

Kafkawize

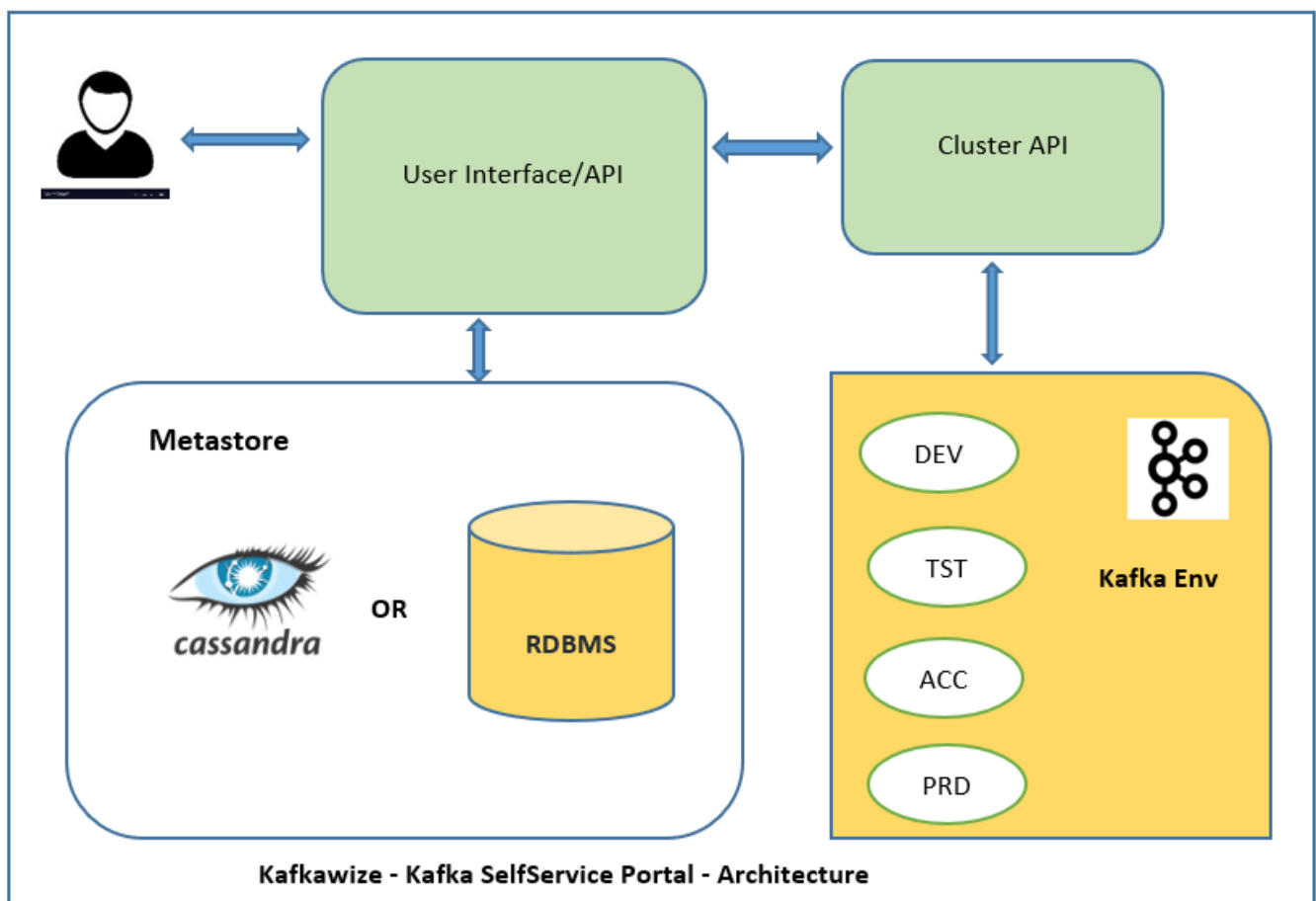
M medium.com/@murali.basani/kafkawize-1908b223c082

A Self service Apache Kafka Topic Management portal/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.

Github : <https://github.com/muralibasani/kafkawize>

Latest release : kafkawize 2.0 includes support for Rdbms as metastore

Technical Architecture



Application details :

Kafkawize contains two main APIs. (User Interface API and Cluster management API)

User Interface API directly communicates between users 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.

Apache Cassandra OR Rdbms(MySql/Oracle/..) datastore stores all users, teams, topics, request and execution data from all the users., and to maintain source of truth.

Spring Security, Spring Boot frameworks, JPA/Hibernate are used to develop this application.

Functionalities :

(Broadly divided into two categories based on user roles.)

ROLE : USER can request for creation of kafka components, and browse kafka components.

ROLE : ADMIN can approve and execute the requests of users to create kafka components.

ROLE : SUPERUSER can synchronize topics meta information with Cassandra Datastore from Kafka Brokers (Source of Truth.)

Browse: (ROLE : USER, ADMIN, SUPERUSER)

All users can Browse Topics

All users can Browse Acls

All users can view the producers and consumers of all topics.

Requests:(ROLE : USER)

Users can request for Kafka Topics

Users can request for Kafka Acl

Users can request for Schemas to be registered on Confluent Schema registry.

Audit :

Users can view all the requests from his team.

Users can view all the activities from his team or as a super user from all teams

Environments:(ROLE : USER)

Users can view the available environments

Approve—Execute :(ROLE : ADMIN)

Users can approve requests for creating Kafka Topics

Users can approve requests for creating Kafka Acls

Users can approve requests for uploading schemas on topics

Users :(ROLE : ADMIN)

Users can view all user details

Users can add new users

Teams :(ROLE : ADMIN, SUPERUSER)

Users can view all team details

Users can add new team

Environments:(ROLE : SUPERUSER)

Users can add a new environment environments

Synchronize Metadata :(ROLE : SUPERUSER)

Users can synchronize topic information from Brokers with Cassandra OR Rdbms(MySql/Oracle/..) datastore. (Update team info.)

Users can synchronize acls information from Brokers with Cassandra OR Rdbms(MySql/Oracle/..) datastore. (Update team info.)

My Profile :(ROLE : USER, ADMIN)

Users can view their profile.

Change Password :(ROLE : USER, ADMIN)

Users can change their passwords.

Logout :(ROLE : USER, ADMIN)

Users can logout.

How to Run the application

KafkaWize needs the following applications to be up and running.

1. Spring boot application KafkaWize

<https://github.com/muralibasani/kafkawize>

2. Spring boot application KafkaWize ClusterApi
<https://github.com/muralibasani/kafkawizeclusterapi>
3. Apache Cassandra OR Rdbms(MySql/Oracle/..)

Steps to run with **Cassandra** as Metastore:

1. Install Apache Cassandra
2. Setup project KafkawizeClusterApi and update server.port if necessary in application properties
3. Start KafkaClusterApi
4. Setup project KafkaWize, and configure Cassandra running host, Cluster api host, in application properties
5. Set db.storetype=cassandra in application.properties
6. Start KafkaWize
7. Cassandra db setup will be done on the startup of the application. We do not have to create manually.

Steps to run with **Rdbms** as Metastore:

1. Install an Rdbms database (Mysql or Oracle or ..)
2. Setup project KafkawizeClusterApi and update server.port if necessary in application properties
3. Start KafkaClusterApi
4. Setup project KafkaWize, and configure Cassandra running host, Cluster api host, in application properties
5. Set db.storetype=rdbms and few other datasource properties in application.properties
6. Start KafkaWize
7. Run the ddl and insert scripts available in src/main/resources/scripts/base/rdbms.

By default KafkaWize runs on port 9097. Access it by <http://localhost:9097>

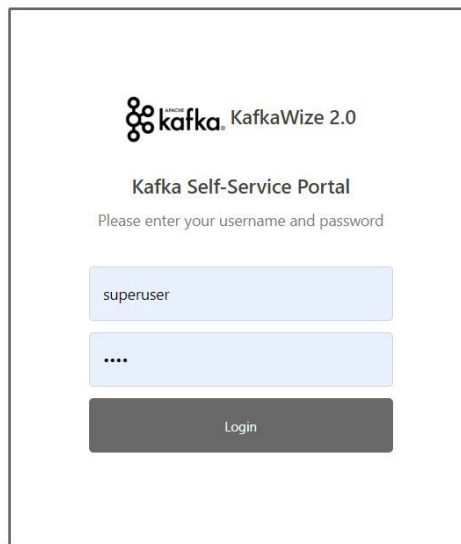
Default Teams Team1 Team2 Team3

Default Users(with pwds)

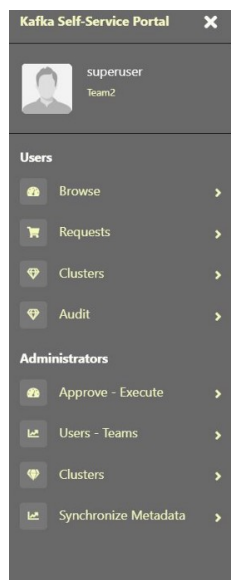
uiuser1/user from Team1
uiuser2/user from Team2
uiuser4/user from Team1 Admin
uiuser5/user from Team2 Admin
superuser/user from Team2 Superuser

Screenshots :

localhost:9097/login



The login page for KafkaWize 2.0. It features the Apache Kafka logo and the text 'KafkaWize 2.0'. Below this is the title 'Kafka Self-Service Portal' and a prompt 'Please enter your username and password'. There are two input fields: the first contains 'superuser' and the second contains four dots. A 'Login' button is at the bottom.



The sidebar of the Kafka Self-Service Portal. It shows the user 'superuser' from 'Team2'. Under the 'Users' section, there are links for 'Browse', 'Requests', 'Clusters', and 'Audit'. Under the 'Administrators' section, there are links for 'Approve - Execute', 'Users - Teams', 'Clusters', and 'Synchronize Metadata'.

Welcome to KafkaWize 2.0 !!

Kafka Self Service Portal

 **kafka**
KafkaWize - My ABC Company

About the Application

Objective :

Automate the process of creating and browsing Kafka components, by introducing roles, teams and users.

Architecture:

There are two main apis.

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.

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

User Interface API directly communicates between users and Cluster API.

Apache Cassandra/Rdbms(Mysql/Oracle..) datastore is used to store all users, teams, topicRequests, request and execution information from all the users., and to maintain source of truth.

Spring Security, Spring Boot frameworks are used to develop this application.

- Browse Topics
- Browse Acls
- Request Topics
- Request Acls
- Request Schemas
- View Envs
- Execute Topics
- Execute Acls
- Execute Schemas

Kafka Self-Service Portal

uiuser4

Online

Users

Browse

Requests

Environments

Administrators

Approve - Execute

Users - Teams

Environments

Add Environment

Synchronize Metadata

Kafka Self-Service Portal

uiuser4

Online

Users

Browse

Requests

Environments

Administrators

Approve - Execute

Users - Teams

View Users

Add Users

View Teams

Add Teams

Environments

Synchronize Metadata

Kafka Self-Service Portal

uiuser4

Online

Users

Browse

Requests

Environments

Administrators

Approve - Execute

Users - Teams

View Users

Add Users

View Teams

Add Teams

Environments

Synchronize Metadata

Add New Environment

kafka

KafkaWize - My ABC Company

Existing Environments

New Environment name

Host

Port

Protocol

Environment Type

ACC_KAFKA

kafka-acc-env

9092

plain

kafka

Add

Add New Team

kafka

KafkaWize - My ABC Company

Existing Teams

New Team Name

Team Mail

Team Phone

Team Contact Person

Team Application

Team2

Team4

team4@team.com

+3142342432

Kafka Wizer

My Kafka App

Submit New Team

Add New User

KafkaWize - I

User Name

Full Name

Password

Team

Role

kafkawizeuser10

Kafka Wizer

.....

Team2

USER

Submit New User

6/10

Kafka Self-Service Portal

uiuser4

Online

Users

Browse

Requests

Environments

Administrators

Approve - Execute

Topics

Acls

Schemas

Users - Teams

Environments

Synchronize Metadata

Kafka Self-Service Portal

uiuser4

Online

Users

Browse

Requests

Environments

Administrators

Approve - Execute

Topics

Acls

Schemas

Users - Teams

Environments

Synchronize Metadata

Kafka Self-Service Portal

uiuser1

Online

Users

Browse

Topics

Acls

Producers-Consumers

Requests

Environments

Administrators


Approve - Execute

Users - Teams

Environments

Synchronize Metadata

Execute Acls


kafka

KafkaWize - My ABC Company

	TopicName	AcType	ConsumerGroup	Environment	Requestor	Ac_IP	Ac_SSL	Application	Requesting Team	Remarks	Status	RequestedTime	ApprovingTime
<input checked="" type="radio"/>	demotopic2	Consumer	newgroup	DEV	uiuser1	127.0.0.1		NewApp	Team1	Remarks	created	Sat Dec 01 21:40:52 CET 2018	null

Approve Acl Requests

Execute Topics


kafka

KafkaWize - My ABC Company

	TopicName	TopicPartitions	TopicReplicationFactor	Environment	Requestor	Ac_IP	Ac_SSL	Application	Team	Remarks	Status	RequestedTime	Approving
<input type="radio"/>	demotopic4	2	1	DEV	uiuser1	127.0.0.1		App	Team1	Pls approve	created	Sat Dec 01 21:44:03 CET 2018	null

Approve Topic Request

Browse Acls

kafka

KafkaWize - My ABC Company

Environment

DEV

Show

Pages: [1](#) [2](#)

Sequence	TopicName	ConsumerGroup	Hosts	Principle	AcType	Owned by
1	testtopic		127.0.0.1	User:*	Producer	Team1
2	demotopic1		127.0.0.1	User:*	Producer	Team3
3		testconsumergroup	127.0.0.1	User:*	Consumer	Team2
4	testtopic13		127.0.0.1	User:*	Consumer	Team2
5	testtopic4		127.0.0.1	User:*	Producer	
6	testtopic1		127.0.0.1	User:*	Producer	
7	testtopic		localhost	User:*	Producer	
8	testtopic2		127.0.0.1	User:*	Producer	
9	testtopic12		127.0.0.1	User:*	Producer	
10	testtopic10		127.0.0.1	User:*	Producer	

7/10

Kafka Self-Service Portal

uiuser1

Online

Users

Browse

Topics

ACLs

Producers-Consumers

Requests

Environments

Administrators

Approve - Execute

Users - Teams

Environments

Synchronize Metadata

Browse Topics

Environment

DEV

Show

Pages: [1]

Sequence	TopicName	Partitions	Replicas	Owned by
1	demotopic1	1	1	Team1
2	demotopic2	1	1	Team1
3	testtopic10	1	1	Team3
4	testtopic11	1	1	Team2
5	testtopic12	1	1	Team3
6	testtopic13	1	1	Team1
7	testtopic1	1	1	Team3
8	testtopic2	1	1	Team3
9	testtopic4	1	1	

Kafka Self-Service Portal

uiuser4

Online

Users

Browse

Requests

Environments

Kafka

Schema Registry

Administrators

Approve - Execute

Users - Teams

Environments

Synchronize Metadata

Kafka Environments

Environment	Host	Port	Protocol	Type
DEV_NEW	127.0.0.1	9095	plain	kafka
DEV	localhost	9092	PLAIN	kafka
TST_SSL	tesstserver	9093	SSL	kafka
TST	tesstserver	9092	PLAIN	kafka

Schema Registry Environments

Environment	Host	Port	Protocol	Type
DEV_SCHEMAREGISTRY	localhost	8081	PLAIN	schemaregistry

Kafka Self-Service Portal

uiuser1

Online

Users

Browse

Requests

Request Topic

Request ACL

Request Avro Schema

My Requests

Environments

Administrators

Approve - Execute

Users - Teams

Environments

Synchronize Metadata

My Requests

Topic Requests

	TopicName	Environment	Requestor	ACL_IP	ACL_SSL	Application	Team	Remarks	Status	RequestedTime	ApprovingTime
<input type="radio"/>	demotopic1	DEV	uiuser1	127.0.0.1		App1	Team1	Please create the topic	approved	Sun Nov 18 19:40:32 CET 2018	Sun Nov 18 19:43:12 CET 2018
<input type="radio"/>	demotopic2	DEV	uiuser1	127.0.0.1		App	Team1	Rem	approved	Sat Dec 01 18:15:17 CET 2018	Sat Dec 01 18:15:33 CET 2018

Delete Topic Request

ACL Requests

	TopicName	ACLType	ConsumerGroup	Environment	Requestor	ACL_IP	ACL_SSL	Application	Topic Owner Team	Remarks	Status	RequestedTime	ApprovingTime
<input type="radio"/>	demotopic2	Consumer	newgroup	DEV	uiuser1	127.0.0.1		NewApp	Team1	Remarks	created	Sat Dec 01 21:40:52 CET 2018	null

Delete ACL Request

Kafka Self-Service Portal

uluser1

Online

Users

Browse

Topics

ACLs

Producers-Consumers

Requests

Environments

Administrators

Approve - Execute

Users - Teams

Environments

Synchronize Metadata

Producers-Consumers

Environment

DEV

Show

Topic	Producer Teams	Consumer Teams
testtopic1	Team3	
demotopic1	Team1 Team3	Team3
testtopic13	Team1 Team2	Team2 Team1 Team3
testtopic5	Team2	
testtopic10	Team3	

Kafka Self-Service Portal

uluser1

Online

Users

Browse

Requests

Request Topic

Request ACL

Request Avro Schema

My Requests

Environments

Administrators

Approve - Execute

Users - Teams

Environments

Synchronize Metadata

Request For ACL

Select Environment

DEV

Topic Names

demotopic2

Type

Consumer

Consumer group

newconsumergroup

ACLs IP

127.0.0.1

ACLs SSL

Ex:CN=cname,OU=IS,OU=OU,OU=Services,O=Org

Topic Owner Team

Team1

Team Application

TeamApp1

Remarks

Pls approve

Submit Acl Request

Kafka Self-Service Portal

uluser1

Online

Users

Browse

Requests

Request Topic

Request ACL

Request Avro Schema

My Requests

Environments

Administrators

Approve - Execute

Users - Teams

Environments

Synchronize Metadata

Request For Topics

Select Environment

DEV

Topic Names

demotopic3

Topic Partitions

2

ACLs IP

127.0.0.1

ACLs SSL

Ex:CN=cname,OU=IS,OU=OU,OU=Services,O=Org

Team

Team1

Team Application


App1

Remarks

Pls approve

Submit Topic Request

Kafka Self-Service Portal



superuser

Online

Users

Browse

Requests

Environments

Administrators

Approve - Execute

Users

Environments

Synchronize Metadata

Topics

Acls

Synchronize Acls with Broker

Environment

DEV


Show

Pages: 1 2

Sequence	Team	TopicName	ConsumerGroup	Hosts	Principle	AclType
21		testtopic11		127.0.0.1	User:*	Producer
22	Team3	demotopic1		127.0.0.1	User:*	Consumer
23	Team1	demotopic4		127.0.0.1	User:*	Producer
24	Team2		testgroup	127.0.0.1	User:*	Consumer
25		testtopic12		10.1.1.1	User:*	Producer

Synchronize

Kafka Self-Service Portal



superuser

Online

Users

Browse

Requests

Environments

Administrators

Approve - Execute

Users

Environments

Synchronize Metadata

Topics

Acls

Synchronize Topics with Broker

Environment

DEV

Show

Pages: 1

Sequence	TopicName	Team
1	demotopic1	Team1
2	demotopic2	Team1
3	demotopic4	Team1
4	testtopic10	Team3
5	testtopic11	Team2
6	testtopic12	Team3
7	testtopic13	Team1
8	testtopic1	Team3
9	testtopic2	Team3
10	testtopic4	
11	testtopic5	Team2

Github : <https://github.com/muralibasani/kafkawize>

Kafkawize is an opensource project. Please contribute/checkout to this github project.