

Building a Kafka data platform?

Getting your organization streaming data-ready is no walk in the park.

Maybe you're looking to modernize your applications for real-time streaming by using a powerful open-source technology (like Kafka).

Perhaps your data platform project is already locked and loaded.

But although your streaming project might start smoothly enough, providing self-service for real-time data can quickly grow into a monster as you scale across technologies, teams and territories.

We've made a basic checklist for your platform team to help you consider the right things and ask the right questions when it comes to designing a real-time data platform.

These considerations have been gathered from the open-source community – well-seasoned developers, platform and data engineers keen to share their learnings.

Productive **people** & processes

Self-service access to **data**

Standardized **real-time apps**



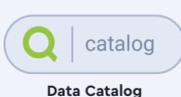
Key challenges for Platform Engineers:

Your data platform should address these challenges.

- ▶ How can I provide self-service access for my developers?
- ▶ How can we future-proof when technologies change so quickly?
- ▶ How can I spend less time operating the data platform?
- ▶ How can I avoid having to use so many siloed tools?
- ▶ How can I meet my corporate governance requirements?
- ▶ How can we control and minimize costs?
- ▶ How can I standardize our application deployment?

READY, STEADY, STREAM?

Components of a real-time data platform for platform engineers:



catalog

Data Catalog

Can users explore metadata across my platform & apps as easily as Google Search, including Kafka & Elasticsearch?



Monitor & Report

Do I have complete visibility on the health of the Kafka platform, and can I share this with my developers & internal customers?



Alert Rules & Notifications

Am I and my internal clients notified when things go wrong via our favorite channels such as Slack?



High Availability

Am I confident that I'm providing high service levels and I'm able to react when disaster strikes?



Team Collaboration

Can techies or Kafka n00bs access our data platform and query data without submitting a ticket?



Fine Grained Permissions

Are we able to authorize data access across different teams and segregate duties?



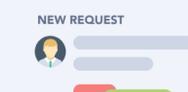
Connection Management

Can I explore data across many different data stores that make up my data platform such as Elasticsearch, Postgres?



Integrations

Can I plug my data platform with all my different corporate services such as ServiceNow, PagerDuty, Splunk & Elasticsearch?



Change Management

Can I apply approval workflows for my devs and internal clients such as creating a new topic?



Audit Logs

Can we meet corporate compliance requirements by ensuring every action from our developers is audited and sent to the security team?



Resource Management

Do I know how many infrastructure resources my internal clients are consuming, and might be impacting other users?



Support and SLAs

Can I monitor and maintain service levels and provide visibility to my internal clients?



Regulations and Compliance

Am I sure I am meeting my compliance & regulatory standards with security controls, auditing and processes?



Identity Providers

Can I authenticate users of my data platform with my corporate identity provider such as Okta?



Manage Deployments

Can I efficiently deploy real-time applications on my data platform and monitor their health?



Cost Management

Can I optimize the costs of running my data platform and am I able to accurately cross-charge my internal clients?



Secret Management

Can I connect all my passwords and secrets with external keystores to ensure we never expose credentials?

See a few of these left unchecked?
Have a look around our DataOps workspace to see how your data platform could take shape.

Try now