Kafka: The Definitive Guide (2nd ed.)

Study Plan Rendered as User Stories

This standalone document contains: (1) a compact guide for writing good user stories, (2) a reusable Story Card template, and (3) filled cards for each chapter.

How to Write Effective User Stories (Quick Guide)

- **Format:** *As a <persona>, I want <capability> so that <benefit>.*
- **Value:** Connect capability to measurable business value (throughput, MTTR, error rate, time-to-market).
- **Acceptance Criteria (BDD): Given/When/Then** statements that are objective, testable, and observable.
- **Non-Functional:** Call out performance, security, reliability, privacy, accessibility, and internationalization.
- **Sizing:** Small enough to complete in a sprint; use story points only for planning—not contracts.
- **Evidence:** Prefer runnable demos, dashboards, or logs over prose.

Story Card Template

TEMPLATE — Title of the Story

Epic / Feature	*Where does this story live? e.g., "Foundations" or "Security"*
Business Value	*Why this matters; how we will measure success*
Priority / Estimate	*Priority: Must/Should/Could; SP: N*
Persona	*Primary actor, e.g., "Kafka developer"*
Dependencies	*Pre-req tools, clusters, data sets*
Assumptions / Risks	*Important constraints and risks*

Story *As a <persona>, I want <capability> so that <benefit>.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

- **Scenario** Happy path
- **Given** preconditions are met (cluster available, topics created, access granted)
- **When** the *Hands-on Objectives* are completed
- **Then** the stated *Outcomes/Deliverables* are observable and recorded

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. **Definition of Done**: All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

- ☐ Concrete task 1 (15–60 minutes).
- ☐ Concrete task 2 (runnable, observable output).
- ☐ Concrete task 3 (test or measurement).
- ☐ Publish evidence (README, screenshot, metrics).

Chapter Cards (Kafka 2nd Edition)

KAF-01 — Meet Kafka

Epic / Feature	Foundations
Business Value	Shared understanding of Kafka primitives to reduce integration risk and align on use cases.
Priority / Estimate	Priority: Must SP: 3
Persona	Backend engineer new to Kafka
Dependencies	Local dev toolchain; diagramming tool
Assumptions / Risks	Misapplied patterns if primitives are misunderstood

Story *As a backend engineer, I want to understand Kafka's core abstractions so that I can identify fit-for-purpose use cases and avoid misuse.*

Non-Functional Performance Reliability Security

Acceptance Criteria (BDD)

- **Given** a study repo and note template
- **When** I complete the hands-on objectives
- **Then** a one-page "Why Kafka" brief and event-flow diagram are committed

Tasks

- ☐ List three candidate use cases and map producers, topics/partitions, consumers.
- ☐ Draw an event flow (sequence from producer to consumer) and annotate retention/ordering.
- ☐ Compare Kafka vs. current integration (2–3 bullet pros/cons each).

KAF-02 — Installing Kafka

Epic / Feature	Foundations
Business Value	Repeatable environment for iterative learning and testing without blocking others.
Priority / Estimate	Priority: Must SP: 5
Persona	DevOps-minded developer
Dependencies	Docker Desktop/Podman; Compose; make
Assumptions / Risks	Local ports may be occupied; resource limits on laptops

Story *As a developer, I want a reproducible multi-broker dev cluster so that I can practice operations safely.*

Non-Functional Reliability Security

Acceptance Criteria (BDD)

- **Given** Docker is installed
- **When** I run `make up`
- **Then** a 3-broker cluster with a 6-partition, RF=3 topic is available

Tasks

- ☐ Author `docker-compose.yml` (3 brokers + controller/KRaft or ZooKeeper as applicable).
- ☐ Create topic `events` with partitions=6, RF=3; verify ISR.
- ☐ Kill one broker and show continued produce/consume; capture logs.

KAF-03 — Producers

Epic / Feature	Client Development
Business Value	Safe, efficient ingestion with predictable latency/throughput.
Priority / Estimate	Priority: Must SP: 5
Persona	Application developer
Dependencies	Ch02 cluster; Schema tooling (Avro/JSON)
Assumptions / Risks	Hot partitions if keys poorly chosen

Story *As an app dev, I want to produce messages with proper batching, compression, and keys so that ingestion is fast and ordered where needed.*

Non-Functional Performance Reliability

Acceptance Criteria (BDD)

- **Given** a producer sample
- **When** I toggle `acks`, `linger.ms`, compression
- **Then** I record latency/throughput and ordering effects

Tasks

- ☐ Implement sync and async producer with callbacks.
- ☐ Send JSON and Avro; document serializer choice.
- ☐ Benchmark 3 configurations; paste metrics in README.

KAF-04 — Consumers

Epic / Feature	Client Development
Business Value	Robust consumption with minimal lag and controlled rebalancing.
Priority / Estimate	Priority: Must SP: 5
Persona	Application developer
Dependencies	Ch02 cluster; metrics tooling
Assumptions / Risks	Long processing may trigger rebalances

Story *As an app dev, I want consumers that handle backpressure and commits safely so that processing is reliable and observable.*

Non-Functional Reliability Performance

Acceptance Criteria (BDD)

- **Given** two consumer instances in one group
- **When** I scale up/down
- **Then** rebalances are logged and lag remains bounded

Tasks

☐ Implement manual vs. auto commit; add rebalance listener.

☐ Provide a “process exactly this offset” backfill tool.

☐ Capture consumer lag charts before/after scaling.

KAF-05 — Administering Kafka Programmatically (AdminClient)

Epic / Feature	Tooling
Business Value	Safer, automated operations; less manual error.
Priority / Estimate	Priority: Should SP: 3
Persona	Platform engineer
Dependencies	Client libs; cluster admin access
Assumptions / Risks	Changing topic configs in prod without guardrails

Story *As a platform engineer, I want an AdminClient CLI so that I can manage topics/-configs/groups via code and CI.*

Non-Functional Reliability Security

Acceptance Criteria (BDD)

- **Given** a dev cluster
- **When** I run the tool
- **Then** topics can be created/altered and configs listed with dry-run support

Tasks

☐ Implement commands: create/alter topic, list groups, delete records (guarded).

☐ Add “add-partitions” with safety checks.

☐ Provide example CI job for read-only inventory.

KAF-06 — Kafka Internals

Epic / Feature	Foundations
Business Value	Better troubleshooting and right-sized configs.
Priority / Estimate	Priority: Should SP: 3
Persona	Platform engineer
Dependencies	Log inspection tools
Assumptions / Risks	Misinterpreting compaction/segment behavior

Story *As a platform engineer, I want to understand replication, logs, and compaction so that I can tune and debug effectively.*

Non-Functional Performance Reliability

Acceptance Criteria (BDD)

- **Given** a compacted topic
- **When** I write updates and tombstones
- **Then** I can show post-compaction state and segment layout

Tasks

- ☐ Create a compacted topic; write keys/updates/tombstones.
- ☐ Inspect log segments and indexes; summarize findings.
- ☐ Map common symptoms to likely internal causes (cheat sheet).

KAF-07 — Reliable Data Delivery

Epic / Feature	Resilience
Business Value	Minimize message loss/duplication; predictable recovery.
Priority / Estimate	Priority: Must SP: 5
Persona	SRE / platform engineer
Dependencies	Chaos testing harness
Assumptions / Risks	Network partitions and broker failures

Story *As an SRE, I want validated at-least-once delivery so that failures don't cause data loss.*

Non-Functional Reliability Performance

Acceptance Criteria (BDD)

- **Given** replication factor 3 and minISR 2
- **When** a broker is killed and network throttled
- **Then** producers/consumers continue and no committed data is lost

Tasks

- ☐ Configure RF=3 and `min.insync.replicas=2`; enable idempotent producer.
- ☐ Run failure drills (kill broker, throttle network); capture outcomes.
- ☐ Document retry/backoff and dead-letter strategies.

KAF-08 — Exactly-Once Semantics (EOS)

Epic / Feature	Resilience
Business Value	Prevent double-counting in financial/critical pipelines.
Priority / Estimate	Priority: Should SP: 5
Persona	Application developer
Dependencies	Transactions enabled; two topics
Assumptions / Risks	Throughput overhead with EOS

Story *As an app dev, I want a read-process-write topology with transactions so that results are exactly-once.*

Non-Functional Reliability

Acceptance Criteria (BDD)

- **Given** transactional producer and consumer
- **When** I restart during processing
- **Then** no double-counts are observed in sinks

Tasks

☐ Implement transactional producer; handle fencing and timeouts.

☐ Compare EOS vs. at-least-once throughput.

☐ Write a replay test to prove invariants.

KAF-09 — Kafka Connect (Pipelines)

Epic / Feature	Integration
Business Value	Faster integration with external systems with less bespoke code.
Priority / Estimate	Priority: Must SP: 5
Persona	Data engineer
Dependencies	Connect runtime; connector plugins
Assumptions / Risks	Bad schemas/SMTs can break downstream

Story *As a data engineer, I want to move data between Kafka and external systems using Connect so that pipelines are standardized.*

Non-Functional Reliability Security

Acceptance Criteria (BDD)

- **Given** a source and sink connector
- **When** I apply SMTs and run
- **Then** data flows; errors go to DLQ

Tasks

☐ File/DB source → Kafka → Postgres/Elasticsearch sink.

☐ Add SMT for enrichment; configure DLQ for error records.

☐ Document schema evolution strategy.

KAF-10 — Cross-Cluster Mirroring

Epic / Feature	Disaster Recovery
Business Value	Regional resilience and data locality.
Priority / Estimate	Priority: Should SP: 5
Persona	Platform engineer
Dependencies	Two clusters; connectivity
Assumptions / Risks	Offset/ACL sync; filtering topics

Story *As a platform engineer, I want to mirror critical topics between clusters so that failover is possible.*

Non-Functional Reliability Security

Acceptance Criteria (BDD)

- **Given** two clusters
- **When** I mirror topics and simulate link degradation
- **Then** consumers can fail over with minimal disruption

Tasks

☐ Configure MirrorMaker (or equivalent) for a subset of topics.

☐ Perform planned failover test; record recovery time.

☐ Write a mirroring playbook (include ACLs and offset sync notes).

KAF-11 — Securing Kafka

Epic / Feature	Security
Business Value	Protect data-in-motion and enforce least privilege.
Priority / Estimate	Priority: Must SP: 5
Persona	Security engineer
Dependencies	PKI/certs; secret storage
Assumptions / Risks	Misconfigured TLS can cause outages

Story *As a security engineer, I want TLS, SASL, and ACLs enforced so that only authorized clients can access topics.*

Non-Functional Security Reliability

Acceptance Criteria (BDD)

- **Given** issued certs and user principals
- **When** I enable mTLS and ACLs
- **Then** unauthorized access fails; authorized paths succeed

Tasks

☐ Enable TLS (brokers and clients); rotate one cert in dev.

☐ Configure SASL (e.g., SCRAM/OAuth) and least-privilege ACLs.

☐ Add a security hardening checklist to the repo.

KAF-12 — Administering Kafka (Day-2 Ops)

Epic / Feature	Operations
Business Value	Safe changes and efficient maintenance.
Priority / Estimate	Priority: Should SP: 5
Persona	SRE / platform engineer
Dependencies	CLI tools; disk/partition metrics
Assumptions / Risks	Risky operations without dry-run/verification

Story *As an SRE, I want standard operating procedures for common Kafka admin tasks so that changes are safe and auditable.*

Non-Functional Reliability

Acceptance Criteria (BDD)

- **Given** topic growth and uneven disk usage
- **When** I add partitions and move replicas
- **Then** data is rebalanced and verification passes

Tasks

- ☐ Run preferred leader election; perform partition reassignments.
- ☐ Execute replica verification tool and record results.
- ☐ Document safe-delete procedures and rollback plans.

KAF-13 — Monitoring Kafka

Epic / Feature	Observability
Business Value	Faster detection and response; fewer incidents.
Priority / Estimate	Priority: Must SP: 5
Persona	SRE
Dependencies	JMX exporter; Prometheus; Grafana
Assumptions / Risks	Alert fatigue if thresholds are noisy

Story *As an SRE, I want SLIs/SLOs and dashboards for brokers, clients, and topics so that health is visible.*

Non-Functional Reliability Performance

Acceptance Criteria (BDD)

- **Given** metrics export
- **When** I deploy dashboards and alerts
- **Then** lag/regressions trigger actionable alerts; black-box probe is green

Tasks

☐ Export broker/consumer metrics; build Grafana dashboards.

☐ Implement consumer-lag alerts and a heartbeat topic.

☐ Define SLOs (e.g., produce p95, consume p95, offline partitions=0).

KAF-14 — Stream Processing (Kafka Streams)

Epic / Feature	Stream Processing
Business Value	Real-time insights with stateful processing.
Priority / Estimate	Priority: Should SP: 8
Persona	Application developer / data engineer
Dependencies	Two topics; Kafka Streams lib; test data
Assumptions / Risks	State store size and rebalancing concerns

Story *As a developer, I want a Kafka Streams topology with joins and windowed aggregations so that I can deliver real-time features.*

Non-Functional Performance Reliability

Acceptance Criteria (BDD)

- **Given** input topics A and B
- **When** I run a join + tumbling window aggregate
- **Then** results are correct under reprocessing and restart

Tasks

- ☐ Implement topology with Interactive Queries and topology tests.
- ☐ Add a replay/reprocessing script; validate determinism.
- ☐ Document scaling and fault recovery behavior.

Capstone (Optional)

Build an end-to-end pipeline: CDC → Kafka → Streams aggregation → Postgres/Elasticsearch sink; dashboards and SLOs; chaos test; optional cross-cluster failover.