Deployment Architecture

Overview

This document describes the deployment architecture for the weather service application. The system is designed for high availability, scalability, and reliability
using **Kubernetes**, **Dockerized microservices**, **message queues**, and **cloud infrastructure**.

Deployment Stack

- **Containerization:** Docker
- **Orchestration:** Kubernetes
- **Cloud Infrastructure:** AWS / Azure / GCP
- **CI/CD Pipeline:** GitLab + Docker + Helm (Auto Build, Test, and Deploy)
- **Monitoring & Logging:** ELK Stack, Prometheus
- **Caching Layer:** Redis
- **Message Queue:** Kafka
- **Security & Rate Limiting:** API Gateway + Custom Security Layer

Deployment Components

1. User Devices

- Mobile, Web, and other clients access the system.

2. Content Delivery Network (CDN)

- Ensures fast content delivery and reduces latency.

3. Load Balancer

- Handles auto-scaling and high availability.

4. API Gateway

- Manages traffic routing, authentication, and request validation.

5. Rate Limiter & Security Layer

- Protects against abuse and ensures fair usage.

6. Global Cache (Redis)

- Caches frequently accessed weather data for quick retrieval.

7. Message Queue (Kafka)

- Ensures asynchronous processing and event-driven architecture.

8. Kubernetes Cluster

- Manages the deployment of the following **Dockerized Services**:
  - **Profile Service** (Dockerized)
  - **Location Service**
  - **Humidity Service** (Dockerized)
  - **Temperature Service** (Dockerized)
  - **Wind Service** (Dockerized)
  - **Service Manager** (Dockerized)
  - **Notification Service** (Kafka Consumer) (Dockerized)
  - **Circuit Breaker** (Hystrix)

9. Weather API Service (3rd Party APIs)

- Fetches real-time weather data from external sources.

10. Logging & Monitoring

- Uses **ELK (Elasticsearch, Logstash, Kibana)** and **Prometheus** for tracking system health and performance.

11. CI/CD Pipeline

- Automates the build, testing, and deployment process using **GitLab CI/CD**, **Docker**, and **Helm**.

12. Cloud Infrastructure

- Deployed on **AWS, Azure, or GCP** for scalability, reliability, and security.