Conduktor Governance

From empowering your application developers to self-serve on Kafka with enforced best practices, to achieving multi-tenancy through cluster virtualization.

Self-serve that works!

Remove the Platform Ops bottleneck by giving teams the power to create topics, on their own virtual clusters, within the guardrails you deem acceptable.

Identify who produced that message

When trouble appears quickly identify where it came from. Gateway allows you to flexibly tag messages as they enter your Kafka architecture, meaning you can quickly identify where the issue is in a hurry.

Kafka your way

Prevent messages being sent that aren't acceptable on production, that have ignored business rules or are lacking a schema. Few are masters of the thousands of Kafka configurations; implement organization-wide controls to prevent bad configurations.

USED BY: **DevOps** **Infrastructure Architects** **Systems Architects**

Create topic policy

Enforce naming conventions and characteristics of topics that could be tweaked by projects. Allow teams to create topics only if they meet the requirements. Set the maximum, minimum, what to do if fails and an override value for topic creation.

- Number of partitions
- Replication factor
- Applied by regex conditions

Multi-tenancy

Create virtual clusters for isolation and namespacing, per team, project or however you wish. Have visibility on all your Kafka activities in your enterprise in case of problems and for regulations purposes.

- Virtual Clusters
- Cluster Federation

Enforce Kafka best-practices

Provide organization-wide controls that prevent erroneous configurations that could lead to production incidents.

- Technical Rule Enforcement
- Business Policies
- Alter Topic Policy
- Dynamic Header Injection
- Produce Policy
- Producer rate limiting policy
- Topic required schema ID Policy

Try Conduktor For Free

conduktor.io/get-started

Conduktor Governance

Deployments

Scalable: A Docker image that can be deployed like any other within your containerisation setup, such as on Kubernetes or managed services.

Resilient: Like Kafka if a broker dies it can be restarted whilst Gateway keeps running. As Kafka protocol magic is in play your applications remain available.

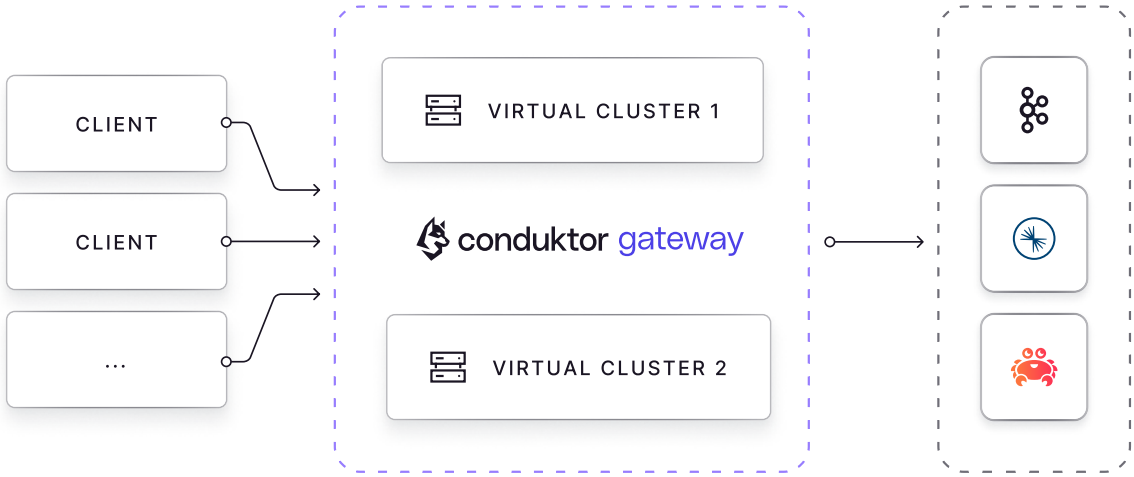
Latency: By default GW is unnoticeable, it will add latency in the order of milliseconds. If you start to add heavier functionality, like encryption, there is naturally a small price to pay.

Docker

Kubernetes

Air-Gapped Deployments

Virtual cluster overview



Try Conduktor For Free

conduktor.io/get-started