

Deploy Strimzi in the Developer Sandbox

The developer sandbox does not include an AMQ-Streams / Strimzi operator, and as a user, you don't have permission to deploy operators. This guide helps you deploy an all-in-one Kafka server and ui for your development activities.

Access the developer sandbox

You can access a free-to-use Openshift environment. Follow the instructions in this article:

[How to access the Developer Sandbox for Red Hat OpenShift](#)

Deploy a Kafka server

Using the console

From Openshift's Developer view, follow these steps:

- 1) Developer → +Add → Import Yaml

Unset

```
---
kind: Deployment
apiVersion: apps/v1
metadata:
  name: zkless-kafka
spec:
  replicas: 1
  selector:
    matchLabels:
      app: zkless-kafka
  template:
    metadata:
      labels:
        app: zkless-kafka
    spec:
      containers:
        - resources:
```

```
1 kind: Deployment
2 apiVersion: apps/v1
3 metadata:
4   name: zkless-kafka
5   labels:
6     app: zkless-kafka
7   replicas: 1
8   selector:
9     matchLabels:
10      app: zkless-kafka
11   template:
12     metadata:
13       labels:
14         app: zkless-kafka
15     spec:
16       containers:
17         - resources:
18           limits:
19             cpu: 100m
20             memory: 512Mi
21           requests:
22             cpu: 100m
23             memory: 512Mi
24           - zkless-kafka
25           - zkless-kafka
26           - zkless-kafka
27           - zkless-kafka
28           - zkless-kafka
29           - zkless-kafka
30           - zkless-kafka
31           - zkless-kafka
32           - zkless-kafka
33           - zkless-kafka
34           - zkless-kafka
35           - zkless-kafka
36           - zkless-kafka
37           - zkless-kafka
38           - zkless-kafka
39           - zkless-kafka
40           - zkless-kafka
41           - zkless-kafka
42           - zkless-kafka
43           - zkless-kafka
44           - zkless-kafka
45           - zkless-kafka
46           - zkless-kafka
47           - zkless-kafka
48           - zkless-kafka
49           - zkless-kafka
50           - zkless-kafka
51           - zkless-kafka
52           - zkless-kafka
53           - zkless-kafka
54           - zkless-kafka
55           - zkless-kafka
56           - zkless-kafka
57           - zkless-kafka
58           - zkless-kafka
59           - zkless-kafka
60           - zkless-kafka
61           - zkless-kafka
62           - zkless-kafka
63           - zkless-kafka
64           - zkless-kafka
65           - zkless-kafka
66           - zkless-kafka
67           - zkless-kafka
68           - zkless-kafka
69           - zkless-kafka
70           - zkless-kafka
71           - zkless-kafka
72           - zkless-kafka
73           - zkless-kafka
74           - zkless-kafka
75           - zkless-kafka
76           - zkless-kafka
77           - zkless-kafka
78           - zkless-kafka
79           - zkless-kafka
80           - zkless-kafka
81           - zkless-kafka
82           - zkless-kafka
83           - zkless-kafka
84           - zkless-kafka
85           - zkless-kafka
86           - zkless-kafka
87           - zkless-kafka
88           - zkless-kafka
89           - zkless-kafka
90           - zkless-kafka
91           - zkless-kafka
92           - zkless-kafka
93           - zkless-kafka
94           - zkless-kafka
95           - zkless-kafka
96           - zkless-kafka
97           - zkless-kafka
98           - zkless-kafka
99           - zkless-kafka
100          - zkless-kafka
```

```


      limits:
        cpu: 250m
        memory: 512Mi
      terminationMessagePath: /dev/termination-log
      name: zkless-kafka
      command:
        - /bin/sh
        - '-c'
        - >-
          export CLUSTER_ID=$(bin/kafka-storage.sh random-uuid) &&
          bin/kafka-storage.sh format -t $CLUSTER_ID -c
          config/kraft/server.properties && bin/kafka-server-start.sh
          config/kraft/server.properties --override
          advertised.listeners=${KAFKA_ADVERTISED_LISTENERS}
      env:
        - name: LOG_DIR
          value: /tmp/logs
        - name: KAFKA_ADVERTISED_LISTENERS
          value: 'PLAINTEXT://zkless-kafka-bootstrap:9092'
      ports:
        - containerPort: 9092
          protocol: TCP
      imagePullPolicy: IfNotPresent
      terminationMessagePolicy: File
      image: 'registry.redhat.io/amq-streams/kafka-35-rhel8:2.5.0'
      restartPolicy: Always
      terminationGracePeriodSeconds: 30
      dnsPolicy: ClusterFirst
      securityContext: {}
      schedulerName: default-scheduler
    strategy:
      type: RollingUpdate
      rollingUpdate:
        maxUnavailable: 25%
        maxSurge: 25%
      revisionHistoryLimit: 10
      progressDeadlineSeconds: 600
  ---
  apiVersion: v1
  kind: Service
  metadata:
    name: zkless-kafka-bootstrap
  spec:
    ports:

```







```
- port: 9092
  protocol: TCP
  targetPort: 9092
selector:
  app: zkless-kafka
type: ClusterIP
```

2) Click Create

The screen will show:

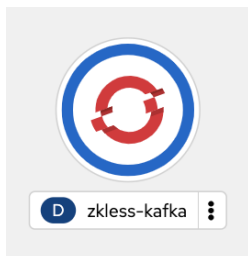


Resources successfully created

Name	Namespace	Creation status
 zkless-kafka	 bmesegue-dev	 Created
 zkless-kafka-bootstrap	 bmesegue-dev	 Created

[Import more YAML](#)

You should see the server created in your Topology view:



Deploy a Kafka UI

Create UI client configuration

Create a new ConfigMap with the following configuration:

Unset

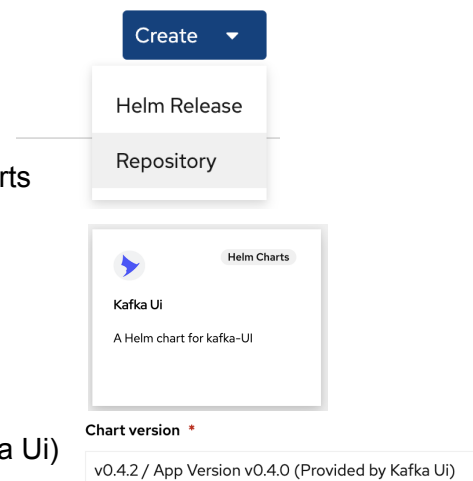
```
kind: ConfigMap
apiVersion: v1
metadata:
  name: kafka-ui-configmap
data:
  config.yml: |-
    kafka:
      clusters:
        - name: yaml
          bootstrapServers: zkless-kafka-bootstrap:9092
      auth:
        type: disabled
      management:
        health:
          ldap:
            enabled: false
```

```
1 kind: ConfigMap
2 apiVersion: v1
3 metadata:
4   name: kafka-ui-configmap
5 data:
6   config.yml: |-
7     kafka:
8       clusters:
9         - name: yaml
10           bootstrapServers: zkless-kafka-bootstrap:9092
11       auth:
12         type: disabled
13       management:
14         health:
15           ldap:
16             enabled: false
```

Using the console

From Openshift's Developer view, follow these steps:

- 3) Developer → Helm → Create → Repository
 - a) Name: kafka-ui
 - b) URL: <https://provectus.github.io/kafka-ui-charts>
- 4) Click Create
- 5) Developer → Helm → Create → Helm Release
 - a) Filter by Kafka
 - b) Choose the "Kafka UI" chart
- 6) Click Create
- 7) Select the latest Chart version:
 - 0.7.5 / App Version v0.7.1 (Provided by Kafka Ui)
- 8) Configure YAML
 - a) Delete the default YAML definition
 - b) Use instead:



Unset

```
yamlApplicationConfigConfigMap:  
  keyName: config.yml  
  name: kafka-ui-configmap
```

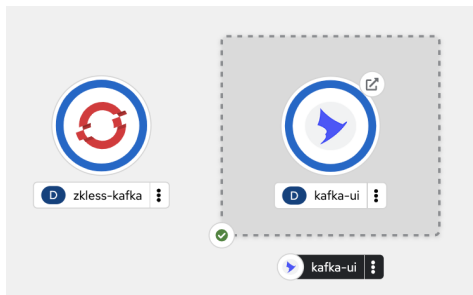
```
1  yamlApplicationConfigConfigMap:  
2    keyName: config.yml  
3    name: kafka-ui-configmap
```

9) Click Create

Create a Route

- 1) Administrator → Networking → Routes → Create
 - a) Name: **kafka-ui**
 - b) Service: **kafka-ui**
 - c) Target Port: **80** → **http (TCP)**
- 2) Click Create

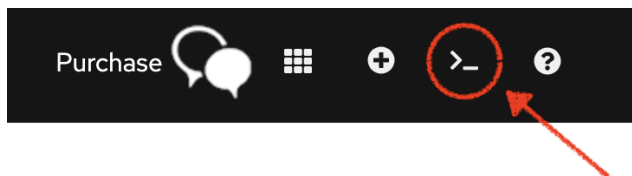
You should now see both the Kafka server and the Kafka UI from your Topology view:



Using CLI

Alternatively, you could use a terminal from where you can use the 'helm' command.

- 1) Open a terminal



If the ConfigMap containing the client configuration hasn't been created yet, follow these steps:

- 1) Create a ConfigMap (as described above)

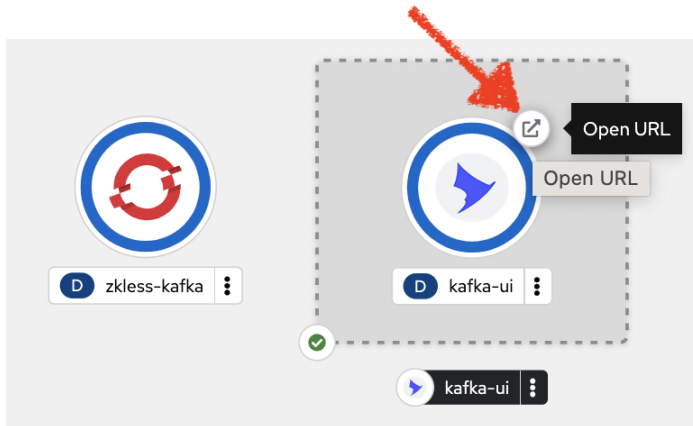
Install the Kafka UI by executing the commands below:

Unset

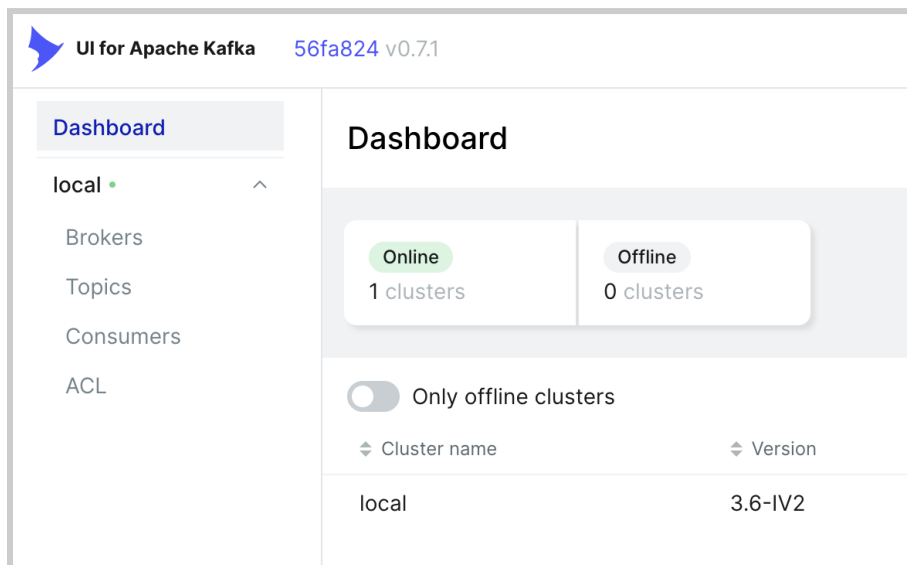
```
helm repo add kafka-ui https://provectus.github.io/kafka-ui-charts  
helm install --set  
yamlApplicationConfigConfigMap.name="kafka-ui-configmap",yamlApplicationConfigC  
onfigMap.keyName="config.yml" kafka-ui kafka-ui/kafka-ui
```

Open the Kafka UI

From your Topology view, click on the route to open the Web UI to manage your Kafka instance:

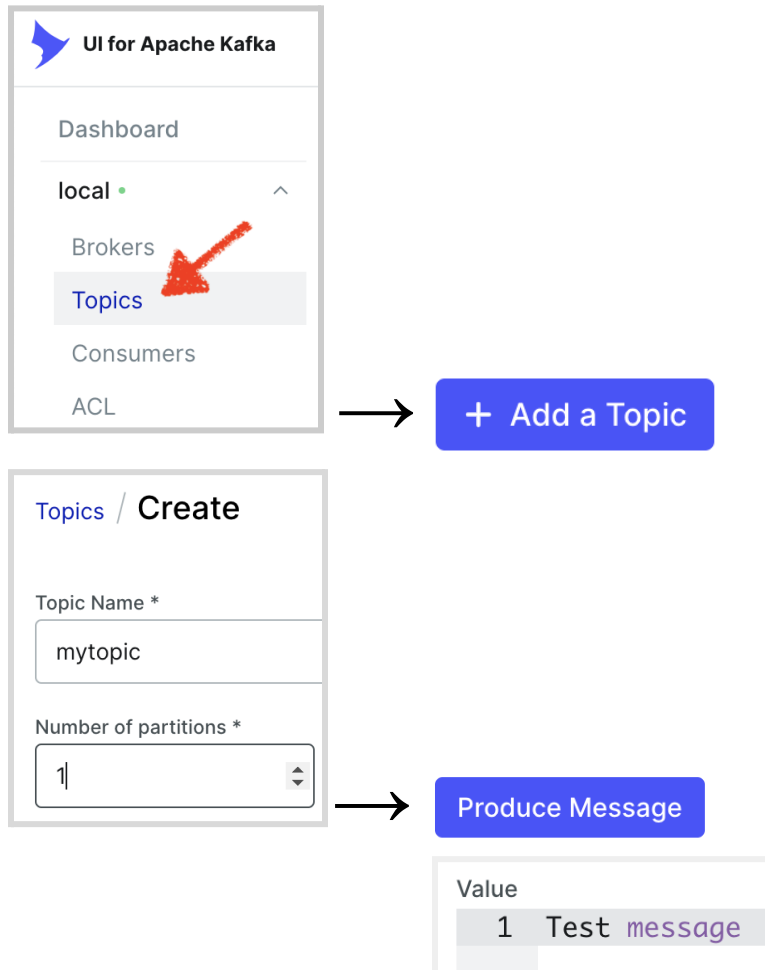


You should see the Dashboard as follows:



Test Kafka

From the Web UI, you can create a topic, and send a test message to validate all works as expected:



You should see your message listed in your browser:

The screenshot shows the 'Topics / mytopic' page with the 'Messages' tab selected. The table displays a single message with the following details:

	Offset	Partition	Timestamp	Key	Value
+	0	0	10/20/2023, 17:23:08		Test message