Apache Pulsar

Pulsar is a multi-tenant, high-performance solution for server-to-server messaging. Originally developed by Yahoo,and donated to Apache .

Its biggest advantage is that its architectural structure is designed in such a way that it can handle hundreds of billions of events daily.

As it is a streaming platform too. It provides high-quality streaming; also, it is a multi-tenant, high-performance solution. Pulsar is highly scalable and can manage the most demanding data movement out there.

Pulsar combines the best features of a traditional messaging system like RabbitMQ with those of a pub-sub system like Kafka.

Key Features

Let's start with a quick look at some of the key features:

- Inbuilt support for multiple clusters
- Support for Geo-replication of messages across multiple clusters
- Multiple subscription modes
- Scalable to millions of topics
- Uses Apache BookKeeper to guarantee message delivery.
- Low latency

Terminologies:

Apache ZooKeeper — Stores metadata information about Pulsar clusters.

Load Balancer: Pulsar has an inbuilt load balancer which distributes load across all brokers internally.

Global Replicators: Helps in replicating the data between n-brokers as configured for the namespace.

Global ZK: Global ZooKeeper helps in enablement of geo-replication.

Broker — Stateless component exposes REST and native endpoints to administer message transfer and storage.

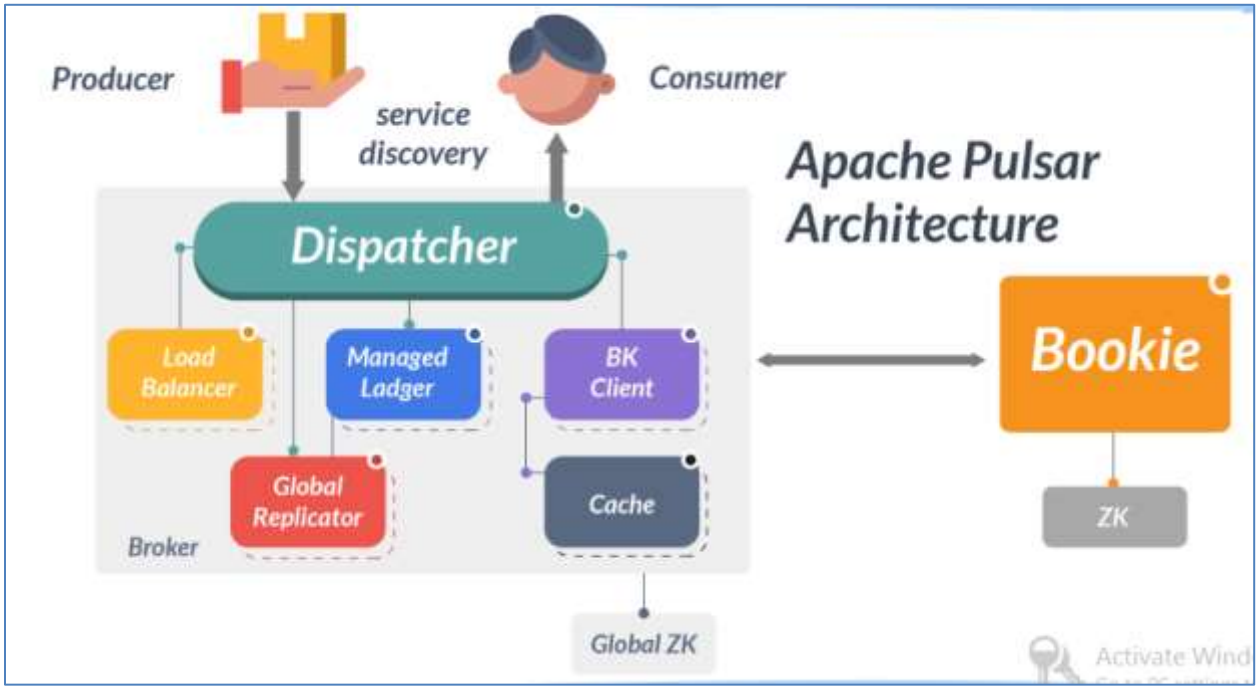
Bookie — Bookie is an instance of Apache BookKeeper that stores the messages. This is the persistent store for Pulsar clusters.

BookKeeper cluster consisting of one or more bookies to handles persistent storage of message

Metadata store: Pulsar uses apache zookeeper to store the metadata storage, cluster config and coordination.

Proxy : The proxy is needed to expose the brokers to the outside world when deployed in the cloud or Kubernetes. Proxy by itself can provide authentication and authorization, and seamlessly connect with the broker both on TLS or without TLS. Proxy has an inbuilt feature to pass authorization tokens to the broker for namespace permission validation.

Architecture:



Apache Pulsar takes a cloud-friendly approach by separating the serving and storage layers.

Pulsar has a layered architecture with data served by stateless “broker” nodes, while data storage is handled by “bookie” nodes.

This architecture provides the following benefits:

- Scales brokers independently.
- Scales bookies independently.
- Containerizes the ZooKeeper, broker and bookies.
- ZooKeeper provides the configuration and state of the cluster.

Built-in Authentication Plugins:

- mTLS authentication
- Athenz authentication
- Kerberos authentication
- JSON Web Token (JWT) authentication
- OAuth 2.0 authentication
- OpenID Connect
- HTTP basic authentication