CONFLUENT

Considerations for Getting Started with Confluent

Checklist

Preface

How to use this guide: This checklist is intended to be a 'print out & use' utility which allows you to plan for your upcoming Confluent project by asking & answering the right questions from the start. As you review each item, frame it with the question 'Have we thought about ...'

This is not meant to be an exhaustive list. We understand every customer's journey is different so we encourage you to take this framework & amend it according to your unique requirements.

As you start to plan your engagement, there are plenty of useful resources to guide you. Here are a few of our recommendations:

- The Confluent Data In Motion Blog
- Support Guides:
  - Getting Started
  - Welcome to Support
  - Confluent Cloud Onboarding Checklist

Note: All bookmarks in this guide are available for you to download as an html file & import into your favorite browser from this link.

Section 1: Enablement

- ☐ Have we set up a corporate / team repository for knowledge sharing (best practices, whitepapers, contacts, roles & responsibilities etc)?
- ☐ Have Confluent's top recommended resources (see Top Resources section of Bookmarks) been shared with all team members?
- ☐ Have we downloaded & reviewed the Guide to Data In Motion to plan our journey & identify knowledge gaps?
- ☐ Have any gaps in enablement been identified & do we have a plan to resolve those gaps?
- ☐ Do we know how to add the necessary individuals to the support portal & when to use it?
- ☐ Have all the necessary teams / individuals received sufficient Confluent enablement so that:
  - ☐ Each understands the Confluent Cloud concepts (e.g. Cluster Types, Resource Limits, Access Management, Networking options etc)?
  - ☐ Each is aware of the various Confluent Components (e.g. ksqlDB, Connect, Schema Registry) & the problems they solve?
    - ☐ If Kafka Connect is to be used, do we understand the important concepts (e.g. Monitoring, Troubleshooting, Error handling, DLQs, SMTs)?
    - ☐ If ksqlDB is to be used, do we understand ksqlDB's architecture & how ksqlDB can be used to achieve use cases like Materialized Caches, Streaming ETL Pipelines & Event Driven Microservices?
    - ☐ If schema registry is to be used, do we understand concepts such as schema compatibility settings, schema validation, supported formats (e.g. JSON, PROTOBUF)?
  - ☐ Each individual has enough knowledge to identify which components are to be fully managed & those which will be self-managed?
  - ☐ Each understands how to get assistance they need through the support portal?
- ☐ There is sufficient knowledge of security concepts? For example:
  - ☐ Do we need to set up SSO & do we know how?
  - ☐ Are we aware of the difference between Cloud UI user accounts vs. service accounts?
  - ☐ Have we had sufficient enablement to setup authorization with ACLs or RBAC?
  - ☐ Do we understand the various networking options Confluent Cloud provides & feel confident in selecting the right options for our use case & cloud provider?
- ☐ Have our developers been educated on how to work with events? More specifically:
  - ☐ Implications & best practices for setting client parameters in producer/consumer code with Confluent Cloud?
  - ☐ Concepts such as at least once, at most once, exactly once? What is needed & how to code for it.
  - ☐ How to write code to meet business requirements for throughput vs. latency & durability vs. availability?
  - ☐ How to decouple Producer vs. Consumer with schema registry & protect against bad messages?

Section 2: Roles & Responsibilities

- ☐ Have individuals and/or teams been identified for the following tasks?
    - ☐ Operations (e.g. Monitoring with Metrics API or Integrate with 3rd Party Solutions, Auditing)
    - ☐ Automation (E.g. Terraform or Pulumi providers)
    - ☐ Infrastructure management (e.g. Upgrades)
    - ☐ Change Control
    - ☐ Other
- ☐ Have the right people been identified for defining business requirements & SLAs (uptime, latency, retention)?
- ☐ Have the right people been identified to provide requirements for:
    - ☐ Encryption
    - ☐ E2E Encryption: Message Payloads, masking/tokenizing fields etc
    - ☐ Secrets
    - ☐ Authorization (ACLs vs. RBAC)
    - ☐ Authentication
    - ☐ Auditing

Section 3: Infrastructure

- ☐ Have all external systems which will need to connect to Kakfa (upstream & downstream) been identified?
    - ☐ Do we know which external systems are available as source/sink connectors via Kafka Connect & how to check if they are available self vs. fully managed?
- ☐ Have network boundaries / security policies been reviewed & the appropriate cloud solution identified?
- ☐ Have requirements been captured & shared for the following?
    - ☐ Encryption
    - ☐ E2E Encryption: Message Payloads, masking/tokenizing fields etc
    - ☐ Secrets
    - ☐ Authorization (ACLs vs. RBAC)
    - ☐ Authentication
    - ☐ Auditing
- ☐ Is automation of cluster management required?
    - ☐ Are we aware of the supported tools / mechanisms within Confluent Cloud?
        - ☐ Terraform or Pulumi providers
        - ☐ Confluent Cloud CLI
        - ☐ AdminClient API
- ☐ Will downtime be permitted for maintenance (upgrades etc) & have clients been developed with these requirements in mind?
- ☐ It is strongly advised that we enforce a strategy for topic naming; Do we feel enabled to take the right approach for this?
- ☐ Do we need to enforce a standard for provisioning of Confluent Components?
    - ☐ e.g. do we need to capture process/procedures for provisioning/deprovisioning components?
    - ☐ How will we protect against over-provisioning (where tenants create components with too many partitions/replicas) which can impact cost/performance?
    - ☐ How do we protect against 'noisy neighbors' where high throughput applications can impact other tenants?
- ☐ Stream Lineage: Are we aware of how to check what components have been created within Confluent Cloud, where they sit within our pipeline, and what the flow of data looks like?

Section 4: Operations

- ☐ SLAs:
    - ☐ What are requirements for uptime, latency & throughput?
    - ☐ What are the requirements for data retention?
    - ☐ Do we know how to translate our SLAs into monitoring alerts?
    - ☐ Are we able to let the right people know when workloads are too high or too low (for example, can we differentiate between a “slow day” and a problem upstream)?
- ☐ Monitoring:
    - ☐ Have we set the appropriate monitoring thresholds to alert on?
    - ☐ What tool will be used for alerting/monitoring from those available?
- ☐ Support:
    - ☐ Have the right people been added to the support portal?
    - ☐ Do you understand the various Cloud Support Plans & has it been set via the Confluent Cloud UI?
    - ☐ Do you know the SLAs for the various support levels & how to escalate?
- ☐ Process & Procedures:
    - ☐ Are run-books needed & who will produce them?

Section 5: Define & test your use-cases

- ☐ Security:
    - ☐ Encryption
    - ☐ Authorization
    - ☐ Authentication
    - ☐ Secrets
- ☐ Functionality:
    - ☐ Streaming ETL:
      Demo: Develop a Streaming ETL pipeline
      WhitePaper: Top 6 Reasons to Modernize Legacy Messaging Infrastructure
      Blog: Streaming ETL SFDC Data for Real-Time Customer Analytics
    - ☐ Database Modernisation:
      Demo: Modernize your Database with Confluent and MongoDB
      Blog: Accelerate Cloud Database Modernizations and Migrations with Confluent
    - ☐ Mainframe Modernization:
      Blog: Enterprise Mainframe CDC to Apache Kafka with tcVISION and Confluent
      Podcast: Building a Microservices Architecture with Apache Kafka
      Online Talk: Mainframe Integration, Offloading and Replacement with Apache Kafka
      Webinar: Bringing the Mainframe to the Data Lake
    - ☐ IOT:
      Webinar: 0-60: Tesla's Streaming Data Platform
      Blog: Stream Processing with IoT Data: Challenges, Best Practices, and Techniques
      Blog: Apache Kafka Native MQTT at Scale with Confluent Cloud and Waterstream
      Podcast: Resilient Edge Infrastructure for IoT Using Apache Kafka ft. Kai Waehner
- ☐ Performance:
    - ☐ Throughput / Latency
    - ☐ Durability / Availability
    - ☐ Scaling
- ☐ SLAs:
    - ☐ Retention
    - ☐ Recoverability
    - ☐ RPO/RTO