

Strimzi Glossary

Table of Contents

A	2
Access Operator	2
Authentication	2
Authorization (cluster-wide)	2
C	4
Clients CA	4
Cluster CA	4
Cluster Operator	4
Cruise Control	4
D	5
Drain Cleaner	5
E	6
Encryption	6
Entity Operator	6
F	7
Feature gate	7
K	8
Kafka (custom resource)	8
HTTP Bridge	8
KafkaBridge (custom resource)	8
KafkaConnect (custom resource)	8
KafkaConnector (custom resource)	8
KafkaExporter	8
KafkaMirrorMaker2 (custom resource)	8
KafkaNodePool (custom resource)	9
KafkaRebalance (custom resource)	9
KafkaTopic (custom resource)	9
KafkaUser (custom resource)	9
L	10
Listener	10
Logging (configuration)	10
M	11
Metrics	11
Metrics Reporter	11
N	12
Network policy	12
S	13
Storage (configuration)	13

Strimzi API schema	13
Strimzi Operator.....	13
Strimzi operators	13
StrimziPodSet (custom resource).....	13
Super user	13
T	14
Tiered storage.....	14
Topic Operator	14
U	15
Upgrade	15
User Operator	15

This glossary explains terminology unique to Strimzi and its components. It supports consistent usage and understanding across Strimzi documentation. Terms from Kafka and Kubernetes are outside its scope.

A

Access Operator

An optional operator that simplifies the sharing of Kafka connection information and credentials between namespaces. Connection details are stored centrally in a `Secret` resource.

Authentication

Defines how clients prove their identity to the Kafka cluster. Strimzi manages authentication as a client-server relationship:

- **Server-Side:** The Kafka cluster's listeners are configured to require a specific authentication type.
- **Client-Side:** A client (a `KafkaUser` or a client-based Kafka component managed by Strimzi) must be configured to provide matching credentials.

Listener authentication (Server-Side)

Listener authentication is configured per listener in the `spec.kafka.listeners` array of the `Kafka` custom resource. Supported types include `tls`, `scram-sha-512`, and `custom`.

Client authentication (Kafka user)

For Kafka users, authentication is managed using the `KafkaUser` custom resource. Supported types are `tls`, `tls-external` (using an external CA), and `scram-sha-512`. Strimzi automatically creates the necessary `Secret` resources for the user.

Client authentication (Kafka components)

For Strimzi-managed components, authentication is managed in the custom resource of the component, such as `KafkaConnect`. Supported types include `tls`, `scram-sha-256`, `scram-sha-512`, `plain`, and `custom`.

For more information, see the following:

- [Configuring client authentication on listeners](#)
- [Configuring user authentication](#)

Authorization (cluster-wide)

Defines which actions an authenticated client is permitted to perform on Kafka resources, such as writing to or reading from a topic. Configuration involves setting a cluster-wide mechanism and then, if required, defining user-specific rules.

Cluster-wide authorization

This defines the overall mechanism used by the Kafka cluster to control client actions. It's configured in the `spec.kafka.authorization` section of the `Kafka` custom resource. Supported types include `simple` (using the Kafka's built-in authorizer) and `custom` (using custom

authorizers).

User authorization (ACLs)

This defines specific Access Control Lists (ACLs) for a user, granting permissions to perform actions on `Kafka` resources. The ACLs are defined in the `spec.authorization` section of the `KafkaUser` custom resource. If using a custom authorization mechanism, user permissions are typically managed within the external authorization system and not through the `KafkaUser` resource.

For more information, see the following:

- [Configuring authorized access to Kafka](#)
- [Configuring user authorization](#)

C

Clients CA

A Certificate Authority managed by the Strimzi Cluster Operator that issues TLS certificates for Kafka clients. These certificates are used for mutual TLS (mTLS) authentication between external clients and Kafka brokers.

Cluster CA

A Certificate Authority managed by the Strimzi Cluster Operator that issues TLS certificates to secure communication between Kafka brokers, internal components, and Kafka clients. These certificates enable encrypted and authenticated communication over TLS.

Cluster Operator

The central operator responsible for deploying and managing Kafka clusters, Kafka Connect, Kafka MirrorMaker, and related components.

For more information, see the following:

- [Cluster Operator](#)
- [Configuring the Cluster Operator](#)

Cruise Control

A component that provides automated Kafka cluster rebalancing and optimization. Cruise Control is configured through the `Kafka` custom resource, while rebalancing operations are managed using the `KafkaRebalance` custom resource.

For more information, see the following:

- [Using Cruise Control for cluster rebalancing](#)
- [Using Cruise Control to modify topic replication factor](#)
- [Using Cruise Control to reassign partitions on JBOD disks](#)

D

Drain Cleaner

A utility installed as a separate component that ensures safe pod evictions during rolling updates to prevent data loss or downtime.

For more information, see [Evicting pods with the Strimzi Drain Cleaner](#).

E

Encryption

Strimzi supports Transport Layer Security (TLS) to encrypt communication between Kafka and its clients. TLS is enabled per listener in the `Kafka` custom resource, and communication between internal components is always encrypted.

Entity Operator

The Entity Operator runs the Topic Operator and User Operator in separate containers within its pod, allowing them to handle topic and user management.

F

Feature gate

Used to enable or disable specific features and functions managed by Strimzi operators. New features may be introduced initially through feature gates.

For more information, see [Feature gates](#).

K

Kafka (custom resource)

A custom resource for deploying and configuring a Kafka cluster, including settings for nodes, listeners, storage, security, and internal components like Cruise Control and the Entity Operator.

For more information, see the [Kafka schema reference](#).

HTTP Bridge

Provides a RESTful interface that allows HTTP-based clients to interact with a Kafka cluster.

For more information, see [Using the HTTP Bridge](#).

KafkaBridge (custom resource)

A custom resource used to deploy and configure a HTTP Bridge instance, specifying replicas, authentication, and connection details.

For more information, see the [KafkaBridge schema reference](#).

KafkaConnect (custom resource)

A custom resource used to deploy and configure a Kafka Connect cluster for integrating external systems with Kafka.

For more information, see the [KafkaConnect schema reference](#).

KafkaConnector (custom resource)

A custom resource for managing individual Kafka connectors in a Kafka Connect cluster declaratively and independently of the [KafkaConnect](#) deployment.

For more information, see the [KafkaConnector schema reference](#).

KafkaExporter

The Kafka Exporter exposes Kafka metrics for Prometheus. It is configured as part of the [Kafka](#) custom resource.

For more information, see the [KafkaExporterSpec schema reference](#).

KafkaMirrorMaker2 (custom resource)

A custom resource for deploying a Kafka MirrorMaker 2 instance to replicate data between Kafka

clusters.

For more information, see the [KafkaMirrorMaker2 schema reference](#).

KafkaNodePool (custom resource)

A custom resource used to configure distinct groups of nodes within a Kafka cluster. Nodes in a node pool can be configured to operate as Kafka brokers, controllers, or both.

For more information, see the [KafkaNodePool schema reference](#).

KafkaRebalance (custom resource)

A custom resource that triggers and manages cluster rebalancing through Cruise Control by setting optimization goals.

Rebalance modes:

full

Load rebalanced across all brokers

add-brokers

Replicas moved to newly added brokers

remove-brokers

Replicas moved off brokers being removed

remove-disks

Data moved between storage volumes within the same broker

For more information, see the [KafkaRebalance schema reference](#).

KafkaTopic (custom resource)

A custom resource for managing Kafka topics (creation, configuration, deletion) through the Topic Operator.

For more information, see the [KafkaTopic schema reference](#).

KafkaUser (custom resource)

A custom resource for managing Kafka users (creation, configuration, deletion) through the User Operator, including their authentication credentials and access permissions.

For more information, see the [KafkaUser schema reference](#).

L

Listener

Defines how clients connect to the Kafka cluster. Strimzi supports several listener types for exposing Kafka internally or externally.

Listener types:

internal

Kafka exposed only within the Kubernetes cluster

route

Kafka exposed externally using OpenShift Routes

loadbalancer

Kafka exposed externally using a **LoadBalancer** service

nodeport

Kafka exposed externally using **NodePort** services

ingress

Kafka exposed externally using Kubernetes NGINX Ingress with TLS passthrough

cluster-ip

Kafka exposed using a per-broker **ClusterIP** service

Logging (configuration)

Logging for Kafka components and Strimzi operators is configured through their custom resources. The configuration uses Log4j2 and supports dynamic updates without restarting pods.

Configuration methods:

inline

Loggers and levels are defined directly in the custom resource. Used for simple changes to log levels.

external

Loggers and levels are defined in a **ConfigMap** referenced by the custom resource. Used for complex, reusable, or filtered configurations.

M

Metrics

Strimzi components can expose Prometheus-formatted metrics for monitoring. Metrics for components are enabled through its custom resource.

For more information, see [Introducing metrics](#).

Metrics Reporter

A component that exposes metrics from Strimzi-managed components such as Kafka brokers, Kafka Connect, Kafka MirrorMaker 2, and HTTP Bridge in Prometheus format. The Metrics Reporter is enabled through the `metricsConfig` property in the corresponding custom resource.

N

Network policy

Strimzi automatically creates a `NetworkPolicy` resource for each listener, allowing connections from all namespaces by default. You can configure the `networkPolicyPeers` property to restrict access to specific applications or namespaces.

S

Storage (configuration)

Defines disk storage for Kafka nodes within the `KafkaNodePool` custom resource.

Supported storage types:

ephemeral

Temporary storage tied to the pod lifecycle

persistent-claim

Durable storage using PersistentVolumeClaims (PVCs)

jbod

Multiple disks or volumes (ephemeral or persistent)

For more information, see [Configuring Kafka storage](#).

Strimzi API schema

The formal specification that defines the structure, properties, and validation rules for Strimzi custom resources. Also referred to as the **Strimzi custom resource schema**.

Strimzi Operator

The primary deployment artifact for Strimzi. An operator that installs and configures components for running Kafka on Kubernetes, including the Cluster Operator.

Strimzi operators

The suite of Kubernetes operators (Cluster Operator, Topic Operator, User Operator) that automate Kafka cluster management.

StrimziPodSet (custom resource)

A custom resource used by the Strimzi Cluster Operator to manage the lifecycle of broker pods, replacing Kubernetes `StatefulSet` resources to provide greater control over pod identity and updates.

Super user

A Kafka user principal with full administrative access that bypasses all ACL checks. Super users are configured via the `superUsers` property in the `Kafka` custom resource when simple authorization is enabled.

T

Tiered storage

A capability enabling Kafka brokers to store topic log segments across different storage tiers, such as local disk and remote object storage. It is configured through the `Kafka` custom resource.

For more information, see the following:

- [Tiered storage](#)
- [TieredStorageCustom schema reference](#)

Topic Operator

The operator responsible for managing Kafka topics through `KafkaTopic` custom resources.

- [Topic Operator](#)
- [Using the Topic Operator to manage Kafka topics](#)

U

Upgrade

The process of updating the Cluster Operator and the Kafka cluster it manages. Upgrade typically involves upgrading the operator first, then the Kafka version, and finally the metadata version.

Upgrade paths:

incremental upgrade

Move between consecutive minor versions

multi-version upgrade

Skip one or more minor versions

For more information, see [Upgrading Strimzi](#).

User Operator

The operator responsible for managing Kafka users and ACLs through `KafkaUser` custom resources.

- [User Operator](#)
- [Using the User Operator to manage Kafka users](#)