
	<p>Politechnika Bydgoska im. J. J. Śniadeckich Wydział Telekomunikacji, Informatyki i Elektrotechniki <b>Zakład Systemów Teleinformatycznych</b></p>		
<b>Przedmiot</b>	Podstawy inżynierii danych		
<b>Prowadzący</b>	prof. dr hab. inż. prof. PBS Piotr Cofta		
<b>Temat</b>	Laboratorium 2		
<b>Student</b>	Cezary Tytko		
<b>Ocena</b>		<b>Data oddania spr.</b>	24.10.2024

## Etap 1. – Instalacja jednej instancji brokera.

```

cezary@LAPTOP-SF015Q2L:/$ ./srv/lib/tiny-kafka.sh
Deleted Containers:
5a43a430f263221378db7dd9e3c5070d874bfff7d1eb384cc75b631de04c65009
78383ebdfdebfd7a39ed7a7dea69aeda8ba48371fbde828937fafda70c98cd7e

Total reclaimed space: 2B
Unable to find image 'confluentinc/cp-kafka:7.4.0' locally
7.4.0: Pulling from confluentinc/cp-kafka
36c12cb044ac: Pull complete
3cded401634e: Pull complete
bcafa67fbd8f: Pull complete
241ffc663b43: Pull complete
015862cc789d: Pull complete
c180ee728a3e: Pull complete
e1e6a53f9063: Pull complete
8c144520e9de: Pull complete
1fe5b5b014c2: Pull complete
6bb510d48a09: Pull complete
f54714f83ac5: Pull complete
Digest: sha256:187dac6627e7906c350f5f8c982f80ce735fff1a0e571a20de6000a309a12ce63
Status: Downloaded newer image for confluentinc/cp-kafka:7.4.0
92f00ea12a3bd1bd8c98d7901e93384f7a5260304d25276b99d6ad3c56e7d30a
cezary@LAPTOP-SF015Q2L:/$ docker ps
CONTAINER ID   IMAGE                                COMMAND                                     CREATED        STATUS        PORTS
92f00ea12a3b   confluentinc/cp-kafka:7.4.0         "/etc/confluent/dock..."               22 seconds ago Up 12 seconds 0.0.0.0:9092-9093
->9092-9093/tcp, :::9092-9093->9092-9093/tcp, 0.0.0.0:29092->29092/tcp, :::29092->29092/tcp   kafka
cezary@LAPTOP-SF015Q2L:/$

```

Skrypt uruchomił się bez błędów, odpowiednie warstwy obrazu zostały pobrane i został utworzony kontener kafki.

## Etap 2. Utworzenie tematu.

```

cezary@LAPTOP-SF015Q2L:/$ docker exec -it kafka /bin/bash
[appuser@kafka ~]$ kafka-topics --bootstrap-server kafka:9092 --create --topic pbs --partitions 4 --replication-factor 1
Created topic pbs.
[appuser@kafka ~]$ kafka-topics --bootstrap-server kafka:9092 --list
pbs
[appuser@kafka ~]$ kafka-topics --bootstrap-server kafka:9092 --describe --topic pbs
Topic: pbs      TopicId: eF_gd5XkQc6v9mAutyAtqQ PartitionCount: 4      ReplicationFactor: 1      Config
s:
    Topic: pbs      Partition: 0      Leader: 1      Replicas: 1      Isr: 1
    Topic: pbs      Partition: 1      Leader: 1      Replicas: 1      Isr: 1
    Topic: pbs      Partition: 2      Leader: 1      Replicas: 1      Isr: 1
    Topic: pbs      Partition: 3      Leader: 1      Replicas: 1      Isr: 1
[appuser@kafka ~]$

```

Efektem wykonania poleceń jest powyższy zrzut ekranu.

Partition – powoduje podział tematu na części co zwiększa wydajność prze możliwość zrównoleglenia operacji

Leader – jest to broker odpowiadający za obsługę operacji zapisów i odczytów dla wskazanej partycji. W tym przykładnie wszystkie partycje mają tego samego lidera „1”.

Replicas – to kopia danych z partycji przechowywanych na innych brokerach. Wartość 1 oznacza, że każda partycja ma tylko jedną replikę (czyli ma tylko oryginał), więc w wypadku awarii nie ma brokera, który mógłby zostać nowym leaderem.

Isr - (In-Sync Replica) to lista replik zsynchronizowanych z leaderem (posiadających pełną kopię danych z partycji). Wartość 1 oznacza, że jedyna replika w pełni odpowiada obecnemu liderowi, w naszym przypadku jest to tylko sam oryginał.

### Etap 3. Wysyłanie i odbiór komunikatów.

Producer wysyłający komunikaty (w tym przypadku wszystkie komunikaty zostały wysłane po utworzeniu pierwszego konsumenta)

```
[appuser@kafka ~]$ kafka-console-producer --bootstrap-server kafka:9092 --topic pbs
>komunikat 1
>komunikat 2
>komunikat 3
>komunikat 4
>komunikat 5
>
```

Konsument nasłuchujący/czekający na komunikaty, otrzymuje komunikaty wysłane już po jego utworzeniu

```
[appuser@kafka ~]$ kafka-console-consumer --bootstrap-server kafka:9092 --topic pbs
komunikat 1
komunikat 2
komunikat 3
komunikat 4
komunikat 5
```

## Konsument w trybie --from-beginning

```
[appuser@kafka ~]$ kafka-console-consumer --bootstrap-server kafka:9092 --topic pbs --from-beginning
komunikat 1
komunikat 2
komunikat 3
komunikat 4
komunikat 5
|
```

W momencie uruchomienia przesyłana jest do niego lista wszystkich komunikatów dla danego tematu od początku, z zachowaniem kolejności

### Etap 4. Wiele procesów.

The image displays four terminal windows illustrating the setup and operation of a Kafka cluster with multiple consumers.

**Top Left Window:** Shows the creation of a topic 'pbs' with 4 partitions and the subsequent listing and description of the topic. The description shows 4 partitions, each with 1 leader and 1 replica. It also shows the start of a console producer sending messages 'komunikat 1' through 'komunikat 5'.

**Top Right Window:** Shows a Docker exec session running a Kafka console producer. It sends two messages: 'komunikat 1' and 'komunikat 2'.

**Bottom Left Window:** Displays a table of running containers and a list of active processes.

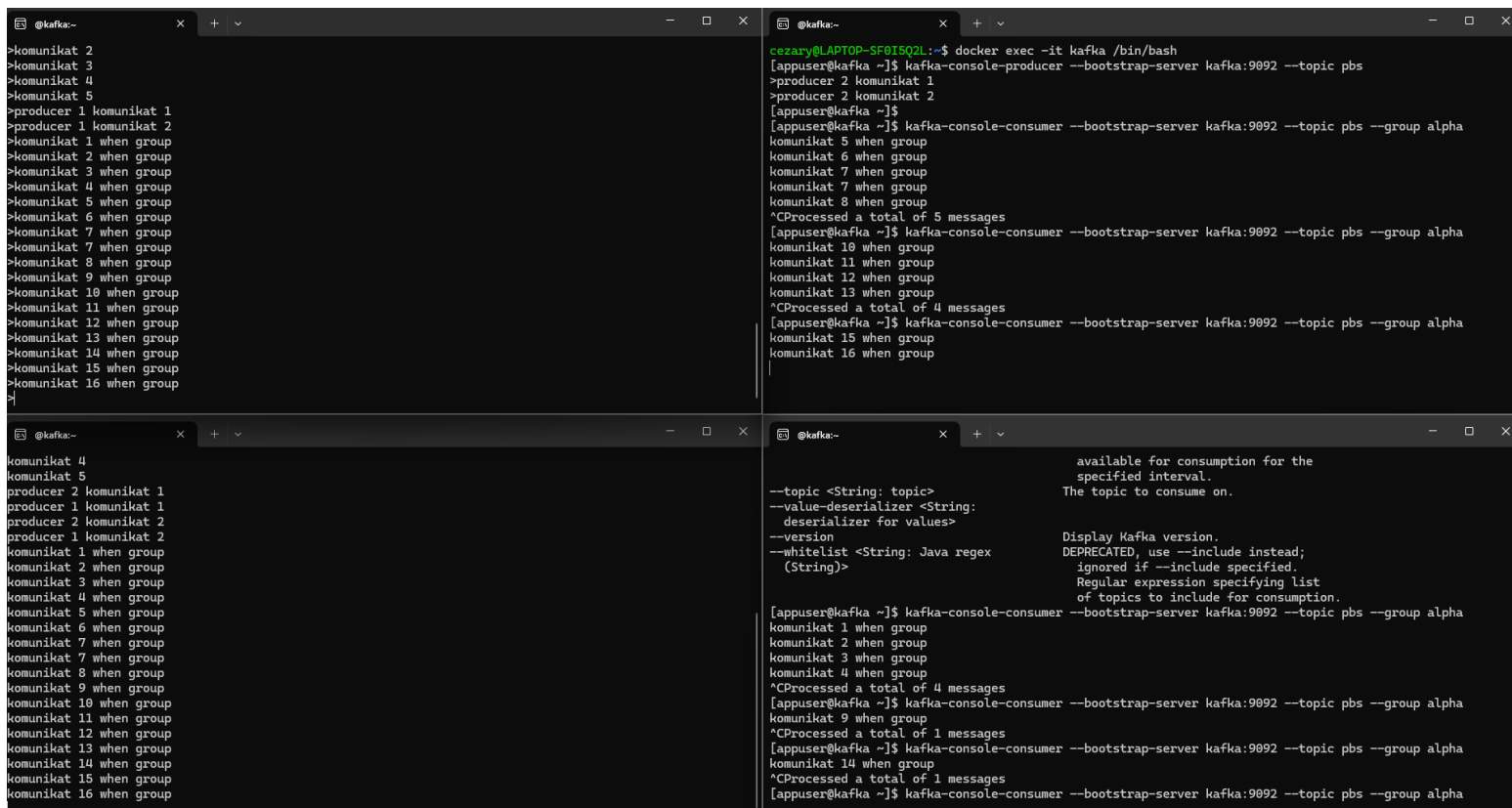
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	P
ORTS					
92f00ea12a3b	confluentinc/cp-kafka:7.4.0	"/etc/confluent/dock_	8 minutes ago	Up 8 minutes	0
.0.0.0:9092-9093->9092-9093/tcp, :::9092-9093->9092-9093/tcp, 0.0.0.0:29092->29092/tcp, :::29092->29092/tcp	kafka				

Below the table, it shows the execution of a Docker exec command and the start of a Kafka console consumer that reads from the beginning of the 'pbs' topic, displaying the received messages.

**Bottom Right Window:** Shows the output of the Kafka console consumer, which has received and processed all five messages ('komunikat 1' through 'komunikat 5') from the 'pbs' topic.

Komunikaty niezależnie od producenta z którego zostały wysłane trafiają zawsze do obu aktywnych słuchaczy

## Etap 5. Grupa konsumentów.



```
@kafka:~$ docker exec -it kafka /bin/bash
[appuser@kafka ~]$ kafka-console-producer --bootstrap-server kafka:9092 --topic pbs
>producer 2 komunikat 1
>producer 2 komunikat 2
[appuser@kafka ~]$
[appuser@kafka ~]$ kafka-console-consumer --bootstrap-server kafka:9092 --topic pbs --group alpha
komunikat 5 when group
komunikat 6 when group
komunikat 7 when group
komunikat 7 when group
komunikat 8 when group
^CProcessed a total of 5 messages
[appuser@kafka ~]$ kafka-console-consumer --bootstrap-server kafka:9092 --topic pbs --group alpha
komunikat 10 when group
komunikat 11 when group
komunikat 12 when group
komunikat 13 when group
^CProcessed a total of 4 messages
[appuser@kafka ~]$ kafka-console-consumer --bootstrap-server kafka:9092 --topic pbs --group alpha
komunikat 15 when group
komunikat 16 when group

komunikat 4
komunikat 5
producer 2 komunikat 1
producer 1 komunikat 1
producer 2 komunikat 2
producer 1 komunikat 2
komunikat 1 when group
komunikat 2 when group
komunikat 3 when group
komunikat 4 when group
komunikat 5 when group
komunikat 6 when group
komunikat 7 when group
komunikat 7 when group
komunikat 8 when group
komunikat 9 when group
komunikat 10 when group
komunikat 11 when group
komunikat 12 when group
komunikat 13 when group
komunikat 14 when group
komunikat 15 when group
komunikat 16 when group

--topic <String: topic>          available for consumption for the
--value-deserializer <String:   specified interval.
                                The topic to consume on.
--value-deserializer <String:   deserializer for values>
--version                        Display Kafka version.
--whitelist <String: Java regex  DEPRECATED, use --include instead;
                                ignored if --include specified.
                                Regular expression specifying list
                                of topics to include for consumption.

[appuser@kafka ~]$ kafka-console-consumer --bootstrap-server kafka:9092 --topic pbs --group alpha
komunikat 1 when group
komunikat 2 when group
komunikat 3 when group
komunikat 4 when group
^CProcessed a total of 4 messages
[appuser@kafka ~]$ kafka-console-consumer --bootstrap-server kafka:9092 --topic pbs --group alpha
komunikat 9 when group
^CProcessed a total of 1 messages
[appuser@kafka ~]$ kafka-console-consumer --bootstrap-server kafka:9092 --topic pbs --group alpha
komunikat 14 when group
^CProcessed a total of 1 messages
[appuser@kafka ~]$ kafka-console-consumer --bootstrap-server kafka:9092 --topic pbs --group alpha
```

Przy komunikacji jeżeli w danej grupie jest tylko jeden odbiorca działa tak samo jak do tej pory, komunikaty są wysłane do odbiorcy bez grupy jak i tego w grupie.

Natomiast jeżeli dodamy do grupy kolejnego odbiorcę to wszystkie komunikaty będą przesyłane tylko do jednego z grupy (do odbiorców bez grupy bez zmian są przesyłane wszystkie), i zawsze tego samego odbiorcy, kiedy odłączymy odbiorcę z grupy który aktualnie przejmuje komunikaty, to jego rolę przejmie kolejny dostępny z grupy odbiorca, zaobserwowałem również że jeżeli przywrócić wyłączonego odbiorcę, to on zacznie na powrót odbierać nowe komunikaty.