

POC 03: Smart Building Management Platform - Monitoring, Backup and Cost Optimization

Prepared for: **Saurabh Dindokar**
Role: **AWS DevOps Engineer**
Public-safe project name: **Smart Building Management Platform**

Confidentiality Note

This document is intentionally public-safe. It does not include real client names, internal project names, organization-owned source code, private URLs, IP addresses, credentials, account IDs, screenshots, or any confidential implementation details.

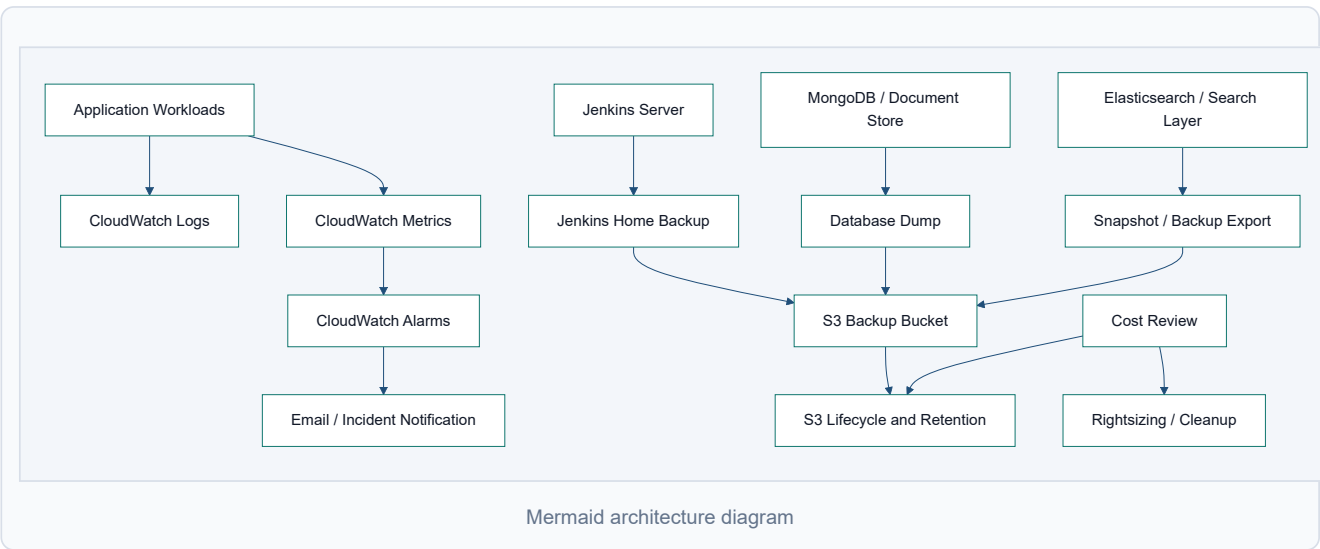
Work Sample Summary

Public-safe work sample covering monitoring, alerting, backup automation, retention control, Jenkins backup, database backup, log visibility, and practical cost optimization checks.

Problem Statement

- â€¢ Operational teams needed better visibility into compute, deployment, backup, and application health.
- â€¢ Backup activity required documented scripts, retention behavior, and restoration verification.
- â€¢ Cloud resources needed periodic cost review for unused or oversized components.

Mermaid Architecture Diagram



My Responsibilities

- â€¢ Prepared backup and retention approach for CI/CD server data and database components.
- â€¢ Documented CloudWatch dashboards, alarms, log groups, and troubleshooting checkpoints.
- â€¢ Created cost optimization checklist covering unused resources, storage lifecycle, right sizing, and backup retention.

â€¢ Defined restore verification steps so backup success is not assumed without validation.

Implementation Approach

â€¢ Backup scripts create timestamped archives and store them in a controlled S3 path.

â€¢ Retention rules keep recent backups while reducing long-term storage cost.

â€¢ CloudWatch alarms are mapped to actionable failure conditions, not noisy generic metrics only.

â€¢ Cost review focuses on EC2/EBS snapshots, S3 lifecycle, idle resources, and oversized compute.

Reliability, Security and Operations Focus

â€¢ Health checks, validation steps, and rollback planning were treated as part of deployment quality, not afterthoughts.

â€¢ Secrets, access boundaries, private networking, backup retention, and monitoring visibility were documented wherever applicable.

â€¢ The POC is written so another engineer can understand the flow, reproduce the approach, and extend it safely without exposing confidential data.

Validation Checklist

Review architecture and confirm each component has a clear responsibility.

Validate deployment or automation steps in a non-production environment first.

Confirm logs, health checks, backup behavior, and rollback steps are documented.

Confirm access, secrets, and network boundaries follow least-privilege expectations.

Capture final runbook notes for handover and support readiness.

Expected Outcomes

â€¢ Improved backup confidence through restore validation.

â€¢ Better operational visibility for incidents.

â€¢ Reduced avoidable cloud spend through retention and cleanup planning.

â€¢ Clear support runbook for monitoring and backup ownership.

Technologies Used

CloudWatch, S3, Jenkins, Shell Scripting, Elasticsearch, MongoDB, AWS Cost Explorer, EC2, EBS, IAM

Naukri Work Sample Description

Public-safe DevOps POC by Saurabh Dindokar covering architecture, responsibilities, implementation approach, validation checklist, operational focus, and measurable delivery outcomes. The document uses generic project naming and does not disclose any confidential client or organization information.