**Take-Home Exam: Scalable Notification System**

**Objective:**
Design a scalable notification system that can send notifications via multiple channels (e.g., email, SMS, push notifications). The system should be able to handle a high volume of notifications and ensure that notifications are delivered reliably. Additionally, the services should be containerized and the design should consider deployment in AWS.

**Requirements:**

1. **Notification Channels:**
    - Implement at least two notification channels (e.g., email and SMS).
    - Each channel should have its own configuration and delivery mechanism.
2. **Scalability:**
    - The system should be able to handle a high volume of notifications.
    - Use a message queue to manage notification delivery and ensure scalability.
3. **Reliability:**
    - Implement retry logic for failed notifications.
    - Ensure that notifications are delivered at least once.
4. **Extensibility:**
    - Design the system in a way that allows easy addition of new notification channels in the future.
5. **Logging and Monitoring:**
    - Implement logging for notification delivery attempts and failures.
    - Provide a way to monitor the system's performance and health.
6. **Containerization:**
    - Containerize the services using Docker.
    - Provide Dockerfiles for building the service images.
    - Use Docker Compose to define and run multi-container Docker applications.
7. **AWS Deployment Consideration:**
    - Document the AWS services that would be used for deployment.
    - Provide a high-level architecture diagram showing how the services would be deployed in AWS.
8. **Documentation:**
    - Provide clear documentation on how to set up and run the service.
    - Include API documentation for sending notifications.
    - Include a section on AWS deployment considerations.

**Deliverables:**

1. Source code for the notification system, including tests and documentation.
2. Dockerfiles for building the service images.
3. A Docker Compose file for running the services locally.
4. A README file with instructions on how to set up and run the service.
5. A brief report (1-2 pages) explaining your design decisions, any challenges faced, and how you addressed them.
6. Documentation on AWS deployment considerations, including a high-level architecture diagram.