command

git add .  
git commit -m "finalized cloud ops design and documentation"  
git push