

Compare NATS



- [Language and Platform Coverage](#)
- [Built-in Patterns](#)
- [Delivery Guarantees](#)
- [Multi-tenancy and Sharing](#)
- [AuthN](#)
- [AuthZ](#)
- [Message Retention and Persistence](#)
- [High Availability and Fault Tolerance](#)
- [Deployment](#)
- [Monitoring](#)
- [Management](#)
- [Integrations](#)
- [References](#)

This feature comparison is a summary of a few of the major components in several of the popular messaging technologies of today. This is by no means an exhaustive list and each technology should be investigated thoroughly to decide which will work best for your implementation.

In this comparison, we will be featuring NATS, Apache Kafka, RabbitMQ, Apache Pulsar, and gRPC.

Language and Platform Coverage

Project	Client Languages and Platforms
NATS	Core NATS: 48 known client types, 11 supported by maintainers, 18 contributed by the community. NATS Streaming: 7 client types supported by maintainers, 4 contributed by the community. NATS servers can be compiled on architectures supported by Golang. NATS provides binary distributions.
gRPC	13 client languages.

Project	Client Languages and Platforms
Kafka	18 client types supported across the community and by Confluent. Kafka servers can run on platforms supporting java; very wide support.
Pulsar	7 client languages, 5 third-party clients - tested on macOS and Linux.
Rabbit	At least 10 client platforms that are maintainer-supported with over 50 community supported client types. Servers are supported on the following platforms: Linux, Windows NT.

Built-in Patterns

Project	Supported Patterns
NATS	Streams and Services through built-in publish/subscribe, request-reply, and load-balanced queue subscriber patterns. Dynamic request permissioning and request subject obfuscation is supported.
gRPC	One service, which may have streaming semantics, per channel. Load Balancing for a service can be done either client-side or by using a proxy.
Kafka	Streams through publish/subscribe. Load balancing can be achieved with consumer groups. Application code must correlate requests with replies over multiple topics for a service (request-reply) pattern.
Pulsar	Streams through publish/subscribe. Multiple competing consumer patterns support load balancing. Application code must correlate requests with replies over multiple topics for a service (request-reply) pattern.
Rabbit	Streams through publish/subscribe, and services with a direct reply-to feature. Load balancing can be achieved with a Work Queue. Applications must correlate requests with replies over multiple topics for a service (request-reply) pattern.

Delivery Guarantees

Project	Quality of Service / Guarantees
NATS	At most once, at least once, and exactly once is available in JetStream.

Project	Quality of Service / Guarantees
gRPC	At most once.
Kafka	At most once, at least once, and exactly once.
Pulsar	At most once, at least once, and exactly once.
Rabbit	At most once, at least once.

Multi-tenancy and Sharing

Project	Multi-tenancy Support
NATS	NATS supports true multi-tenancy and decentralized security through accounts and defining shared streams and services.
gRPC	N/A
Kafka	Multi-tenancy is not supported.
Pulsar	Multi-tenancy is implemented through tenants; built-in data sharing across tenants is not supported. Each tenant can have its own authentication and authorization scheme.
Rabbit	Multi-tenancy is supported with vhosts; data sharing is not supported.

AuthN

Project	Authentication
NATS	NATS supports TLS, NATS credentials, NKEYS (NATS ED25519 keys), username and password, or simple token.
gRPC	TLS, ALT, Token, channel and call credentials, and a plug-in mechanism.
Kafka	Supports Kerberos and TLS. Supports JAAS and an out-of-box authorizer implementation that uses ZooKeeper to store connection and subject.
Pulsar	TLS Authentication, Athenz, Kerberos, JSON Web Token Authentication.
Rabbit	TLS, SASL, username and password, and pluggable authorization.

AuthZ

Project	Authorization
NATS	Account limits including number of connections, message size, number of imports and exports. User-level publish and subscribe permissions, connection restrictions, CIDR address restrictions, and time of day restrictions.
gRPC	Users can configure call credentials to authorize fine-grained individual calls on a service.
Kafka	Supports JAAS, ACLs for a rich set of Kafka resources including topics, clusters, groups, and others.
Pulsar	Permissions may be granted to specific roles for lists of operations such as produce and consume.
Rabbit	ACLs dictate permissions for configure, write, and read operations on resources like exchanges, queues, transactions, and others. Authentication is pluggable.

Message Retention and Persistence

Project	Message Retention and Persistence Support
NATS	Supports memory and file persistence. Messages can be replayed by time, count, or sequence number, and durable subscriptions are supported. With NATS streaming, scripts can archive old log segments to cold storage.
gRPC	N/A
Kafka	Supports file-based persistence. Messages can be replayed by specifying an offset, and durable subscriptions are supported. Log compaction is supported as well as KSQL.
Pulsar	Supports tiered storage including file, Amazon S3 or Google Cloud Storage (GCS). Pulsar can replay messages from a specific position and supports durable subscriptions. Pulsar SQL and topic compaction is supported, as well as Pulsar functions.
Rabbit	Supports file-based persistence. Rabbit supported queue-based semantics (vs log), so no message replay is available.

High Availability and Fault Tolerance

Project	HA and FT Support
NATS	Core NATS supports full mesh clustering with self-healing features to provide high availability to clients. NATS streaming has warm failover backup servers with two modes (FT and full clustering). JetStream supports horizontal scalability with built-in mirroring.
gRPC	N/A. gRPC relies on external resources for HA/FT.
Kafka	Fully replicated cluster members are coordinated via Zookeeper.
Pulsar	Pulsar supports clustered brokers with geo-replication.
Rabbit	Clustering Support with full data replication via federation plugins. Clusters require low-latency networks where network partitions are rare.

Deployment

Project	Supported Deployment Models
NATS	The NATS network element (server) is a small static binary that can be deployed anywhere from large instances in the cloud to resource constrained devices like a Raspberry PI. NATS supports the Adaptive Edge architecture which allows for large, flexible deployments. Single servers, leaf nodes, clusters, and superclusters (cluster of clusters) can be combined in any fashion for an extremely flexible deployment amenable to cloud, on-premise, edge and IoT. Clients are unaware of topology and can connect to any NATS server in a deployment.
gRPC	gRPC is point to point and does not have a server or broker to deploy or manage, but always requires additional pieces for production deployments.
Kafka	Kafka supports clustering with mirroring to loosely coupled remote clusters. Clients are tied to partitions defined within clusters. Kafka servers require a JVM, eight cores, 64 GB to 128 GB of RAM, two or more 8-TB SAS/SSD disks, and a 10-Gig NIC. (1)

Project	Supported Deployment Models
Pulsar	Pulsar supports clustering and built-in geo-replication between clusters. Clients may connect to any cluster with an appropriately configured tenant and namespace. Pulsar requires a JVM and requires at least 6 Linux machines or VMs. 3 running ZooKeeper. 3 running a Pulsar broker and a BookKeeper bookie. (2)
Rabbit	Rabbit supports clusters and cross-cluster message propagation through a federation plugin. Clients are unaware of topology and may connect to any cluster. The server requires the Erlang VM and dependencies.

Monitoring

Project	Monitoring Tooling
NATS	NATS supports exporting monitoring data to Prometheus and has Grafana dashboards to monitor and configure alerts. There are also development monitoring tools such as nats-top. Robust side car deployment or a simple connect-and-view model with NATS surveyor is supported.
gRPC	External components such as a service mesh are required to monitor gRPC.
Kafka	Kafka has a number of management tools and consoles including Confluent Control Center, Kafka, Kafka Web Console, Kafka Offset Monitor.
Pulsar	CLI tools, per-topic dashboards, and third-party tools.
Rabbit	CLI tools, a plugin-based management system with dashboards and third-party tools.

Management

Project	Management Tooling
NATS	NATS separates operations from security. User and Account management in a deployment may be decentralized and managed through a CLI. Server (network element) configuration is separated from security with a command line and configuration file which can be reloaded with changes at runtime.
gRPC	External components such as a service mesh are required to manage gRPC.

Project	Management Tooling
Kafka	Kafka has a number of management tools and consoles including Confluent Control Center, Kafka, Kafka Web Console, Kafka Offset Monitor.
Pulsar	CLI tools, per-topic dashboards, and third-party tools.
Rabbit	CLI tools, a plugin-based management system with dashboards and third-party tools.

Integrations

Project	Built-in and Third Party Integrations
NATS	NATS supports WebSockets, a Kafka bridge, an IBM MQ Bridge, a Redis Connector, Apache Spark, Apache Flink, CoreOS, Elastic, Elasticsearch, Prometheus, Telegraf, Logrus, Fluent Bit, Fluentd, OpenFAAS, HTTP, and MQTT, and more .
gRPC	There are a number of third party integrations including HTTP, JSON, Prometheus, Grift and others. (3)
Kafka	Kafka has a large number of integrations in its ecosystem, including stream processing (Storm, Samza, Flink), Hadoop, database (JDBC, Oracle Golden Gate), Search and Query (ElasticSearch, Hive), and a variety of logging and other integrations.
Pulsar	Pulsar has many integrations, including ActiveMQ, Cassandra, Debezium, Flume, Elasticsearch, Kafka, Redis, and others.
Rabbit	RabbitMQ has many plugins, including protocols (MQTT, STOMP), WebSockets, and various authorization and authentication plugins.

References

- https://docs.cloudera.com/HDPDocuments/HDF3/HDF-3.1.0/bk_planning-your-deployment/content/ch_hardware-sizing.html
- <https://pulsar.apache.org/docs/4.0.x/deploy-bare-metal/>
- <https://github.com/grpc-ecosystem>