Kafkawize Documentation

Release 5.0.3

Read the Docs

Aug 27, 2022

Contents

Kafkawize is a Self service Apache Kafka Topic Management tool/portal. It is a web application which automates the process of creating and browsing Kafka topics, acls, schemas by introducing roles/authorizations to users of various teams of an organization. It is Open source. Read more...

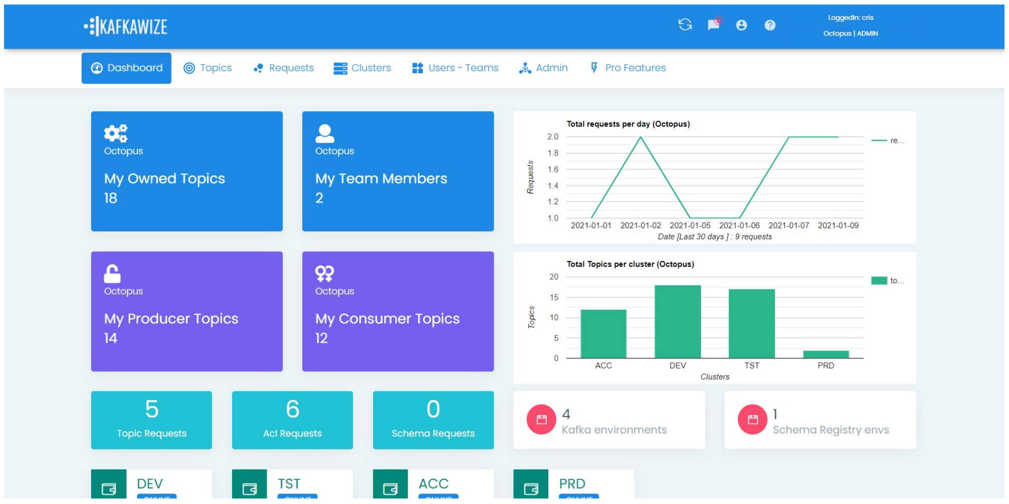
Contents:

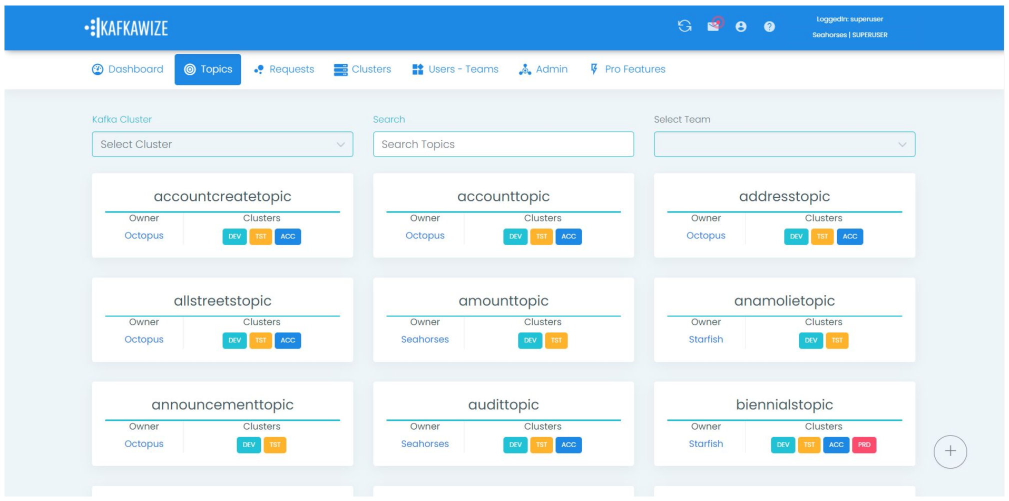
CHAPTER 1

Introduction

1.1 What is Kafkawize?

Kafkawize is a Self service Apache Kafka Topic Management tool/portal. It is a web application which automates the process of creating and browsing Kafka topics, acls, schemas by introducing roles/authorizations to users of various teams of an organization. It is Open Source. Below here are couple of screen shots of Dashboard and Topics pages.





Kafka Cluster: Select Cluster
Search: Search Topics
Select Team

accountcreatetopic
Owner: Octopus
Clusters: DEV TST ACC

accounttopic
Owner: Octopus
Clusters: DEV TST ACC

addresstopic
Owner: Octopus
Clusters: DEV TST ACC

allstreetstopic
Owner: Octopus
Clusters: DEV TST ACC

amounttopic
Owner: Seahorses
Clusters: DEV TST

anamolietopic
Owner: Starfish
Clusters: DEV TST

announcementtopic
Owner: Octopus
Clusters: DEV TST

audittopic
Owner: Seahorses
Clusters: DEV TST ACC

biennialstopic
Owner: Starfish
Clusters: DEV TST ACC PRD

1.2 Background

Let me give some background. Apache Kafka is widely being adapted in organizations irrespective of the scale. It's unique features like scalability, retention and reliability unlike the traditional messaging platforms, makes it stand out.

While adapting Kafka, you would notice there are few manual activities like creating topics, acls, updating configuration etc. There have been few topic management tools in the market which provide automation to certain extent, however its not complete. Assume there is a team requesting for a kafka topic with in an organization. They would either request for it through email with a template or any user interface. The actual kafka team in the organization executes the commands to create topics. With Development, Test, Acceptance and Production environments, managing these configurations, by the Kafka team and handling all the requests from various teams would become a hassle as you see the growth of teams using Kafka. It is just not manual activity, but Audit, time and effort, the team has to put in.

Apart from that, this meta data (Kafka topics, acls) which is stored in ZK could be easily stored in a good readable format and can be used as source of truth.

Hence I saw the need of Kafkawize – a self service topic management tool., to keep ZERO interaction by other Teams with Kafka team for topics, acls or schemas creation by introducing roles and authorizations to users of teams. It is developed with a simple front end, and backend with Spring boot. It saves time and effort, governs information, avoids risk in kafka config loss, avoids manual activities and mistakes, maintains up to date SOT (source of truth).

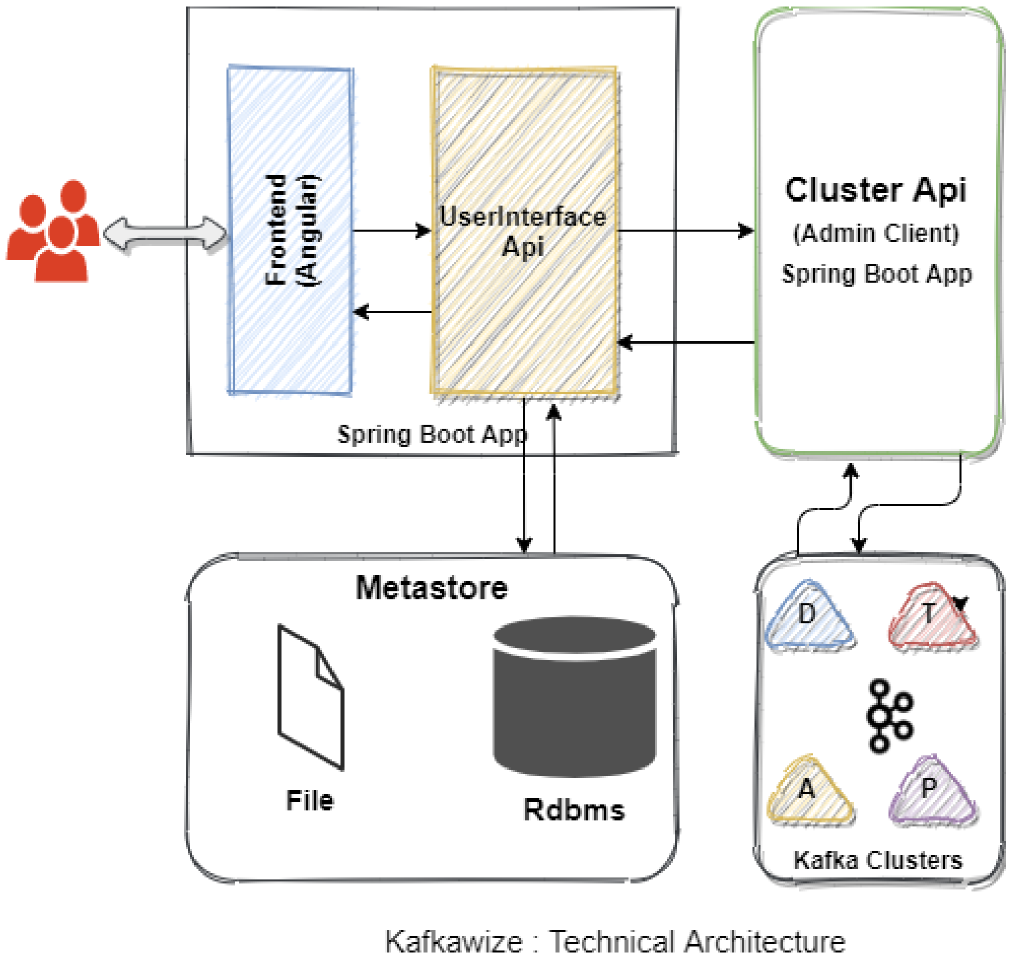
1.3 Key features

```
-    4 eyed principle - Requesting and Approving topics/acls/schemas
-    Spring based security
-    Fully automated
-    Browse Acls,  Producers and Consumers
-    Synchronize Source of truth with Meta store
-    Support for an Rdbms or File system
```

1.4 How is it developed

Simple technologies have been used to develop Kafkawize. Front end with Angular, HTML, CSS, MaterialView bootstrap theme and Javascript. Backend with Java, Kafka Admin client libraries, Spring boot and Spring security frameworks. First version of Kafkawize has been released in Sep 2018 with basic features., and gradually enhanced it.

1.5 Architecture



Kafkawize contains two main Apis (User Interface API and Cluster management API) and a Front end.

User Interface Api directly communicates between Frontend UI and Cluster API.

Front end is built with AngularJs, HTML, and Java script.

Cluster API acts as middle layer between Kafka brokers and UserInterface API.

Cluster API creates Kafka Admin Client and executes the requests for Topic, Acls or Schema registry. Kafka Admin client libraries are used to create the client.

File database (H2 - Mysql style) Or RDBMS(MySql for ex) datastore stores all the meta information like users, teams, topicRequests, request and execution data from all the users., and to maintain source of truth.

Requests from users are directed to cluster api., and also data is stored metastore.

On the backend side, Spring Security, Spring Boot frameworks, Hibernate are used to develop this application.

1.6 Git Repositories

UserInterface Api : https://github.com/muralibasani/kafkawize

Cluster Api : https://github.com/muralibasani/kafkawizeclusterapi

1.7 Developer

Muralidhar Basani

LinkedIn : https://www.linkedin.com/in/muralibasani/

Web : https://kafkawize.com

Email : kafkawize@gmail.com

CHAPTER 2

Getting Started

2.1 Demo 5.0.3 (Prev)

Lets start with setting up Kafkawize on your Windows/Linux/Unix systems.

2.2 Prerequisites

- Any IDE like IntelliJ/Netbeans or Eclipse to build the applications.
- Java installed machines to run the applications
- Rdbms (like Oracle/MySQL..) or File based

2.3 1 Download Kafkawize

Download the latest version (5.0.3) from github

https://github.com/muralibasani/kafkawize

https://github.com/muralibasani/kafkawizeclusterapi

This bundle includes the libraries of kafkawizeuapi and kafkawizeclusterapi

2.4 2 Download Metastore

If Rdbms is used as metastore, download

Mysql : https://dev.mysql.com/downloads/

Oracle : https://www.oracle.com/database/technologies/oracle-database-software-downloads.html of file based/

Or any other Rdbms is ok.

2.5 3 KW Cluster Api Application

This Api does the below

- Receives and responds to calls from UI Api
- Connects to Kafka Brokers with Kafka AdminClient Api
- There is no connection to any metastore
- Swagger enabled

run mvn clean install generates the jar (kafkawizeclusterapi-5.0.3.jar)

2.6 5 Configure Kafkawize UI Api Application

This Api does the below

- Users interact with UI interface with this Api
- All the end points in this application either connect to Metastore or Cluster Api or both

Build mvn clean install

Configure application properties:

  In application.properties

default port is 9097, if port has to be changed, else

```
server.port:9097
```

Set metastore to rdbms (from step 4):

```
kafkawize.db.storetype=rdbms
```

- Install and run Rdbms (like Mysql/Oracle) and create a db schema or database

2.6.1 Configure Cluster Api

configure cluster api host and port details:

```
kafkawize.clusterapi.url:http://localhost:9343
```

2.7 6 Metastore setup

2.7.1 Metastore Rdbms

Configure in application properties:: # Database settings # To create all the required tables of Kafkawize. Need grants to create and alter. Values true/false:

```
kafkawize.dbscripts.create.tables=false
```

# db scripts execution 'auto' or 'manual'. If set to manual, user has to execute the scripts from resources dir manually:

```
kafkawize.dbscripts.insert.basicdata=false
```

# Location of db scripts for insert or create:

```
kafkawize.dbscripts.location=./scripts/base/rdbms/
```

- Install and run Mysql/Oracle and create a db schema or database
- Create tables and run insert scripts in Database

  /scripts/base/rdbms/ddl-jdbc.sql

  /scripts/base/rdbms/insertdata.sql
- Above scripts will create tables and insert initial set of Environments, Teams and Users which you can delete anytime from UI.

The jar (kafkawize-5.0.3.jar) is available in the downloaded bundle.

2.8 7 Run KW and KWClusterApi

Run:

```
java -jar kafkawizeclusterapi-5.0.3.jar --spring.config.location=classpath:/
↪application.properties
```

Run:

```
java -jar kafkawize-5.0.3.jar --spring.config.location=classpath:/application.
↪properties
```

If application is running, you can access UI from http://{[}host{]}:{[}port{]}/kafkawize

2.9 8 Kafka Connectivity

Cluster Api Application connects to Kafka brokers with Kafka AdminClient Api., and needs Describe access on all topics through the cluster. Hence the below wildcard acl has to be executed.

- If Acls are enabled on Kafka brokers, make sure "Cluster Api" application host is authorized to read topics (A read Acl is enough on the topic)

  Examples SSL Based Acl (Note of double quotes in the below command if copied properly):

  bin/kafka-acls –authorizer-properties zookeeper.connect=localhost:2181 –add –allow-principal User:CN=MO,OU=MO,O=WA,L=WA,ST=WA,C=HO" –operation All –topic "*" –cluster Cluster:kafka-cluster

  Examples IP Based Acl:

  bin/kafka-acls –authorizer-properties zookeeper.connect=localhost:2181 –add –allow-principal User:"*" –allow-host 127.0.0.1 –operation All –topic "*" –cluster Cluster:kafka-cluster

2.10 9 Final Check

- Cluster Api is running
- Metastore (Rdbms or file system) is running and has tables and data
- UI Api is running
- Cluster Api is authorized to read topics and acls on topics information(Acls should be configured)
- Access UI from http://{[}host{]}:{[}port{]}/kafkawize where host and port are UI Api application Example : http://localhost:9097/kafkawize

  Default users, passwords and roles:

  superadmin/kwsuperadmin123$$ (also configured in application.properties)

CHAPTER 3

Migration Kafkawize from earlier versions to 5.0.3

There are several database changes to migrate from earlier versions to 5.0.3.

Hence we suggest to drop all and recreate.

CHAPTER 4

Features

Kafkawize is built with the key features as below:

```
-    4 eyed principle - Requesting and Approving topics/acls/schemas
-    Spring based security
-    Fully automated
-    Browse Topics, Acls, Schemas, Connectors,  Producers and Consumers
-    Synchronize Source of truth with Meta store, Restore config
-    Support for an Rdbms or file system as metastore
```

4.1 Security

There are two main applications of Kafkawize. UiApi and ClusterApi. There is a security applied on Cluster Api. End points are exposed to anyone who can reach the host and has access., and swagger is enabled too. Users can directly post/get requests to this application.

On the UI Api, Spring security is enabled. As the application boots, it loads all the users from database into its memory and would be referred to it.

Security on the Kafka Broker is out of scope for Kafkawize., , however for Kafkawize to connect to Kafka broker, Acls/AdminClient properties need to be provided to avoid authorization issues.

4.2 Teams and Role Management

Kafkawize can be configured with any number of roles and permissions:

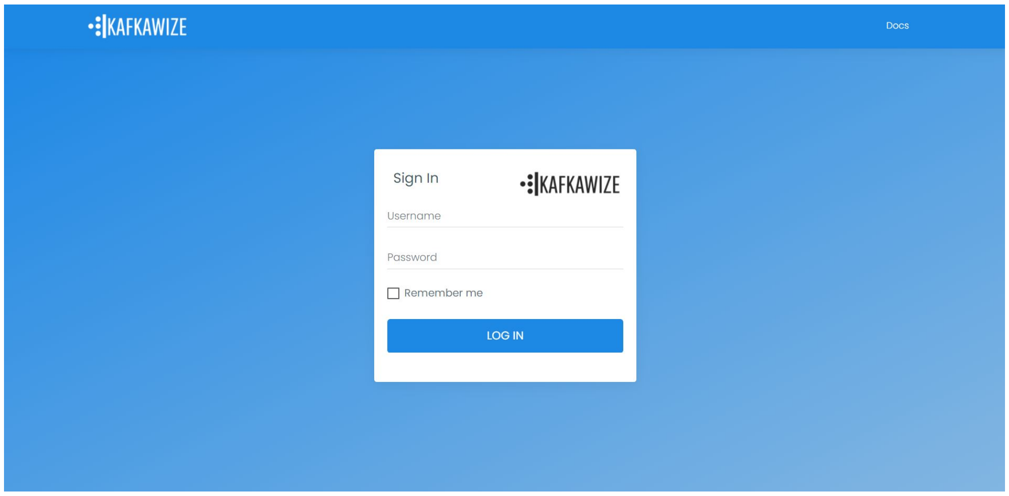
By default 'USER' roles can request and approve topic/acl/schema/connector requests.

4.3 Login

Portal can be accessed only if the user logs in. New users are created by Super users.

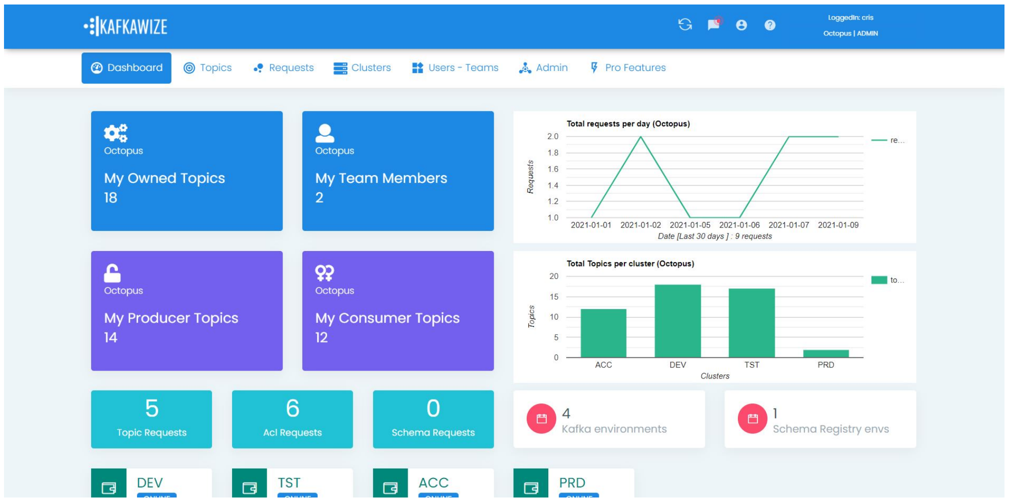
After a user is logged in, a session is created for the user, and depending on the role , he/she is authorized to perform certain actions.

Default login url : http://localhost:9097/kafkawize



4.4 Dashboard

After a user is logged in , he/she will be redirected to homepage.



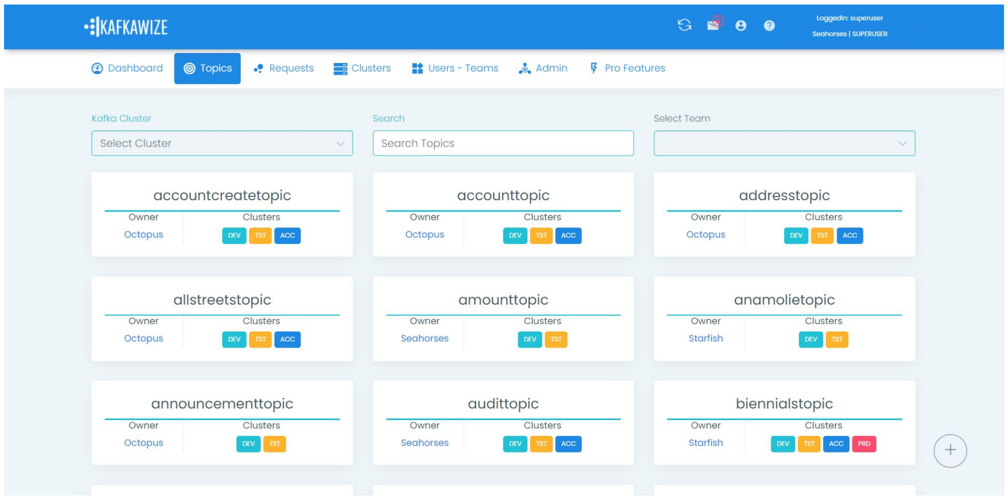
4.5 Browse

All the logged in users with any roles can browse the below components.

4.5.1 Topics

A user is allowed to a select an environment, and the topics are loaded. If you do not see the topics for some reason, check for any errors in Cluster Api application and UI Api application.

A filter option is also provided and users can search for specific topics.



Topic	Owner	Clusters
accountcreatetopic	Octopus	DEV TST ACC
accounttopic	Octopus	DEV TST ACC
addresstopic	Octopus	DEV TST ACC
allstreetstopic	Octopus	DEV TST ACC
amounttopic	Seahorses	DEV TST
anamolietopic	Starfish	DEV TST
announcementtopic	Octopus	DEV TST
audittopic	Seahorses	DEV TST ACC
biennialstopic	Starfish	DEV TST ACC PRD

Every topic has a topic overview page, which displays topic partitions, replication factor in all available environments.

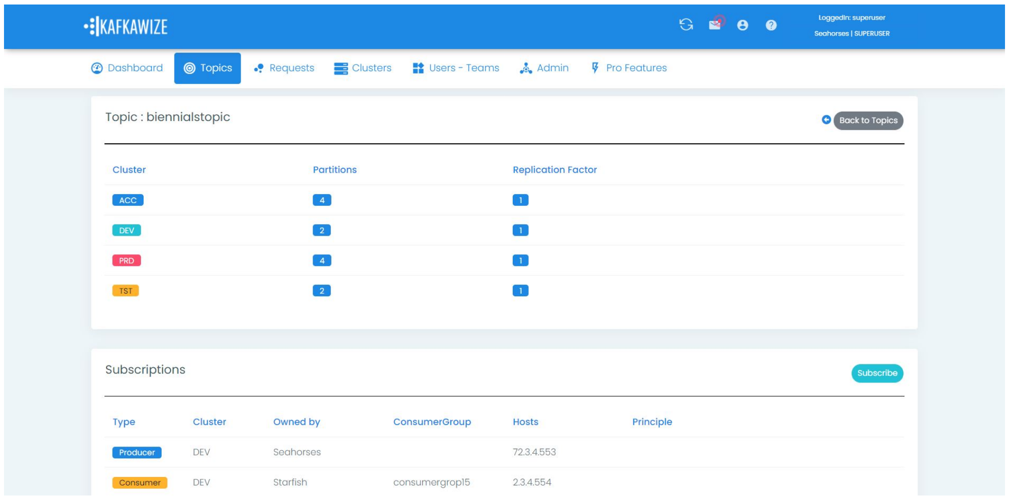
Every topic has a topic overview page, which displays topic subscriptions in all available environments.

Note : Make sure kafkawize has access to read topics (Acl is required if acls are enabled on the cluster)

4.5.2 Acls

Acl - Access Controls

From the Browse topics screen, user can select and clicking on a topic displays all the available subscriptions of that topic.



Topic : biennialstopic

Cluster	Partitions	Replication Factor
ACC	4	1
DEV	2	1
PRD	4	1
TST	2	1

Subscriptions

Type	Cluster	Owned by	ConsumerGroup	Hosts	Principle
Producer	DEV	Seahorses		72.3.4.553	
Consumer	DEV	Starfish	consumergrop15	2.3.4.554	

Acls are either Producers or Consumers. Producers have write access on the topics and Consumers have read access on the topics.

Acls are either IP Address based or SSL based. If IP based, you should see the IP addresses of client who has access. If SSL based, the DN name should be seen.

Every Acl should be owned by a team. If no team is assigned to an Acl, it can be assigned by Synchronize Acl option, but by a Super user.

There can be multiple producers and multiple consumers for a topic owning by different teams.

4.6 Requests

All the users can request for topics, acls or avro schemas in any environment.

Topic Requests can be approved by Admins or Super users from the same team.

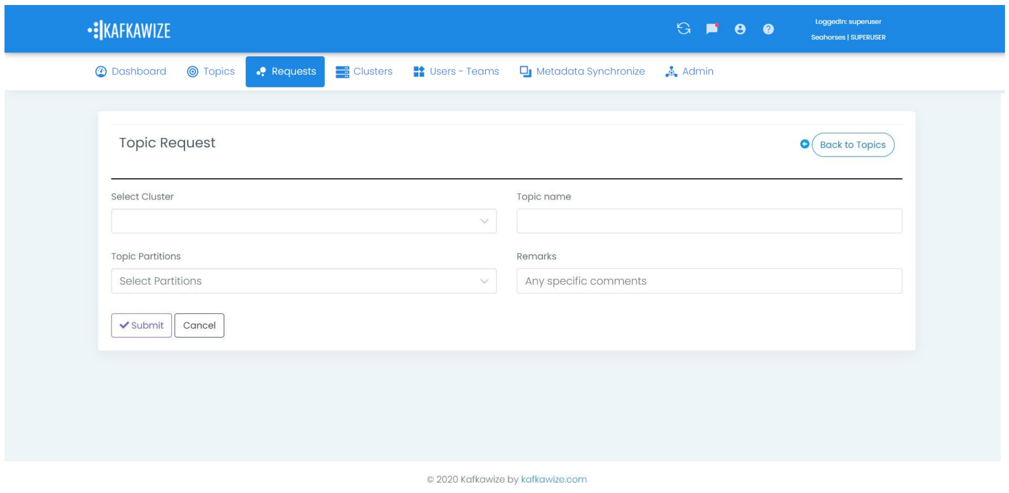
Acl(Subscription) Requests can be approved by Admins or Super users who belong to Topic Owner's team.

Schema Requests can be approved by Admins or Super users from the same team.

- Notifications: After every request is raised, there is a notification (bell icon) at the top for the relevant user who can approve these requests.

4.6.1 Topics

To request for a topic, all the mandatory parameters need to be provided.



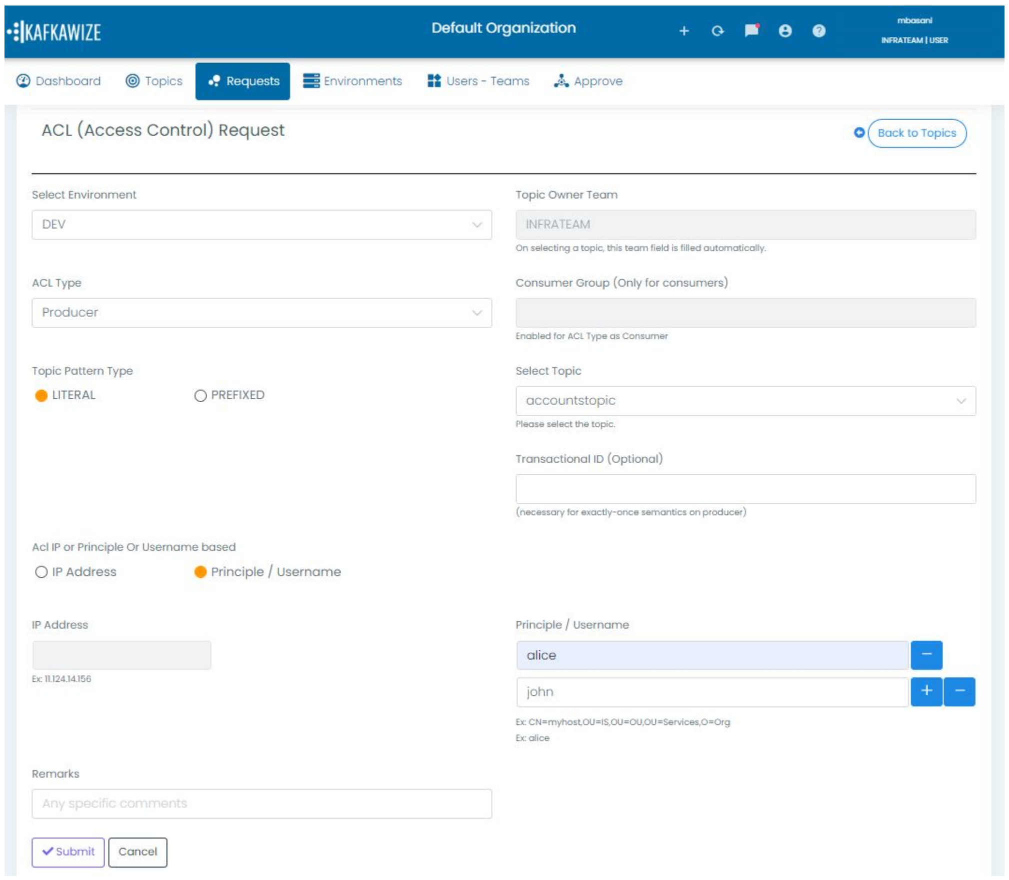
Topic name, partitions, owning team, application name, and if any remarks. On submission you can view your requests in Audit/My Requests., and it can be deleted if required.

Default partitions size, maximum partitions size and default replication factor can be configured per environment in Clusters page.

To request for a topic in higher environments, it is required for the topic to exist in Base Sync cluster (DEV). This can be configured with property custom.syncdata.cluster

4.6.2 Acls

To request for a acl, all the mandatory parameters need to be provided.



ACL (Access Control) Request

Select Environment	DEV
Topic Owner Team	INFRATEAM
	On selecting a topic, this team field is filled automatically.
ACL Type	Producer
Consumer Group (Only for consumers)	
	Enabled for ACL Type as Consumer
Topic Pattern Type	[✓ LITERAL] PREFIXED
Select Topic	accountstopic
	Please select the topic.
Transactional ID (Optional)	
	(necessary for exactly-once semantics on producer)
Acl IP or Principle Or Username based	IP Address [✓ Principle / Username]
IP Address	
	Ex: 11.124.14.156
Principle / Username	alice
	john
	Ex: CN=myhost,OU=IS,OU=OU,OU=Services,O=Org
	Ex: alice
Remarks	Any specific comments

After a user clicks on a topic, user can view all the subscriptions and a button to subscribe to the topic.

The below fields are required.

Acl type Producer or Consumer.

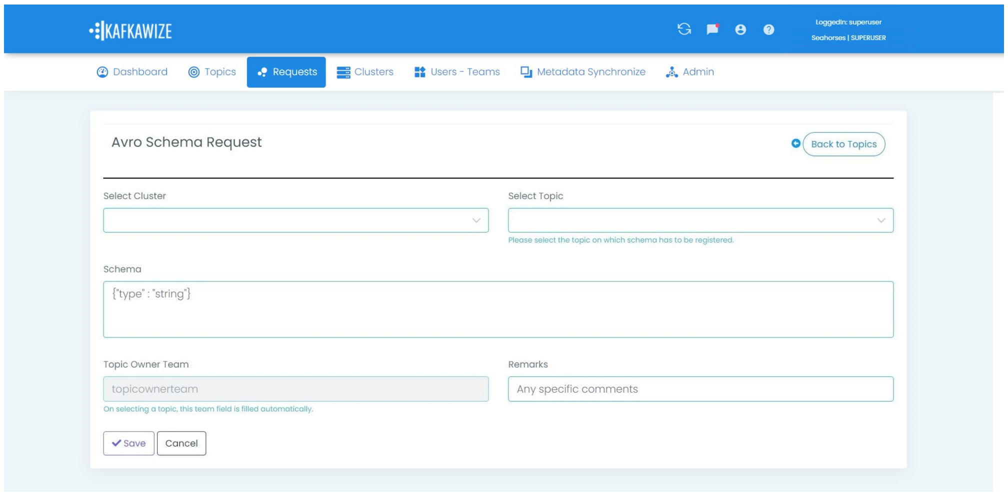
IP Addresses or SSL DN name, or username should be provided for authorizations.

Multiple IP addresses, or principles or usernames can be filled at once

On submission you can view your requests in Audit/My Requests., and it can be deleted if required.

4.6.3 Avro Schemas

An Avro schema can also be requested for a particular subject (topic).

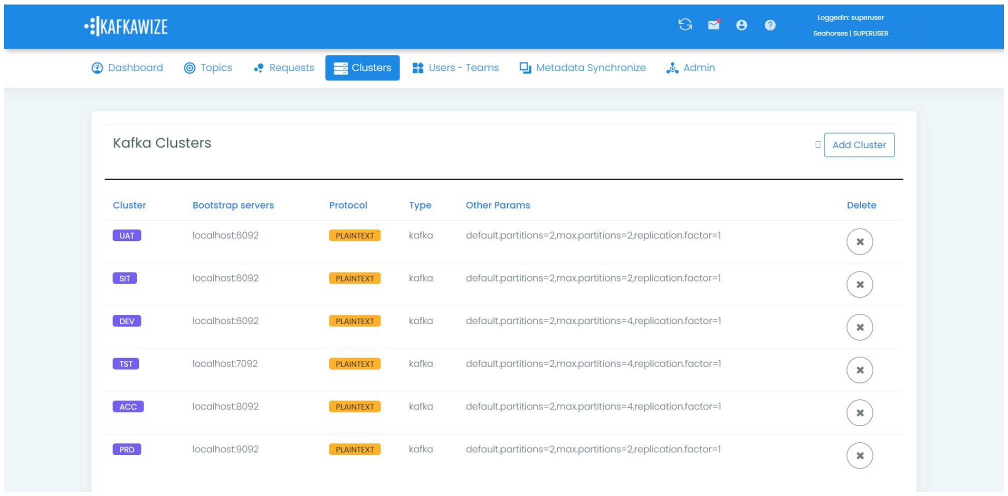


Avro Schema Request

Select Cluster	Select Topic
	Please select the topic on which schema has to be registered.
Schema	{"type" : "string"}
Topic Owner Team	topicownerteam
	On selecting a topic, this team field is filled automatically.
Remarks	Any specific comments

4.7 Clusters

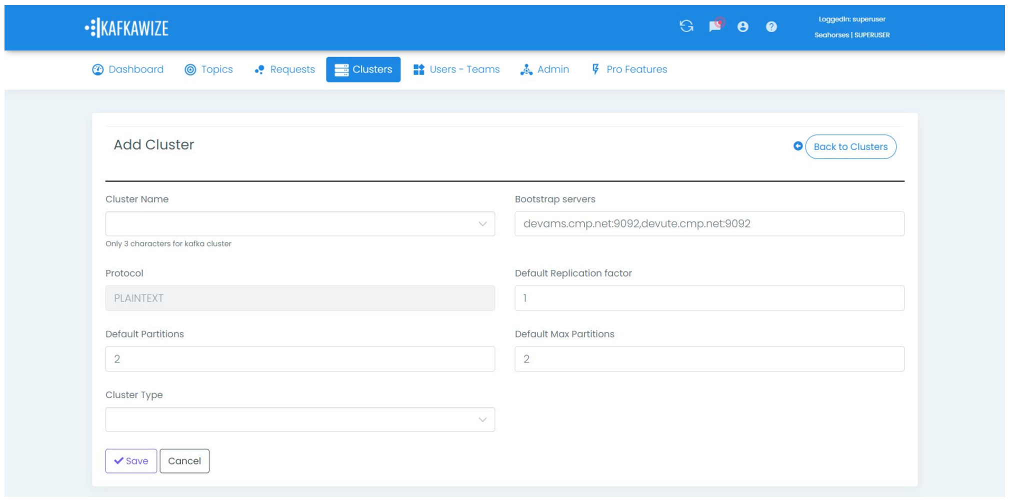
All the clusters are of type 'kafka' or 'schemaregistry'.



Kafka Clusters

Cluster	Bootstrap servers	Protocol	Type	Other Params	Delete
UAT	localhost:6092	PLAINTEXT	kafka	default.partitions=2,max.partitions=2,replication.factor=1	✖
SIT	localhost:6092	PLAINTEXT	kafka	default.partitions=2,max.partitions=2,replication.factor=1	✖
DEV	localhost:6092	PLAINTEXT	kafka	default.partitions=2,max.partitions=4,replication.factor=1	✖
TST	localhost:7092	PLAINTEXT	kafka	default.partitions=2,max.partitions=4,replication.factor=1	✖
ACC	localhost:8092	PLAINTEXT	kafka	default.partitions=2,max.partitions=4,replication.factor=1	✖
PRD	localhost:9092	PLAINTEXT	kafka	default.partitions=2,max.partitions=2,replication.factor=1	✖

New clusters can be added by Superusers only. There is a other parameters field added, into which default partitions, max partitions size and replication factor can be added.



Add Cluster

Cluster Name	
Only 3 characters for kafka cluster	
Bootstrap servers	devams.cmp.net:9092,devute.cmp.net:9092
Protocol	PLAINTEXT
Default Replication factor	1
Default Partitions	2
Default Max Partitions	2
Cluster Type	

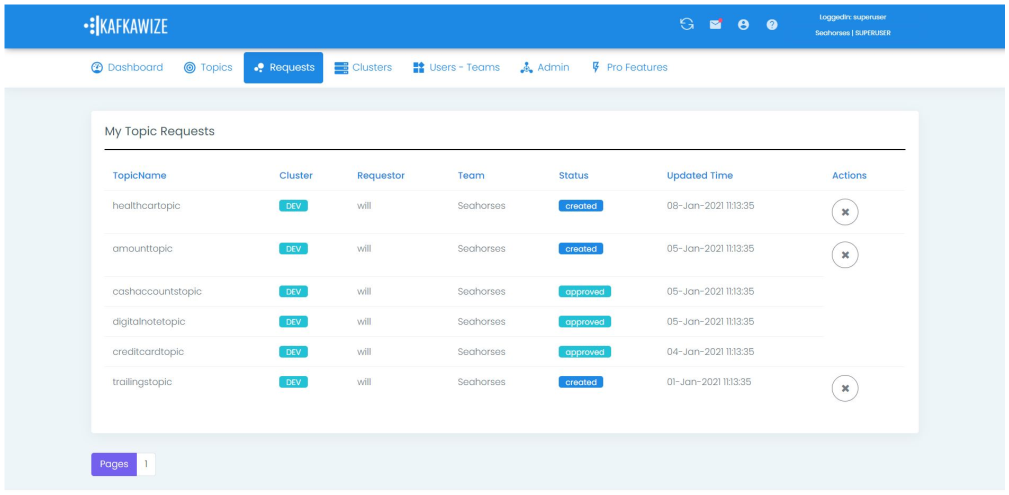
A cluster can be deleted by any Admin user or super user.

4.8 Audit

All the requests from users are audited and can be viewed.

4.8.1 My Topic Requests

Requests (Topics/Acls/Schemas/Connectors) from their own team can be viewed and deleted.

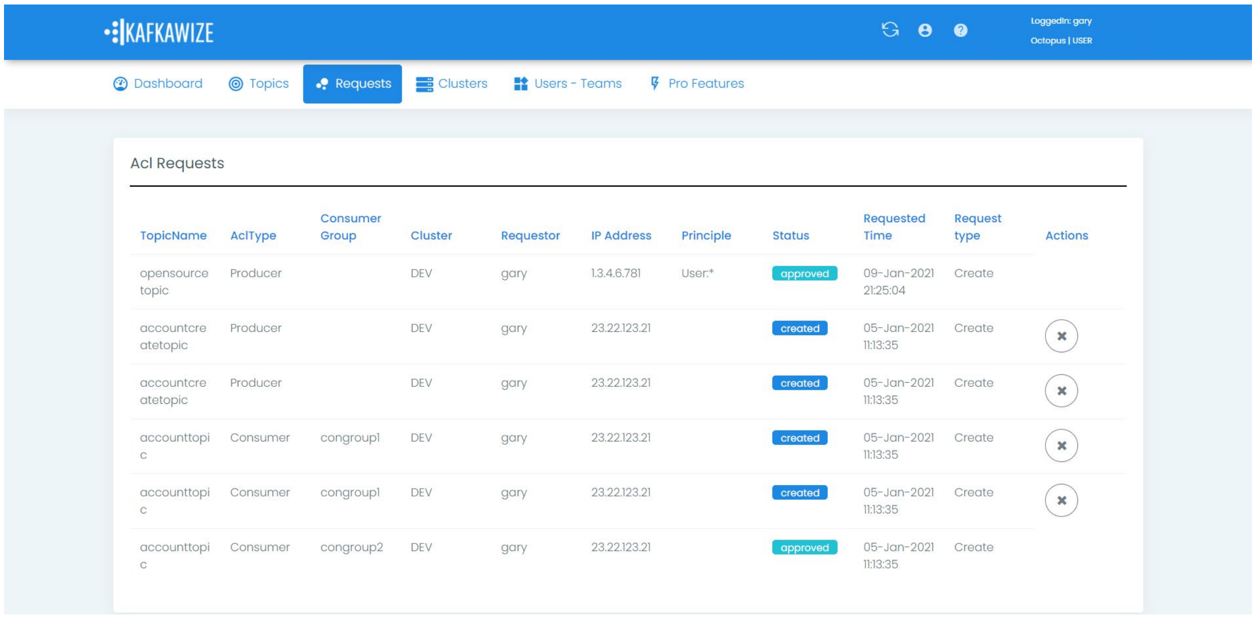


My Topic Requests

TopicName	Cluster	Requestor	Team	Status	Updated Time	Actions
healthcartopic	DEV	will	Seahorses	created	08-Jan-2021 11:13:35	✖
amounttopic	DEV	will	Seahorses	created	05-Jan-2021 11:13:35	✖
cashaccountstopic	DEV	will	Seahorses	approved	05-Jan-2021 11:13:35	
digitalnotetopic	DEV	will	Seahorses	approved	05-Jan-2021 11:13:35	
creditcardtopic	DEV	will	Seahorses	approved	04-Jan-2021 11:13:35	
trailingstopic	DEV	will	Seahorses	created	01-Jan-2021 11:13:35	✖

4.8.2 My Acl Requests

Requests (Topics/Acls/Schemas) from their own team can be viewed and deleted.



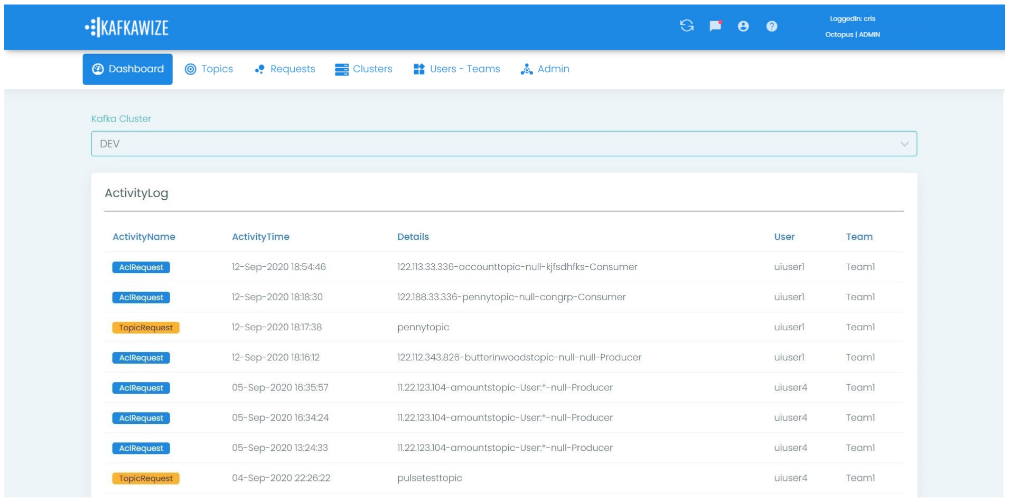
Acl Requests

TopicName	AclType	Consumer Group	Cluster	Requestor	IP Address	Principle	Status	Requested Time	Request type	Actions
opensource topic	Producer		DEV	gary	1.3.4.6.781	User:*	approved	09-Jan-2021 21:25:04	Create	
accountcre atetopic	Producer		DEV	gary	23.22.123.21		created	05-Jan-2021 11:13:35	Create	✖
accountcre atetopic	Producer		DEV	gary	23.22.123.21		created	05-Jan-2021 11:13:35	Create	✖
accounttopi c	Consumer	congroup1	DEV	gary	23.22.123.21		created	05-Jan-2021 11:13:35	Create	✖
accounttopi c	Consumer	congroup1	DEV	gary	23.22.123.21		created	05-Jan-2021 11:13:35	Create	✖
accounttopi c	Consumer	congroup2	DEV	gary	23.22.123.21		approved	05-Jan-2021 11:13:35	Create	

width 500px :align: center

4.8.3 Activity Log

All the requests requested and approved can be viewed. Users login/logout are not part of this log yet.



ActivityLog

ActivityName	ActivityTime	Details	User	Team
AclRequest	12-Sep-2020 18:54:46	122.113.33.336-accounttopic-null-kjfsdhfks-Consumer	uiuser1	Team1
AclRequest	12-Sep-2020 18:18:30	122.188.33.336-pennytopic-null-congrp-Consumer	uiuser1	Team1
TopicRequest	12-Sep-2020 18:17:38	pennytopic	uiuser1	Team1
AclRequest	12-Sep-2020 18:16:12	122.112.343.826-butterinwoodstopic-null-null-Producer	uiuser1	Team1
AclRequest	05-Sep-2020 16:35:57	11.22.123.104-amountstopic-User:*-null-Producer	uiuser4	Team1
AclRequest	05-Sep-2020 16:34:24	11.22.123.104-amountstopic-User:*-null-Producer	uiuser4	Team1
AclRequest	05-Sep-2020 13:24:33	11.22.123.104-amountstopic-User:*-null-Producer	uiuser4	Team1
TopicRequest	04-Sep-2020 22:26:22	pulsetesttopic	uiuser4	Team1

4.9 Approve Or Decline

Topic Requests can be approved or declined by users from the same team. After approval of a topic, it is created on the cluster, how ever no acls are assigned to it yet.

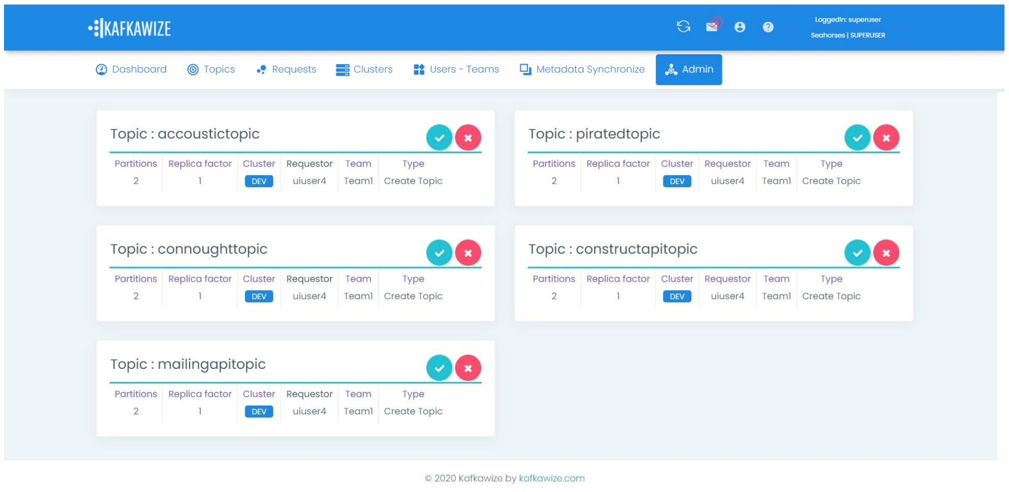
Acl Requests can be approved/declined by users who belong to Topic Owner's team. After approval, acls are created on the cluster.

Schema Requests can be approved by Admins or Super users from the same team.

Notifications are visible on the top right side of the portal.

4.9.1 Topics

If a topic is requested by 'user1' from 'Team1', it can be approved/declined by 'user2' from 'Team1' only., provided 'user2' has same role.



Topic : accoustictopic

Partitions	Replica factor	Cluster	Requestor	Team	Type
2	1	DEV	uiuser4	Team1	Create Topic

Topic : piratedtopic

Partitions	Replica factor	Cluster	Requestor	Team	Type
2	1	DEV	uiuser4	Team1	Create Topic

Topic : connoughttopic

Partitions	Replica factor	Cluster	Requestor	Team	Type
2	1	DEV	uiuser4	Team1	Create Topic

Topic : constructapitopic

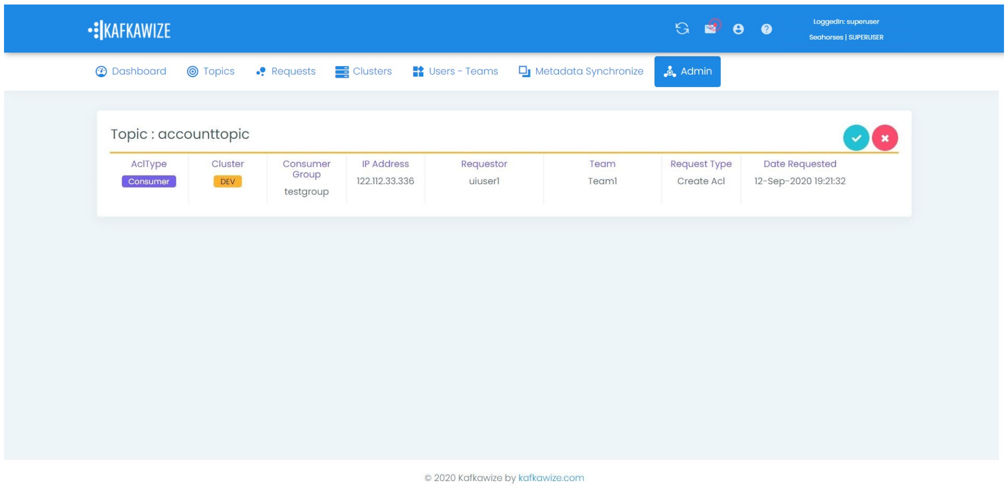
Partitions	Replica factor	Cluster	Requestor	Team	Type
2	1	DEV	uiuser4	Team1	Create Topic

Topic : mailingapitopic

Partitions	Replica factor	Cluster	Requestor	Team	Type
2	1	DEV	uiuser4	Team1	Create Topic

4.9.2 Acls

If a acl is requested by 'user1' on topic which is owned by 'Team2', it can be approved/declined by 'user2' from 'Team2' only., provided 'user2' has same role.



Topic : accounttopic

AclType	Cluster	Consumer Group	IP Address	Requestor	Team	Request Type	Date Requested
Consumer	DEV	testgroup	122.112.33.336	uiuser1	Team1	Create Acl	12-Sep-2020 19:21:32

4.9.3 Avro Schemas

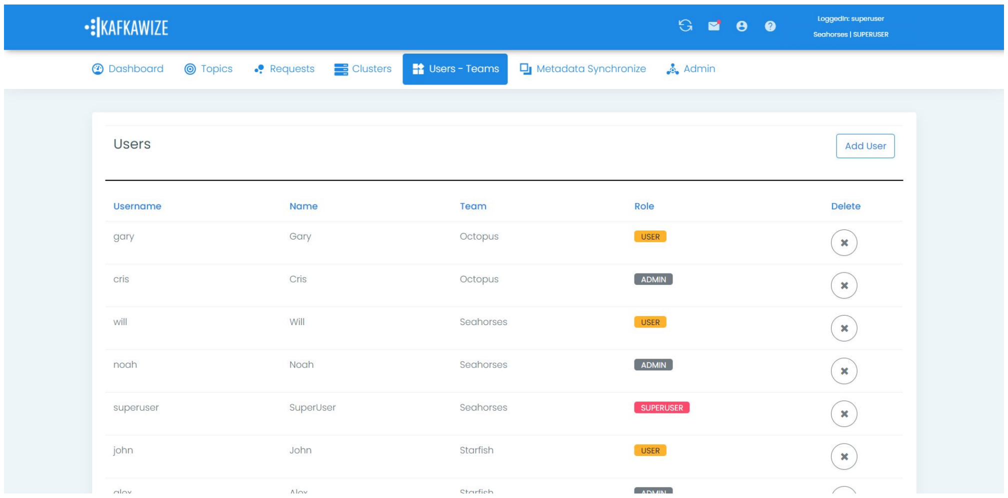
If a schema is requested by 'user1' from 'Team1', it can be approved by 'user2' from 'Team1' only., provided 'user2' has same role.

4.10 Users

All the users are visible to any logged in user. A new user can be added/deleted by only Super users.

4.10.1 View Users

From this page all the users can be seen and also be deleted.

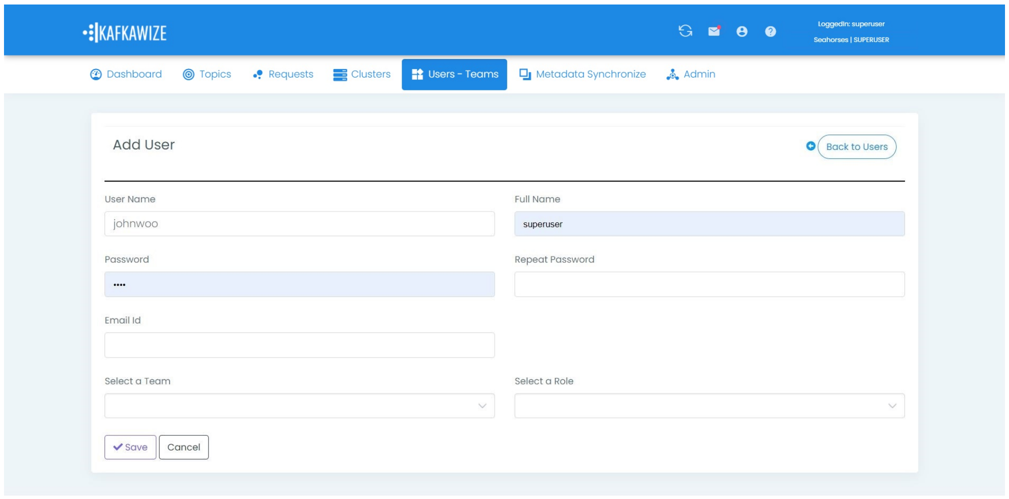


Users

Username	Name	Team	Role	Delete
gary	Gary	Octopus	USER	✖
cris	Cris	Octopus	ADMIN	✖
will	Will	Seahorses	USER	✖
noah	Noah	Seahorses	ADMIN	✖
superuser	SuperUser	Seahorses	SUPERUSER	✖
john	John	Starfish	USER	✖
alex	Alex	Starfish	ADMIN	

4.10.2 Add User

With this form a new user can be added by a SuperUser.



Add User

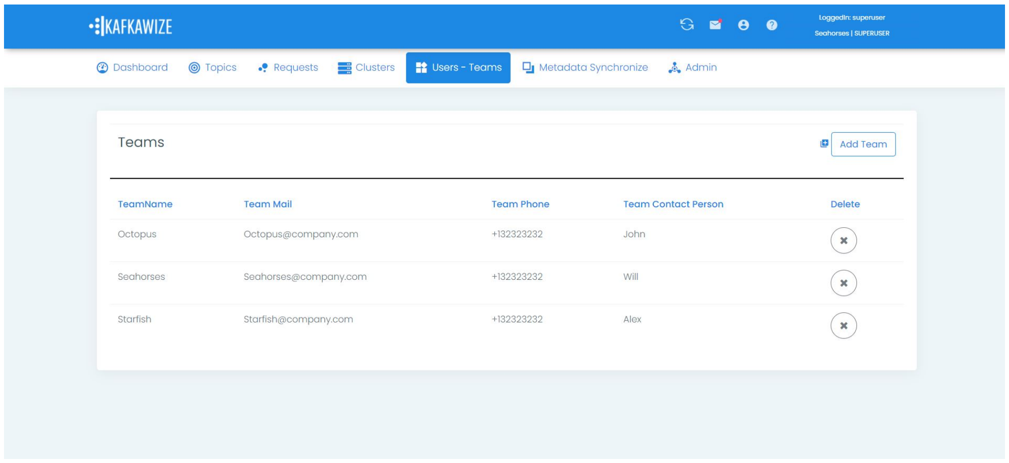
User Name: johnwoo	Full Name: superuser
Password: ••••	Repeat Password:
Email Id:	
Select a Team:	Select a Role:

4.11 Teams

All the teams are visible to any logged in user. A new team can be added, or a team can be deleted by only Super users.

4.11.1 View Teams

From this page all the users can be seen.

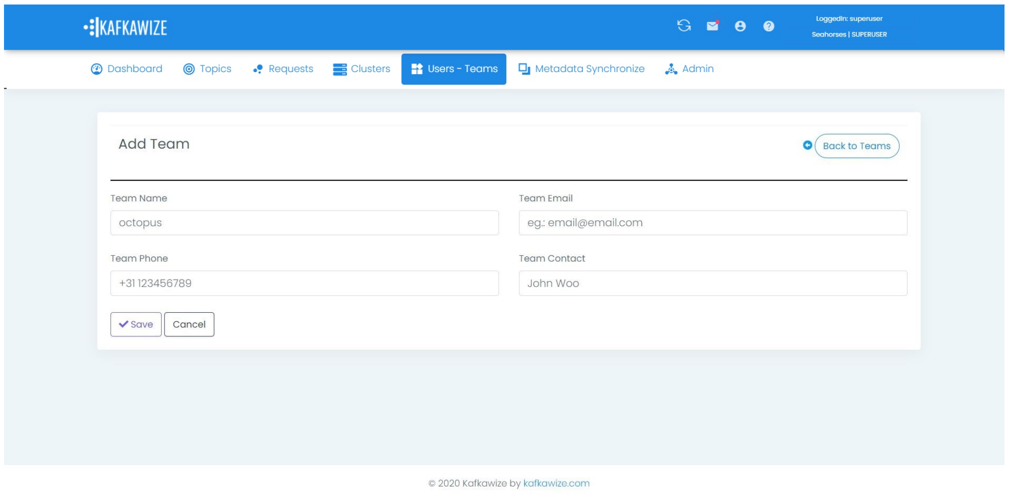


Teams

TeamName	Team Mail	Team Phone	Team Contact Person	Delete
Octopus	Octopus@company.com	+132323232	John	✖
Seahorses	Seahorses@company.com	+132323232	Will	✖
Starfish	Starfish@company.com	+132323232	Alex	✖

4.11.2 Add Team

With this form a new team can be added by a SuperUser.



Add Team

Team Name: octopus	Team Email: eg.: email@email.com
Team Phone: +31 123456789	Team Contact: John Woo

4.12 Admin

4.12.1 Server Config

All the Server configuration including custom properties (application properties can be viewed)

4.13 Profile

All Users can view their profile with the button available in the top right corner of the portal.

4.13.1 Change Password

All the logged in users can change the password, however it will be effective only after the UI Api Application is restarted. Enhancement will be provided on this soon.

4.14 Logout

Users can logout after clicking on the logout button on the top right corner. Session will be killed after this action.

4.15 Promote Topics

If a topic exists in Dev environment, it can be requested for promotion to the next higher environment (TST). If topic exists in Dev and Tst environments, promotion can be requested for Acceptance environment, with the same topic name. This way it restricts users to create topics in adhoc way and at the same time maintains data integrity. This feature provides an easy way to request for topics in higher environments, keeping partitions, replication factor parameters specific to those environments. This deletion request can be triggered only by Topic owner teams. Please note, it follows the same request/approval process.

4.16 Delete Topics

When a topic is no more needed,it is best to get it deleted. So we save on the partitions loaded by the cluster. This feature makes sure a topic can only be deleted when there are no subscriptions for the topic. Hence, no Producers and Consumers are affected by deleting. This deletion request can be triggered only by Topic owner teams. Please note, it follows the same request/approval process.

4.17 Delete Acls

It is possible to delete particular Producer or Consumer subscriptions of a topic with this feature. This deletion request can be triggered by that Acl owner teams. No other teams can request for other's acl deletion. So all the subscriptions are secure this way.

4.18 Ldap Authentication/Active Directory

Users can be authenticated against Ldap/Active Directory while users login to Kafkawize.

If you do not want to create/use a different password for Kafkawize users, rather authenticate from an existing Ldap server of your organization, it is easy to integrate now. Application properties have properties like baseDN, userDNPattern, passwordAttributes etc to configure your Ldap server and connect. The below property enables ldap authentication:

4.19 Cluster Connect Protocols

kafkawizeclusterapi can connect to Kafka cluster over the below protocols.

4.19.1 PLAINTEXT

Kafkawizeclusterapi connects can connect to Kafka clusters over PLAINTEXT protocol When a cluster is being created in Kafkawize, this protocol can be selected.

4.19.2 SSL

Kafkawizeclusterapi connects can connect to Kafka clusters over SSL protocol When a cluster is being created in Kafkawize, this protocol can be selected. Make sure SSL parameters are configured in application.properties in Kafkawizeclusterapi. SSL Parameters like keystore, truststore, passwords, etc.

4.19.3 SASL_PLAIN

Kafkawizeclusterapi connects can connect to Kafka clusters over SASL_PlAIN protocol When a cluster is being created in Kafkawize, this protocol can be selected. Make sure SSL parameters are configured in application.properties in Kafkawizeclusterapi.

SASL stands for Simple Authentication and Security Layer. Kafka Jaas configuration has to be configured on the Kafka cluster with username/passwords. Kafkawizeclusterapi application.properties would be configured with SASL Jaas config too like below:

```
org.apache.kafka.common.security.plain.PlainLoginModule required username='admin'␣
↪password='admin-secret';
```

4.19.4 SASL_SSL

Kafkawizeclusterapi connects can connect to Kafka clusters over SASL_SSL protocol When a cluster is being created in Kafkawize, this protocol can be selected. Make sure SSL parameters are configured in application.properties in Kafkawizeclusterapi.

SASL stands for Simple Authentication and Security Layer. Kafka Jaas configuration has to be configured on the Kafka cluster with username/passwords. Kafkawizeclusterapi application.properties would be configured with SASL Jaas config too like below

It is possible to configure PlainLoginModule or KerberosLoginModule:

```
com.sun.security.auth.module.Krb5LoginModule required \
        useKeyTab=true \
        storeKey=true \
        keyTab="/etc/security/keytabs/kafka_client.keytab" \
        principal="kafkaclient1@EXAMPLE.COM";
```

SSL parameters should also be configured, when ssl encryption is enabled.

SASL mechanism can be PLAIN or GSSAPI(Kerberos)

4.20 Synchronize Metadata

A situation where Kafka cluster already exists and would like to adapt Kafkawize in your organization, all the topics and acls need to have their teams/owners.

This feature is possible with Synchronization of Topics or Acls.

4.20.1 Topics

After a environment is selected, topics are displayed, and a team can be assigned to it. And this action, team becomes the Owner team.

_static/images/SynchronizeTopics.JPG

It is required to synchronize the topic team first with Base sync cluster first. Base sync cluster can be configured with property custom.syncdata.cluster in application properties.

4.20.2 Acls

After a environment is selected, Producer and Consumer Acls are displayed, and a team can be assigned to it. After this action, team becomes the Owner of that subscription (producer or consumer).

_static/images/SynchronizeAcls.JPG

4.21 Restore Config

Configuration of topics and acls from metastore can be restored back on any selected Kafka cluster.

4.22 Multi Tenancy

Each tenant can manage their topics with their own teams in isolation. Every tenant can have their own set of Kafka environments, and users of one tenant cannot view/access topics, acls or any from other tenants. It provides an isolation avoiding any security breach.

4.23 Analytics

Several charts are introduced to give an overview of Clusters and usages by teams. Topics, Acls and Partitions per team Topics, Acls and Partitions per environments Activity log per team and per environments

4.24 Configurable Roles & Permissions

Any new roles can be added and associate different kind of permissions Permissions A whole bunch of permissions can be assigned to roles from User interface, making it very flexible. With immediate effect, users will be able to see the changes. Ex: A permission to request topics, or approve or add users, environments, clusters etc.

4.25 User Registration

New users can register from the home page, and request is forwarded to Super user. After the approval, user is added to the system.

4.26 Email Notifications

For every request and approval, through out the Kafkawize application, emails are sent out to approvers, and after approvals, notify the requesters. When a new user is added, or password changes, etc, notifications are enabled.

4.27 Kafka connectors

Create and approve Kafka connectors

CHAPTER 5

Functionalities

- Create Topics on Kafka Clusters Automated (Request and Approve)
- Create Acls Automated (Request and Approve)
- Register Avro schemas Automated (Request and Approve)
- Search Topics with a User Interface
- View Acls of a topic
- View Producer and Consumer Teams of a topic
- Users, Teams and Roles based authorizations on features
- View server configuration (application properties)
- Audit of activities performed by users
- Login and Logout

CHAPTER 6

On Docker/Azure Cloud

Kafkawize can be run with Docker from version 5.0.3 You can download the docker images from https://hub.docker.com/u/kafkawize

6.1 Step 1 (Docker installation)

Make sure docker is installed on your system.

6.2 Step 2 (Docker-compose Kafkawize - uiapi)

Create a docker compose file(kafkawize_docker_compose_uiapi.yml) like below which contains images, properties of Kafkawize UserInterface Api applications.

For Azure, make sure images are correctly defined.

```
---
version: '2'
services:
  uiapi:
    image: kafkawize/kw_uiapi:5.0.3
    hostname: localhost
    environment:
      KAFKAWIZE_DB_STORETYPE: rdbms
      # Authentication type. Possible values : db
      KAFKAWIZE_LOGIN_AUTHENTICATION_TYPE: db
      KAFKAWIZE_DBSCRIPTS_CREATE_TABLES: "true"
      KAFKAWIZE_DBSCRIPTS_INSERT_BASICDATA: "true"

      # Other settings
      KAFKAWIZE_ORG_NAME: MyOrganization
      KAFKAWIZE_VERSION: 5.0.3
```

(continues on next page)

(continued from previous page)

```
      # Database settings
      SPRING_DATASOURCE_URL: jdbc:h2:file:./kafkawizedbos;DB_CLOSE_ON_EXIT=FALSE;DB_
↪CLOSE_DELAY=-1;MODE=MySQL;DATABASE_TO_LOWER=TRUE;
      SPRING_DATASOURCE_USERNAME: kafkauser
      SPRING_DATASOURCE_PASSWORD: kafkauser123
      SPRING_DATASOURCE_DRIVER_CLASS: org.h2.Driver
      SPRING_JPA_PROPERTIES_HIBERNATE_DIALECT: org.hibernate.dialect.H2Dialect
      SPRING_JPA_HIBERNATE_SHOW_SQL: "false"
      SPRING_JPA_HIBERNATE_GENERATE-DDL: "false"

      # Logging settings
      LOGGING_LEVEL_ORG_HIBERNATE_SQL: info
      LOGGING_LEVEL_ROOT: info
      LOGGING_FILE: kafkawize_uiapi.log
    ports:
      -   "9097:9097"
    extra_hosts:
      - "moby:127.0.0.1"
    network_mode: "host"
```

6.3 Step 3 (Docker-compose Kafkawize - clusterapi)

Create a docker compose file(kafkawize_docker_compose_clusterapi.yml) like below which contains images, properties of Kafkawize ClusterApi Api applications.

```
---
version: '2'
services:
  clusterapi:
    image: kafkawize/kw_clusterapi:5.0.3
    environment:
      LOGGING_FILE: kw_clusterapi.log
    ports:
      -   "9343:9343"
    extra_hosts:
      - "moby:127.0.0.1"
    network_mode: "host"
```

6.4 Step 4 (Start Kafkawize)

```
docker-compose -f .\kafkawize_docker_compose_uiapi.yml up
docker-compose -f .\kafkawize_docker_compose_clusterapi.yml up
```

6.5 Step 5 (Verify processes)

Verify docker processes

```
docker ps
```

6.6 Step 6 (Access Kafkawize)

Access Kafkawize from the below url:

```
http://<dockerhost>:9097/kafkawize
```

6.6.1 Credentials

Default Credentials available to access Kafkawize:

```
superadmin/kwsuperadmin123$$
```

6.6.2 Docker shell

You can login into the docker container shell with the below command:

```
docker exec -ti <docker_container_id> /bin/bash
```

CHAPTER 7

Customize

Kafkawize can be customized according to your requirement very easily.

As the code is available in git, you can make the changes locally in Front end (AngularJS, HTML, CSS ..), or backend (Java..) and customize in the way you need.

CHAPTER 8

Trouble shooting

- Check for logs in Cluster Api, UI Api
- Check if user has the right access
- Check if tables and data in the metastore can be accessible(grants,permissions) by Kafkawize
- Check if all tables are properly created in Rdbms

CHAPTER 9

Release Notes

- The below changes are part of release 5.0.3
- **ACLs can be added based on Usernames (plain text principles). Ex** [alice, john] On kafka cluster, Producer acls reflects like below for user 'alice' for a topic. (principal=User:alice, host=*, operation=WRITE, permissionType=ALLOW) (principal=User:alice, host=*, operation=WRITE, permissionType=ALLOW)
- Introducing Liquibase for all database migrations
- Recaptcha validation can be disabled if running in saas mode

---

Kafkawize 5.0.0

Topics (approval): Create, Update, Delete, Promote Acls (approval): Create Connectors (approval): Create Avro Schemas (approval): Create Topic Overview :

Topic Config Promote Literal and Prefixed subscriptions Topic documentation Consumer offsets/ lag View topic contents

View created, completed, declined, all Topic requests View created, completed, declined, all Acl requests View created, completed, declined, all Connector requests View created, completed, declined, all Avro schema requests

Synchronization from and to kafka clusters Reconciliation and email notifications on differences between Kafkawize and Clusters Restore configuration (topics, acls)

Login : Active directory integration Single Sign-on (OAuth2)

Clusters and Environments Clusters can be created connecting to Kafka clusters. (Cluster Management Api should be configured) Environments are wrappers over clusters, enforcing flexible configs like prefix, suffix etc

Users, Teams & Authorizations Configurable users, teams More than 35 permissions Configurable roles (Roles can be pulled from AD for authorization)

Topic naming conventions Enforce prefix and suffixes per environment

Excel report (for your team and all teams, depending on the role) Topics per cluster (for teams) Partitions per cluster Overall topics in all clusters Acls per cluster (for teams) Producer Acls (for teams) Consumer Acls (for teams) Consumer groups of all environments Requests per day

Analytics View charts of topics, partitions, acls, requests

Multi tenancy Each tenant can manage their topics with their own teams in isolation. Every tenant can have their own set of Kafka environments, and users of one tenant cannot view/access topics, acls or any from other tenants. It provides an isolation avoiding any security breach.

Kafka Connectivity PLAINTEXT, SSL, SASL

Audit All topic, acl, schema and connector requests

Email notifications when requests are created, approved, declined users are created, approved

Help Wizard to setup Kafkawize

Kafkawize 4.5.1

Kafkawize 4.5.1 is a minor release having minor improvements

1. Updated datasource to Hikari
2. Connection parameters now have few caching related parameters for prepared statements.
3. Few code enhancements
4. Removed Cassandra datasource and related dependencies
5. Upgrade of kafka client version in cluster api to 2.6.1

Kafkawize 4.5

Kafkawize 4.5 has several changes including the below.

1. **New dashboard displaying the following details**
    a. My team topics
    b. My producer topics
    c. My consumer topics
    d. My team members
    e. Chart with Activity log overview of your team
    f. Chart with topics on different clusters of your team
    g. Pending Topics requests
    h. Available Clusters and their status
2. Show Producer and Consumer topics when requested from Dashboard
3. Enabled deep linking of urls
4. Color layout similar to pro version, and updated logos
5. Comma separated bootstrap servers for Kafka and schema registry clusters. (Removed explicit port field.)
6. Pagination on users page
7. Users page for a team
8. Bug fix for displaying team on Request acl page
9. Improvement on processing concurrent requests
10. Code improvisation : removed commented blocks and unused FE calls

11. Added Spring shutdown hooks and removed system exits.
12. Removed support for Cassandra metastore option.
13. Support for File based metastore (File - H2 database)
14. Upgraded Kafka client to 2.5.1
15. Database changes, following better naming conventions.

Kafkawize 4.4

1. Changes include improved User interface and few bug fixes.
2. Metadata Synchronize option has been removed
3. SSL connectivity to Kafka cluster has been removed
4. Dashboard updated to show logged in Username, Team and Role
5. Users can now only 4 environments DEV, TST, ACC and PRD. Hierarchy is defined in properties file.
6. New model for UserInfo class is introduced to fix a password bug
7. Password is not stored as encrypted text
8. Validations bug in acls and topics requests
9. Connect to Kafka clusters during start of Kafkawize

Kafkawize 4.3

Changes include improved User interface and few bug fixes.

Kafkawize 4.2

Changes include 1. Critical bug fix - concurrent user access 1. Ability to have environments DEV, TST, ACC and PRD 2. Ability to request for topics in DTAP environments 3. Ability to view topic overview and subscriptions in one page 4. Ability to view topic partitions and replication factor of all environments in topic overview page 5. Ability to view topics and their existence in all environments 6. Updated dashboard to view your team topics

---

Kafkawize 4.1

Changes include 1. New Bootstrap 4 User interface with new appealing look and feel 2. New UI/UX - for great user experience 3. Few bug fixes 1. Critical bug fix - concurrent user access

---

Kafkawize 4.0

Changes include 1. About 320 Unit tests. Above 85% code coverage. 2. Integration tests for both stores Cassandra and Rdbms, with EmbeddedCassandra and Embedded H2 sql database 3. New UI for viewing topics 4. New UI for viewing acls of topic 5. New UI for approving topics 6. New UI for approving acls 7. New UI for login screen 8. New UI for Dashboard, showing cluster api status and kafka cluster statuses 9. Added License key validation 10. Bug fixes and code enhancements

There are several other changes and upgraded dependencies which improved the code quality and efficiency. 1. New Bootstrap 4 User interface with new appealing look and feel 2. New UX - for great user experience 3. Few bug fixes

---

Kafkawize 3.5

Changes include 1. New page (Admin-ServerConfig) to display server configuration - application properties 2. Default replication factor, default partitions and default max partitions can be configured from Clusters page. 3. Couple of minor bug fixes 1. About 320 Unit tests. Above 85% code coverage. 2. Integration tests for both stores Cassandra and Rdbms, with EmbeddedCassandra and Embedded H2 sql database 3. New UI for viewing topics 4. New UI for viewing acls of topic 5. New UI for approving topics 6. New UI for approving acls 7. New UI for login screen 8. New UI for Dashboard, showing cluster api status and kafka cluster statuses 9. Added License key validation 10. Bug fixes and code enhancements

There are several other changes and upgraded dependencies which improved the code quality and efficiency.

Kafkawize 3.4

Changes include

1. Decline Topic requests
2. Decline Acl requests

3. Bug fix in creating topic request 1. New page (Admin-ServerConfig) to display server configuration - application properties 2. Default replication factor, default partitions and default max partitions can be configured from Clusters page. 3. Couple of minor bug fixes

---

Kafkawize 3.3

Changes include search features in almost all screens, bug fixes and code improvements. Changes include 1. Decline Topic requests 2. Decline Acl requests 3. Bug fix in creating topic request

---

Kafkawize 3.2

Changes include search features in almost all screens, bug fixes and code improvements.

Kafkawize 3.1

New features: 1. Support for RDBMS store like MySql to store meta information. 1.0 only supports Apache Cassandra. It is one of the important feature which will support many customers who already have an SQL based solution. Changing property db.storetype=rdbms/cassandra will make the difference.

Bug fixes:

There are few bugs which are fixed in Topic requests, acls and schema registry modules.

Changes include search features in almost all screens, bug fixes and code improvements.

Kafkawize 2.0

Kafkawize is a Kafka Topic management tool (A Web application) which automates the process of creating and browsing Kafka components, by introducing roles/authorizations to users of various teams of an organization

Changes include new feature Rdbms support for metastore, package restructuring, jpa/hibernate implementation, improved code quality and bug fixes.

New features: 1. Support for RDBMS store like MySql to store meta information. 1.0 only supports Apache Cassandra. It is one of the important feature which will support many customers who already have an SQL based solution. Changing property db.storetype=rdbms/cassandra will make the difference.

Bug fixes:

There are few bugs which are fixed in Topic requests, acls and schema registry modules.

CHAPTER 10

Roadmap

Improve User interface.

CHAPTER 11

Contact Me

If you have any issues with kafkawize, please raise it in:

```
https://github.com/muralibasani/kafkawize/issues
```

You can also send me an email at info@kafkawize.com or kafkawize@gmail.com

You can also use the form https://kafkawize.com/contact-me/

Muralidhar Basani

Linkedin : https://www.linkedin.com/in/muralibasani/

CHAPTER 12

Submit a Review

Please do not forget to star my project

```
https://github.com/muralibasani/kafkawize/stargazers
```

Thank you.

CHAPTER 13

Indices and tables

- genindex
- modindex
- search