Architecture Specification

Overview

The architecture of the AutoSync Cloud Service is designed to ensure scalability, modularity, and high performance. It incorporates microservices for specific functionalities, distributed orchestration via Kubernetes, and fault-tolerant communication through Apache Kafka. This section details the system's architecture components, deployment topology, and data flow.

---

1. System Components

1.1 API Gateway

- **Technology:** Kong Gateway
- **Purpose:**
  - Routes incoming requests to appropriate microservices.
  - Handles rate-limiting, authentication, and logging.
- **Key Features:**
  - Integrates with OAuth2 and JWT for secure communication.
  - Supports load balancing across backend services.

1.2 Authentication Layer

- **Options Supported:**
  a. **Keycloak** for centralized identity and access management.
  b. **Firebase Authentication** for mobile-first use cases.
  c. **SSO (Single Sign-On):** Supports SAML and OpenID Connect.
  d. **Traditional Credentials:** Username/password with salted hashing via **BCrypt**.

1.3 Backend Services

- **File Sync Service:** Manages synchronization logic and interacts with storage backends.
- **Conflict Resolution Engine:** Resolves file conflicts using versioning, user priorities, and ML-based predictions.
- **Storage Adapter:** Interfaces with storage systems (AWS S3, Azure Blob, or on-prem MinIO).

1.4 Messaging Layer

- **Technology:** Apache Kafka
- **Purpose:**
  - Facilitates asynchronous communication between services.
  - Provides resilience and scalability for high-throughput sync operations.

1.5 Storage Solutions

- **Primary Storage:** Cloud object storage (AWS S3, Azure Blob).
- **Metadata Storage:** PostgreSQL with optimized indexing for file operations.
- **Caching Layer:** Redis for frequently accessed metadata and configurations.

1.6 Orchestration and Deployment

- **Technology:** Kubernetes with Helm.
- **Features:**

- Horizontal Pod Autoscaling (HPA).
- Rolling updates for zero downtime.
- Helm Charts for simplified deployment and configuration management.

---

2. Deployment Topology

2.1 Multi-Region Deployment

- The system supports multi-region deployment for disaster recovery and low-latency access.
- Uses cloud provider services such as AWS Route 53 for DNS routing.

2.2 Deployment Modes

1. **Cloud-Only Deployment:** Entire system hosted on a cloud provider.
2. **Hybrid Deployment:** Backend services deployed on-premise, while storage remains cloud-based.

---

3. Data Flow

3.1 High-Level Workflow

1. **User Interaction:**
   - The user logs in via the frontend (React.js) using their chosen authentication method (SSO, Firebase, etc.).
2. **API Gateway:**
   - The API Gateway validates the token and routes the request to the appropriate backend service.
3. **File Operations:**
   - File sync and metadata operations are handled by the Sync Service, which communicates with Kafka and the Storage Adapter.
4. **Storage Integration:**
   - Files are stored in the designated cloud backend (e.g., AWS S3) or on-prem storage systems.

Architecture Diagram

(Below is an example of a deployment diagram. A visual depiction will be generated for clarity.)