**Deployment Plan Document**

**Project Title:** HealthHub

**Document Title:** Deployment Plan

**Version:** 1.0

**Date:** 08.08.2024

**1. Introduction**

**1.1 Purpose:** The purpose of this document is to outline the deployment plan for HealthHub, detailing the deployment environments, process, and tools.

**1.2 Scope:** This document covers the deployment of the HealthHub system to development, staging, and production environments, along with rollback and recovery procedures.

**2. Deployment Environments**

**2.1 Development Environment:**

* **Description:** Local development environment for testing and debugging.
* **Tools:** Docker, Docker Compose, Local PostgreSQL, Node.js, React development server

**2.2 Staging Environment:**

* **Description:** Pre-production environment for final testing before deployment to production.
* **Tools:** AWS EC2, RDS for PostgreSQL, S3 for static assets, CloudFront for CDN, Docker, Kubernetes

**2.3 Production Environment:**

* **Description:** Live environment where the application will be available to users.
* **Tools:** AWS EC2, RDS for PostgreSQL, S3 for static assets, CloudFront for CDN, Docker, Kubernetes

**3. Deployment Process**

**3.1 Steps for Deployment:**

1. **Development Environment:**
   * Set up Docker and Docker Compose.
   * Develop and test the application locally.
   * Ensure local PostgreSQL is running and accessible.
2. **Staging Environment:**
   * Deploy the application to AWS EC2 instances.
   * Set up RDS for the database.
   * Store static assets in S3 and configure CloudFront for CDN.
   * Conduct final testing in the staging environment.
3. **Production Environment:**
   * Deploy the application to AWS EC2 instances.
   * Set up RDS for the database.
   * Store static assets in S3 and configure CloudFront for CDN.
   * Monitor the application for performance and issues.

**4. Rollback and Recovery Procedures**

**4.1 Rollback Procedures:**

* **Identify Issue:** Quickly identify and assess the issue requiring rollback.
* **Initiate Rollback:** Use version control (e.g., Git) to revert to the previous stable version.
* **Database Rollback:** If necessary, restore the database from the latest backup.
* **Verify:** Verify the rollback by running tests and ensuring the application is stable.
* **Communicate:** Inform stakeholders and users about the rollback and its impact.

**4.2 Recovery Procedures:**

* **Backup Regularly:** Ensure regular backups of the database and application code.
* **Disaster Recovery Plan:** Have a disaster recovery plan in place with clear steps and responsible parties.
* **Testing:** Regularly test the recovery procedures to ensure they work as expected.
* **Documentation:** Keep detailed documentation of the recovery procedures and update them regularly.