WindeskFM System Architecture

Overview

WindeskFM, designed with a microservices architecture, emphasizes scalability, flexibility, and resilience, functioning as a distributed system. Each microservice is an independent, deployable unit focused on a specific functionality, running in containers within a Kubernetes cluster. This design enables independent development and deployment of each component, offering high resilience and scalability.

System Communication & Management

- **Inter-service Communication:** Communication between services is facilitated through REST APIs and Kafka-based messaging, with load balancing and autoscaling managed by Kubernetes. The integration of CI/CD processes allows seamless updates and feature deployments, ensuring uninterrupted functionality.
- **Observability:** Utilizing Kubernetes' observability tools, service communication, performance monitoring, and error tracking are closely monitored. This architecture enables rapid intervention when necessary, allowing the system to scale with demand and maintain high availability for critical operations.

Architectural Design

1. **Microservices Structure:** Each microservice operates as a distinct functional unit, orchestrated by Kubernetes to handle resource management, load balancing, service discovery, and fault tolerance, enabling both horizontal and vertical scaling.

2. **Node Configuration:** The architecture includes three master nodes managing central operations, with multiple worker nodes designated for specific tasks:
   - Application workloads (APP)
   - Infrastructure services (INFRA)
   - Business intelligence processes (BI)
   - Database operations (DB)

3. **Security & Authentication:**
   - Firewall: Incoming and outgoing traffic is regulated by a firewall, ensuring initial security.

- DNS and Load Balancing: External and internal DNS services, along with load balancers, direct user requests to the correct microservices.
- Keycloak Integration: For user authentication and authorization, Keycloak provides a centralized mechanism using OAuth2, SAML, and OpenID Connect protocols. Keycloak's compatibility with LDAP and Active Directory simplifies integration with enterprise environments.

4. **Service Management with Istio:**
   - Istio is employed for secure inter-service communication, acting as an API gateway and supporting traffic management, load balancing, and fault tolerance.
   - Istio's distributed tracing and observability functions allow detailed system monitoring.

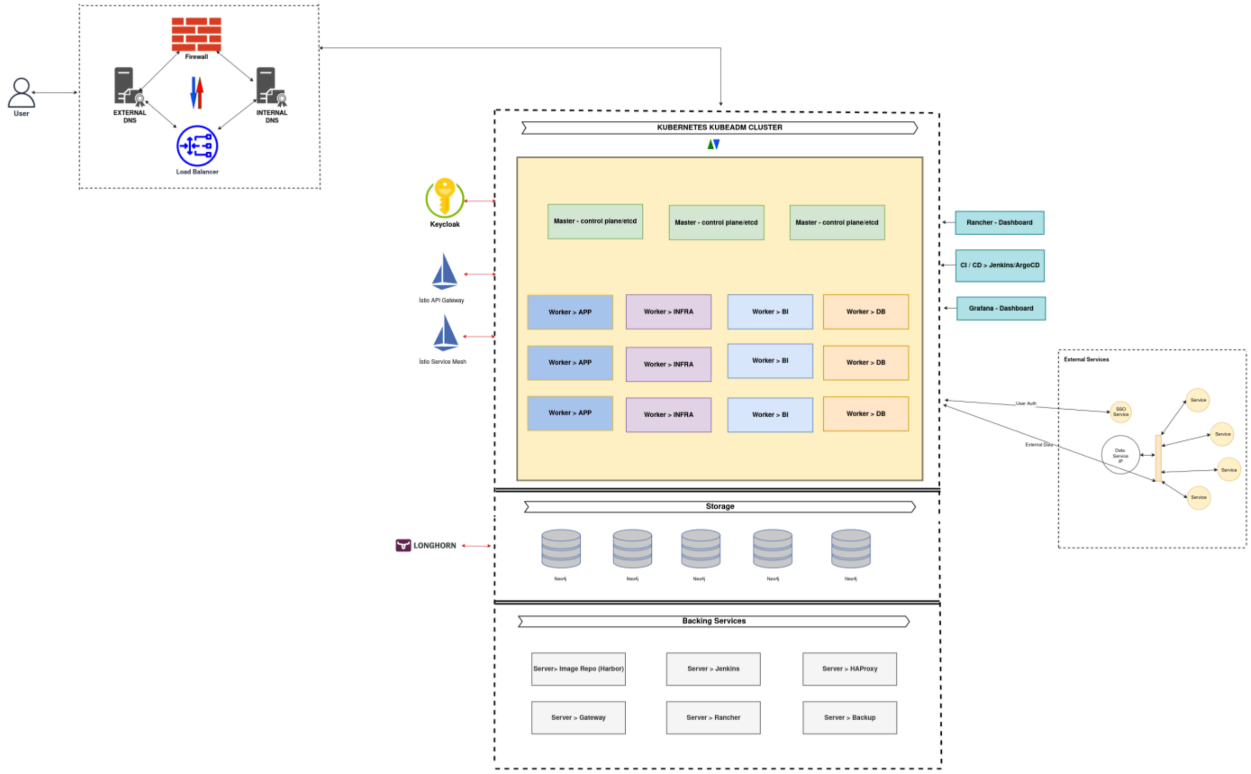

Image 1: Architectural Design of Windesk

Database and Storage Solutions

- **Neo4j Database:** Operating in a cluster setup, Neo4j provides high-performance data management using a graph-based model.
- **MinIO for Media Storage:** MinIO manages media file storage, including PDF, audio, and JPEG formats.
- **Centralized Logging with MongoDB:** Logs are centrally stored in MongoDB, which offers flexible schema support, enabling efficient log processing and storage.

Monitoring & Error Tracking

- **Error Tracking (Sentry):** Sentry facilitates immediate error tracking at the application level.
- **System Monitoring (Prometheus, Jaeger, Loki, Grafana):**
  - Prometheus collects time-series data.
  - Grafana visualizes this data to provide insights into system performance.
  - Jaeger enables distributed tracing, while Loki centralizes log management for efficient analysis.

Data Protection & Backup Policy

To prevent data loss and support swift recovery, a comprehensive backup strategy includes:

- **Backup Schedule:** Full weekly backups and incremental daily backups for Neo4j, Postgres, MongoDB, and MinIO, stored in a centralized location.
- **Verification:** Regular restoration tests are conducted to ensure data integrity and recovery readiness.

Caching and Messaging

- **Redis for Caching:** Redis, with its in-memory structure, ensures quick access to frequently used data, enhancing overall system performance.
- **Kafka for Asynchronous Messaging:** Kafka, a scalable and resilient messaging platform, supports efficient operations even under high workloads, essential for event-driven architecture.

CI/CD Processes

The development and deployment pipelines are automated through:

- **Jenkins for CI:** Continuously integrates code changes, automating tests and builds.
- **ArgoCD & Helm for CD:** ArgoCD automates Kubernetes deployments, while Helm Chart streamlines the setup, enabling swift and secure feature rollouts.

Technology Stack

Component	License	Purpose
Yellowfin 9.12.0	Licensed	Business intelligence and data visualization tool.
Neo4j: 4.4.33	Licensed	Graph database system.
Keycloak: 24.0.4	Open Source	Identity and access management.
MinIO: 2024.4.28	Open Source	Distributed object storage.
Jaeger: 1.47.0	Open Source	Distributed tracing and performance monitoring.
Redis: 7.2.4	Open Source	In-memory data structures and caching solution.
Kafka: 3.7.0	Open Source	Distributed data streaming platform.
Superset: 4.0.1	Open Source	Data visualization and business intelligence tool.
MongoDB: 7.0.9	Open Source	NoSQL database for log management.
PostgreSQL: 16.2.0	Open Source	Relational database for structured data management.

Table 1: Used Technologies in Windesk