9 Core Components of a Microservice App  
  
1. API Gateway  
Serves as the entry point for clients to interact with the system.  
Handles requests from external clients and routes them to the appropriate microservices.  
Provides functionalities like load balancing, rate limiting, authentication, and security.  
Communicates with both the Service Registry and Service Layer.  
  
2. Service Registry  
Maintains a dynamic list of all available microservices.  
Enables microservices to discover and communicate with each other dynamically.  
Facilitates scaling and resilience.  
Sends metrics toMetrics Visualization.  
  
3. Service Layer  
Comprises individual microservices (Service A, B, C, etc.).  
Encapsulates specific business functionalities.  
Each service operates independently, following the microservice architecture principles.  
Sends logs to Logstash and metrics to the Metrics component.  
  
4. Authorization Server  
Validates user credentials and permissions.  
Ensures secure access to the APIs and services.  
Interacts with the API Gateway for authentication and authorization processes.  
  
5. Distributed Layer (Replication)  
Provides data redundancy and consistency across microservices.  
Ensures data availability and fault tolerance.  
Supports the core infrastructure for distributed data.  
  
6. Distributed Cache  
Caches data close to the microservices to improve performance.  
Reduces latency and improves response time.  
Communicates with Load Balancer for efficient request distribution.  
  
7. Load Balancer  
Distributes incoming network traffic across multiple service instances.  
Ensures no single service instance becomes a bottleneck.  
Works closely with the Distributed Cache and Service Layer for efficient request handling.  
  
8. Prometheus (Metrics)  
Collects microservices' performance metrics.  
Monitors system health and operation metrics.  
Feeds data into Grafana for visualization.  
  
9. Grafana (Metrics Visualization)  
Provides dashboards for visual analytics of metrics collected by Prometheus.  
Helps in monitoring system performance, load, and other key metrics.  
  
Supporting Components:  
Logstash, Elasticsearch, Kibana (ELK Stack): For logging and log analysis.  
Logstash collects logs from microservices.  
Elasticsearch stores and indexes logs.  
Kibana visualizes logs for easy analysis.  
Distributed Messaging: Facilitates asynchronous communication between microservices.  
Queue Management Interface: Manages message queues for decoupled communication.  
  
  
  
Want to know more? Follow me or connect🥂  
  
Please don't forget to like❤️ and comment💭 and repost♻️  
  
[**x.com/sina\_riyahi**](http://x.com/sina_riyahi)  
[**medium.com/@Sina-Riyahi**](http://medium.com/@Sina-Riyahi)  
[**Instagram.com/Cna\_Riyahi**](http://instagram.com/Cna_Riyahi)  
[**github.com/sinariyahi**](http://github.com/sinariyahi)

Activate to view larger image,

