Настройка Kafka для работы с wiSLA (и с wiProbe Aggregator)

1. Установка Docker

Перед началом работы необходимо убедиться, что Docker установлен и работает.

Для систем без выхода в интернет

Скачайте архив Docker.zip и распакуйте из него нужные версии deb-пакетов.

Далее вам необходимо загрузить на сервер следующие пакеты:

- containerd.io;
- docker-ce;
- docker-ce-cli;
- docker-buildx-plugin.

Загрузив файлы на сервер установите Docker-engine из бинарных файлов выполнив следующую команду:

```
sudo dpkg -i ./containerd.io_1.6.9-1_amd64.deb  ./docker-ce_23.0.6-1~debian.10~buster_amd64.deb \
./docker-ce-cli_24.0.2-1~debian.10~buster_amd64.deb ./docker-buildx-plugin_0.10.5-1~debian.10~buster_amd64.deb
```

Проверка наличия Docker:

```
docker --version
```

Установка Docker (для Ubuntu/Debian):

Если Docker не установлен, выполните следующие команды:

```
sudo apt-get update
sudo apt-get install docker.io docker-compose
sudo systemctl enable docker
sudo systemctl start docker
```

Добавление пользователя в группу `docker`:

Чтобы избежать необходимости использовать `sudo` при каждом вызове Docker:

```
sudo usermod -aG docker $USER
newgrp docker
```

> ***Примечание для Astra Linux:***
> *В Astra Linux установка Docker имеет специфические особенности, особенно в*

> *изолированных средах без доступа к интернету. Рекомендуется ознакомиться с инструкцией: Установка wiCore без доступа к сети интернет*

2. Развертывание Kafka через Docker Compose

2.1 Создайте файл `docker-compose.yml` со следующим содержимым и замените в `KAFKA_CFG_ADVERTISED_LISTENERS` адрес `127.0.0.1` на актуальный IP-адрес сервера wiSLA:

> Пути к докер манифестам bitnami обновлены в связи с недоступностью новых, Bitnami объявила о переходе многих своих Docker-образов в статус "legacy".

Актуальные данные файла `docker-compose.yml` на 01.10.2025:

```
version: '3.8'

services:
  zookeeper:
    image: docker.io/bitnamilegacy/zookeeper:3.9
    container_name: zookeeper
    ports:
      - "2187:2181"
    environment:
      - ALLOW_ANONYMOUS_LOGIN=yes
    volumes:
      - "zookeeper_data:/bitnami/zookeeper"
    restart: unless-stopped

  kafka:
    image: docker.io/bitnamilegacy/kafka:3.7
    container_name: kafka
    ports:
      - "9092:9092"
      - "29092:29092"
    environment:
      - KAFKA_CFG_ZOOKEEPER_CONNECT=zookeeper:2181
      - ALLOW_PLAINTEXT_LISTENER=yes
      - KAFKA_CFG_LISTENERS=PLAINTEXT://:9092,INTERNAL://:29092
      - KAFKA_CFG_ADVERTISED_LISTENERS=PLAINTEXT://192.168.110.128:9092,INTERNAL://kafka:29092
      - KAFKA_CFG_LISTENER_SECURITY_PROTOCOL_MAP=PLAINTEXT:PLAINTEXT,INTERNAL:PLAINTEXT
      - KAFKA_CFG_INTER_BROKER_LISTENER_NAME=INTERNAL
      - KAFKA_CFG_AUTO_CREATE_TOPICS_ENABLE=true
      - KAFKA_CFG_NUM_PARTITIONS=1
      - KAFKA_CFG_DELETE_TOPIC_ENABLE=true
      - KAFKA_CFG_LOG_RETENTION_HOURS=24
      - KAFKA_CFG_LOG_SEGMENT_BYTES=5242880
      - KAFKA_CFG_MESSAGE_MAX_BYTES=1048576
      - KAFKA_CFG_SOCKET_REQUEST_MAX_BYTES=10485760
      - KAFKA_CFG_LOG_RETENTION_BYTES=53687091
    volumes:
      - "kafka_data:/bitnami/kafka"
```

```
    depends_on:
      - zookeeper
    restart: unless-stopped

  kafka-console:
    image: docker.redpanda.com/redpandadata/console:latest
    container_name: kafka-console
    ports:
      - "8085:8080"
    environment:
      KAFKA_BROKERS: kafka:29092
      SERVER_PORT: 8080
      CONSOLE_LOG_LEVEL: info
      CONSOLE_FEATURES_TOPIC_INSPECTOR: "true"
    depends_on:
      - kafka
    restart: unless-stopped

volumes:
  zookeeper_data:
    driver: local
  kafka_data:
    driver: local
```

Необходимо прописать конфиг:

```
KAFKA_CFG_ADVERTISED_LISTENERS=PLAINTEXT://{hostname}:9092
```

Данный параметр определяет **адрес (listeners)**, по которым клиенты могут подключаться к брокеру Kafka.

> ***Примечание**: В этом же файле можно дополнительно настроить параметры работы системы, такие как срок хранения данных (по умолчанию - 24 часа), который может очищаться основываясь на указанный временной период, либо при максимальном объеме хранимой информации.*

2.2 Запустите Kafka:

```
sudo docker-compose up -d
```

2.3 Проверьте работоспособности

Убедитесь, что запущены три контейнера: `zookeeper`, `kafka`, `kafka-console`:

```
sudo docker ps
```

```
ppichugin@alfa-test:/opt/kafka$ sudo docker ps
CONTAINER ID   IMAGE                                          COMMAND                  CREATED         STATUS         PORTS                                                                                  NAMES
ac7e7132078a   docker.redpanda.com/redpandadata/console:latest   "./console"              5 minutes ago   Up 5 minutes   0.0.0.0:8085->8080/tcp, [::]:8085->8080/tcp                                            kafka-console
3a0c5858ff60   bitnami/kafka:3.7                              "/opt/bitnami/script…"   5 minutes ago   Up 5 minutes   0.0.0.0:9092->9092/tcp, [::]:9092->9092/tcp, 0.0.0.0:29092->29092/tcp, [::]:29092->29092/tcp   kafka
4788f8d99b4b   bitnami/zookeeper:3.9                          "/opt/bitnami/script…"   5 minutes ago   Up 5 minutes   2888/tcp, 3888/tcp, 8080/tcp, 0.0.0.0:2187->2181/tcp, [::]:2187->2181/tcp              zookeeper
```

Проверьте логи Kafka-контейнера (замените `CONTAINER_ID_KAFKA` на реальный ID):

```
sudo docker logs CONTAINER_ID_KAFKA
```

3. Установка и настройка wiProbe Aggregator (не обязательно)

3.1.Скачивание архива

Скачайте файл `wisla-wiprobe-aggregator.jar` с GitLab.
Подробная инструкция: **Установка wiProbe Aggregator** .

> ***Примечание**: Установка Nginx (который упоминался в статье) не обязательна, если не требуется балансировка нагрузки между несколькими агрегаторами.*

3.2. Создайте каталог для логов (желательно рядом с агрегатором), иначе при запуске агрегатора будут ошибки, что нет прав для записи логов):

```
mkdir -p /home/wisla/aggrlog
chown wisla:wisla /home/wisla/aggrlog
```

3.3. Настройка конфигурации JAR-файла

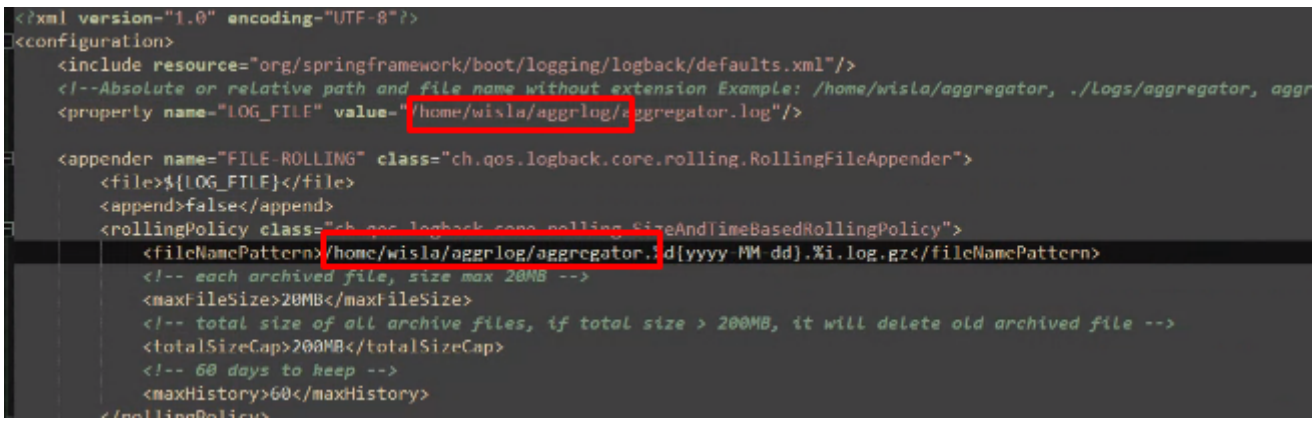
Откройте JAR-архив с помощью архиватора (например, 7-Zip, WinRAR) и отредактируйте:

- **`BOOT-INF/classes/config/application.properties`** указав в параметрах `bootstrap.servers` и `wisla.dc-nodes` необходимый ip:

```
server.servlet.context-path=/aggregator
server.port=18080
aggregator.kafka-enabled=true
aggregator.wisla-kafka.bootstrap.servers=ВАШ_IP:9092  # Например: 10.11.11.52:9092
aggregator.wisla-kafka.consumer-threads-count=4
aggregator.wisla.dc-nodes=http://ВАШ_DC_УЗЕЛ:8080  # Например: http://10.11.11.51:8080
```

- **`BOOT-INF/classes/logback-spring.xml`** в параметрах **property** и **fileNamePattern** прописать правильные пути для логов:

```
<property name="LOG_FILE" value="/home/wisla/aggrlog/aggregator.log" />
.....
<fileNamePattern>/home/wisla/aggrlog/aggregator.%d{yyyy-MM-dd}.%i.log.gz</fileNamePattern>
```



3.4. Настройка автозапуска через systemd

- **Создайте systemd unit** для автоматического запуска агрегатора, с указанием пути, где лежит агрегатор:

```
sudo nano /etc/systemd/system/aggr-jar.service
```

- **Вставьте конфигурацию** (адаптируйте пути и имя пользователя!):

```
[Unit]
Description=WiProbeAggregator

[Service]
ExecStart=/usr/bin/java -jar /home/smpsadmin/wisla-wiprobe-aggregator.jar
Restart=always
User=wisla

[Install]
WantedBy=multi-user.target
```

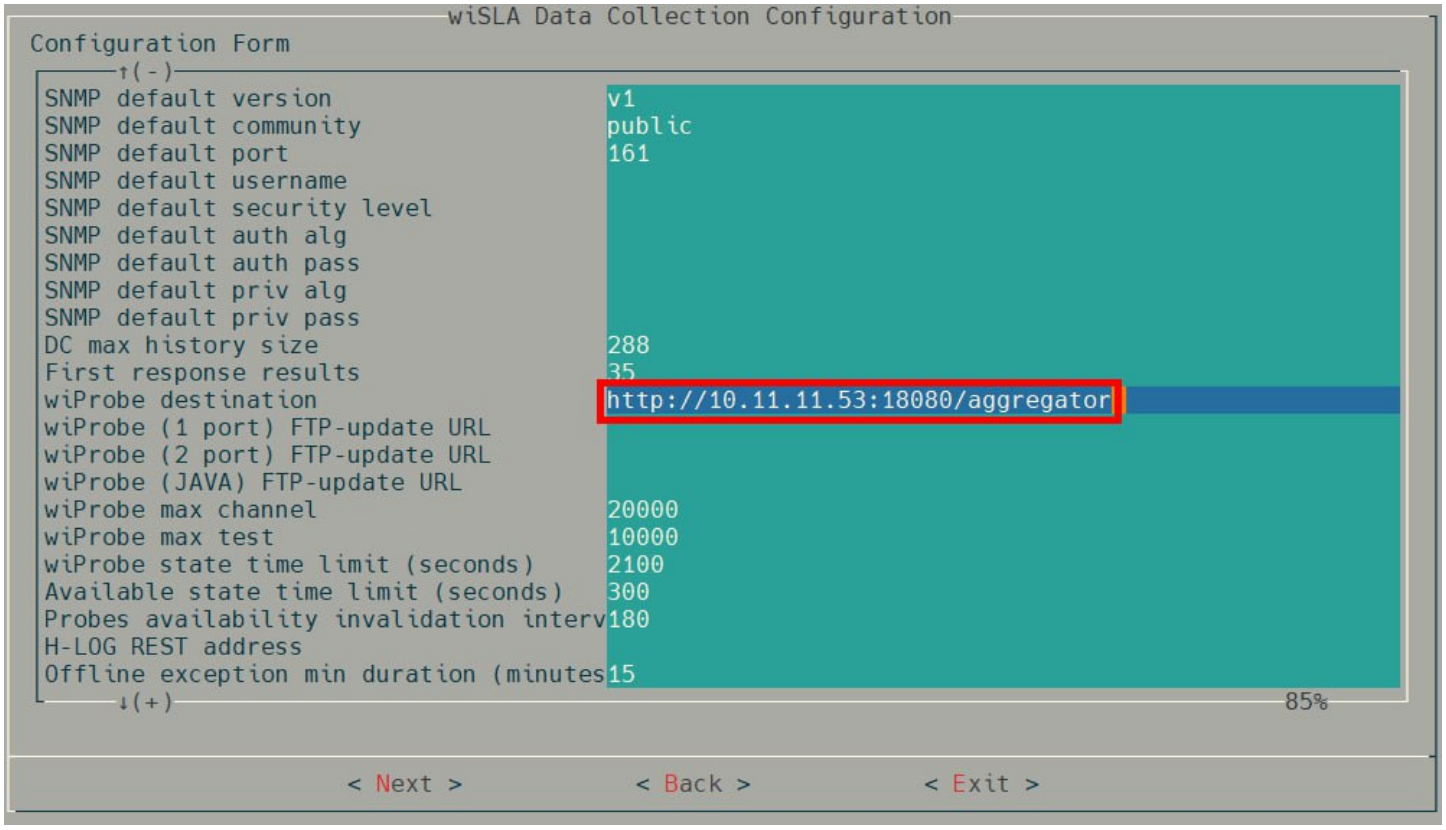
3.5. Запуск и проверка:

```
# Обновить конфигурацию
sudo systemctl daemon-reload

# Включить автозагрузку
sudo systemctl enable aggr-jar.service

# Запустить службу
sudo systemctl start aggr-jar.service

# Проверить статус
sudo systemctl status aggr-jar.service

# Просмотр логов (реальный режим)
journalctl -u aggr-jar.service -f
```

4. Настройка wiSLA для работы с Kafka и Aggregator

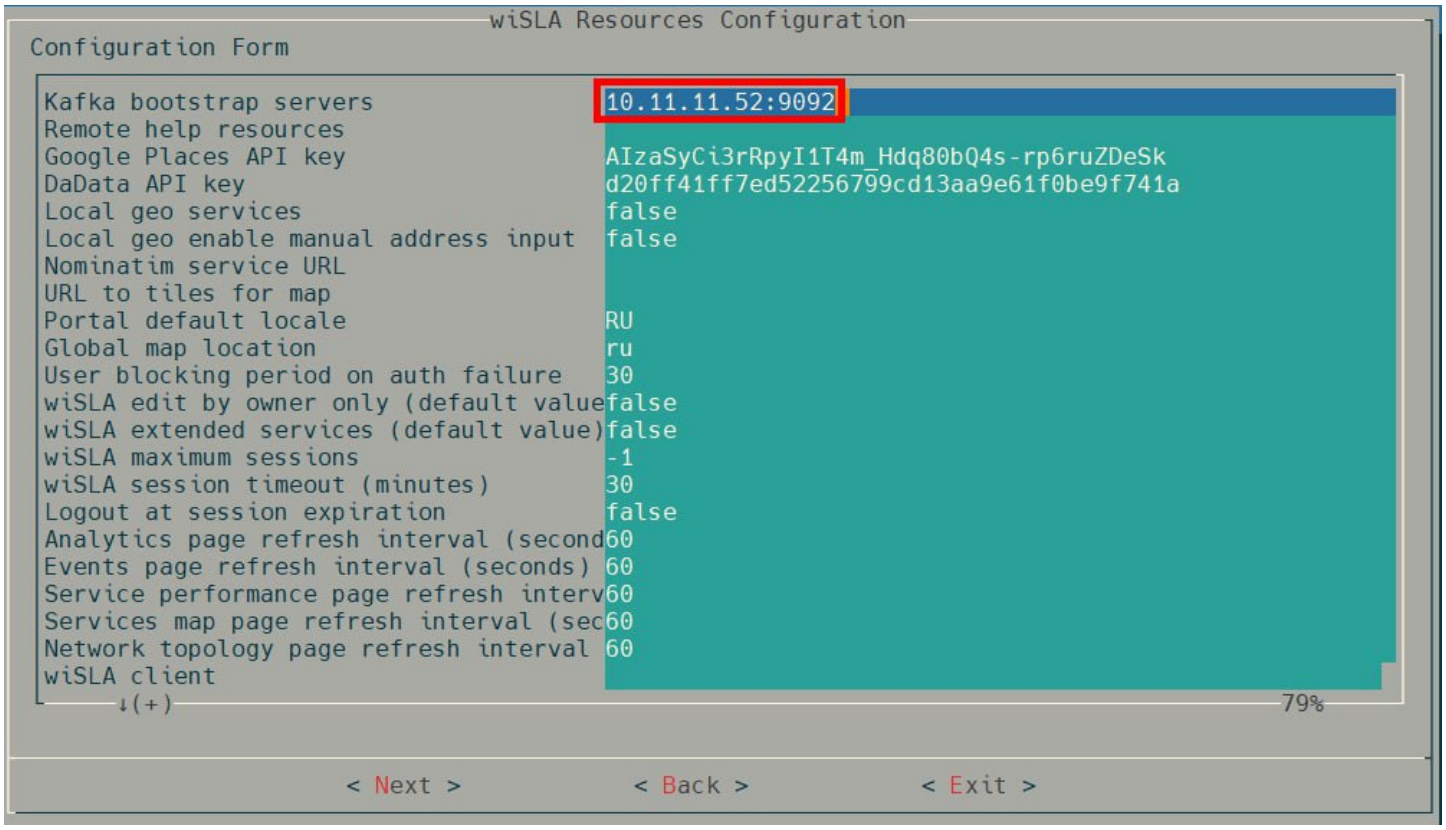
4.1. В `wiSLA Data Collection Configuration` укажите адрес агрегатора:

```
wtProbe destination = http://ВАШ_IP_АГРЕГАТОРА:18080/aggregator
```



4.2. В `wiSLA Resources Configuration` укажите адрес Kafka:

```
Kafka bootstrap servers = ВАШ_IP_КАФКА:9092
```



5. Особенности работы с Kafka и агрегатором

5.1. Варианты конфигурации системы

Standalone-режим (один сервер wiSLA):

- Полноценная работа с логированием
- Данные идут через агрегатор → Kafka
- Просмотр логов через Redpanda Console (http://IP_KAFKA:8085)
- *Конфиг*: WISLA_WIPROBE_DESTINATION = http://aggregator_ip:18080/aggregator

Рекомендуется для новых установок

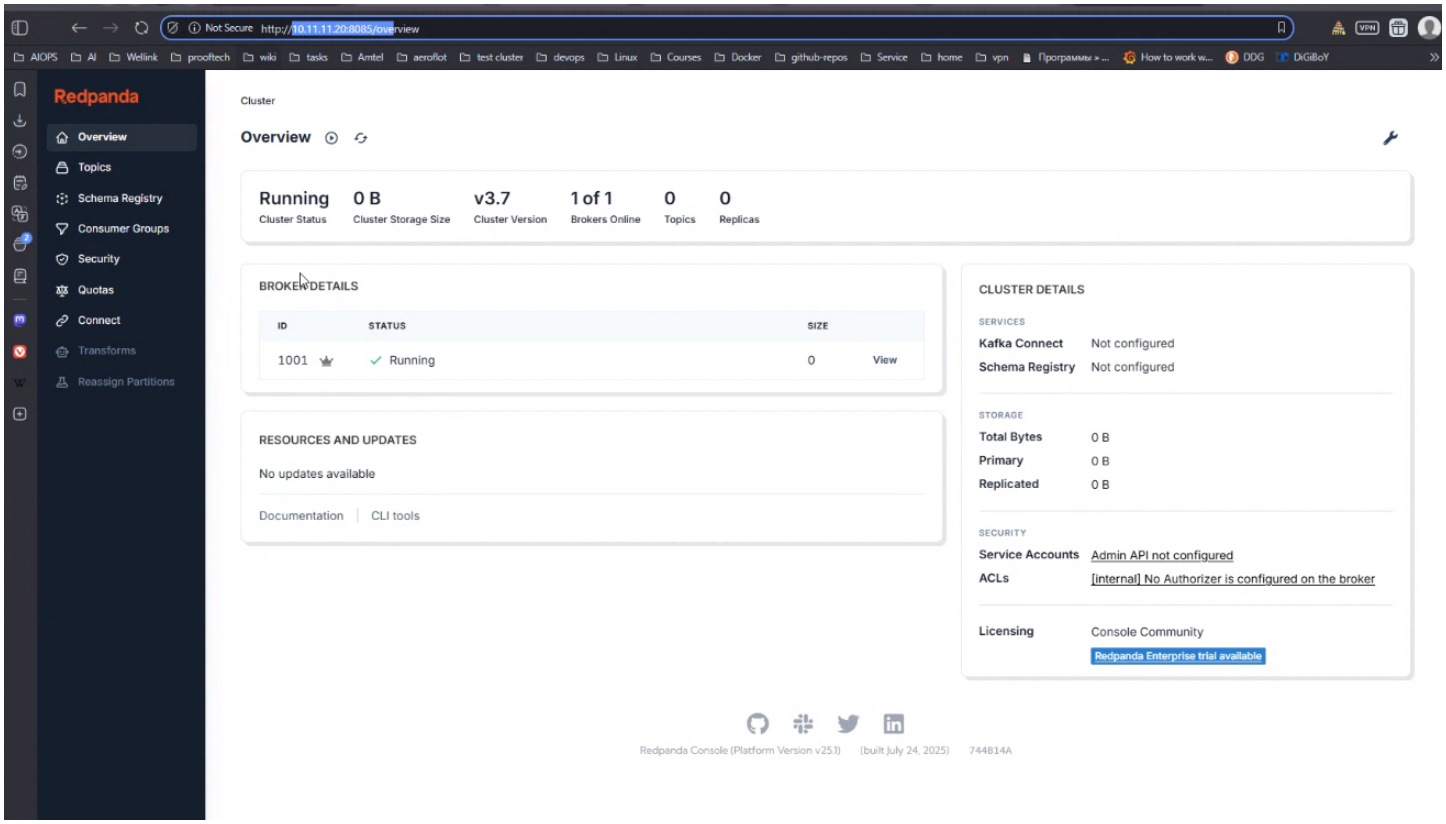
Кластерный режим:

- **Только с Kafka + агрегатор**:
  - Агрегатор - модуль системы принимает данные (результаты измерений) от агентов/зондов (по rest) и складывает в соответствующие топики kafka.
  - *Конфиг*: `WISLA_WIPROBE_DESTINATION` `= http://aggregator_ip:18080/aggregator`

5.2. Важные замечания

- **Redpanda** доступен по указанному в `docker-compose.yml` порту консоли кафки, через: `http://IP_СЕРВЕРА:8085`
- **Настройки** `WISLA_WIPROBE_DESTINATION` - url получаемая wiProbe при авторегистрации для отправки данных (результатов измерений). (url авторегистрации можно изменить только при выполнении команды slamon-conf url .....)
- В **кластере** и **standalone** обязательно указывайте адрес агрегатора (`http://aggregator_ip:18080/aggregator`)

> **Примечание**: Даже для standalone необходимо развернуть Kafka и агрегатор – это обеспечит единый подход к логированию и упростит переход на кластерную архитектуру в будущем.


Not Secure http://10.11.11.20:8085/overview
Redpanda
Overview
Topics
Schema Registry
Consumer Groups
Security
Quotas
Connect
Transforms
Reassign Partitions
Cluster
Overview
Running
Cluster Status
0 B
Cluster Storage Size
v3.7
Cluster Version
1 of 1
Brokers Online
0
Topics
0
Replicas
BROKER DETAILS
ID
STATUS
SIZE
1001
Running
0
View
RESOURCES AND UPDATES
No updates available
Documentation
CLI tools
CLUSTER DETAILS
SERVICES
Kafka Connect
Not configured
Schema Registry
Not configured
STORAGE
Total Bytes
0 B
Primary
0 B
Replicated
0 B
SECURITY
Service Accounts
Admin API not configured
ACLs
[internal] No Authorizer is configured on the broker
Licensing
Console Community
Redpanda Enterprise trial available
Redpanda Console (Platform Version v2.5.1)
(built July 24, 2025)
744814A