

## Département Mathématiques et Informatique : II-BDCC 3

### Travail Pratique 1: Systèmes Distribués

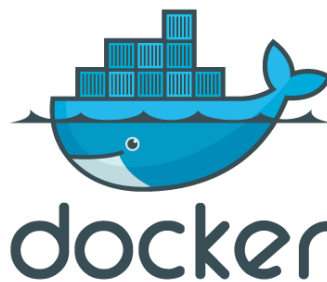
#### Activité pratique N° 1 : Event Driven Architecture

-Réalisé par :

**Madani Cherif Oumaima**

-Diriger par :

**Monsieur Mohamed YOUSSEFI**



## Introduction

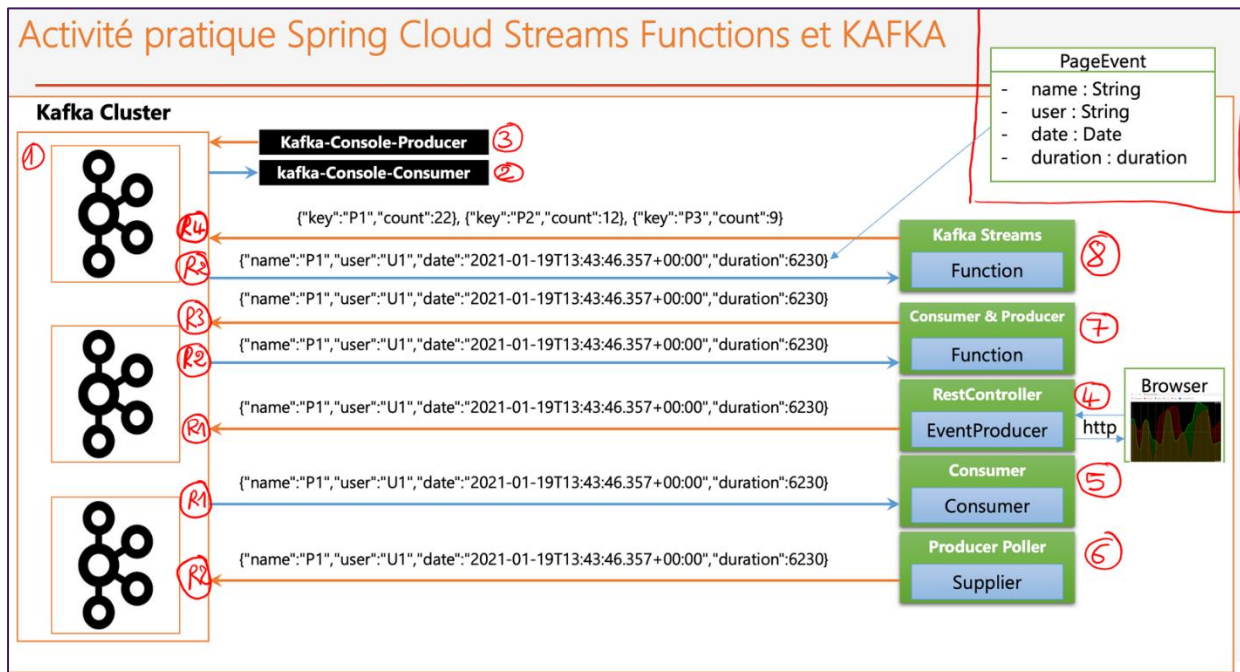
Dans ce TP, nous allons explorer Apache Kafka et Docker pour mettre en place un environnement de streaming de données. La première partie de l'activité se concentre sur l'installation et le démarrage de Kafka, ainsi que sur les tests à l'aide de `kafka-console-producer` et `kafka-console-consumer`. La deuxième partie implique l'utilisation de Docker pour déployer Zookeeper et Kafka, suivie des mêmes tests. Enfin, la troisième partie vise à créer plusieurs services Kafka, y compris un producteur, un consommateur, un fournisseur et un service de traitement de flux analytique en temps réel à l'aide de Kafka Streams, ainsi qu'une application Web pour afficher les résultats du traitement en temps réel.

### **Sommaire :**

<b>Introduction .....</b>	<b>1</b>
<b>Partie 1: Mise en place de KAFKA .....</b>	<b>3</b>
<b>Partie 2: Création d'une Application d'analyse de donnée .....</b>	<b>6</b>
<b>Conclusion .....</b>	<b>9</b>

## Parties pratiques:

### Partie 0: Architecture de l'activité pratique



### Partie 1: Mise en place de KAFKA

- Télécharger Kafka.

PC > Nouveau nom (D:) > kafka\_2.13-3.5.1

Nom	Modifié le	Type	Taille
bin	14/07/2023 17:53	Dossier de fichiers	
config	14/07/2023 17:53	Dossier de fichiers	
libs	14/07/2023 17:53	Dossier de fichiers	
licenses	14/07/2023 17:53	Dossier de fichiers	
logs	21/12/2023 16:18	Dossier de fichiers	
site-docs	14/07/2023 17:53	Dossier de fichiers	
LICENSE	14/07/2023 17:50	Fichier	15 Ko
NOTICE	14/07/2023 17:50	Fichier	28 Ko

Le package est prêt à l'emploi, comme montre l'image en haut.

- Démarrer Zookeeper

```
$ zookeeper-server-start.bat ./zookeeper.properties
[2023-10-04 18:06:31,100] INFO Reading configuration from: ./zookeeper.properties
[2023-10-04 18:06:31,100] WARN \tmp\zookeeper is relative. Prepend .\ to indicate that you're sure! (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2023-10-04 18:06:31,115] INFO clientPortAddress is 0.0.0.0:2181 (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2023-10-04 18:06:31,115] INFO secureClientPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2023-10-04 18:06:31,115] INFO observerMasterPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2023-10-04 18:06:31,115] INFO metricsProvider.className is org.apache.zookeeper.metrics.impl.DefaultMetricsProvider (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2023-10-04 18:06:31,115] INFO autopurge.snapRetainCount set to 3 (org.apache.zookeeper.server.DataDirCleanupManager)
[2023-10-04 18:06:31,115] INFO autopurge.purgeInterval set to 0 (org.apache.zookeeper.server.DataDirCleanupManager)
[2023-10-04 18:06:31,115] INFO Purge task is not scheduled. (org.apache.zookeeper.server.DataDirCleanupManager)
[2023-10-04 18:06:31,115] WARN Either no config or no quorum defined in config, running in standalone mode (org.apache.zookeeper.server.quorum.QuorumPeerMain)
```

- *Démarrer Kafka-server*

[illegible]

- *Tester avec Kafka-console-producer et kafka-console-consumer*

```
C:\Windows\system32\cmd.exe - bin\windows\kafka-console-consumer.bat --boot-
OT_AVAILABLE} (org.apache.kafka.clients.NetworkClient)
[2023-09-30 22:03:28,167] WARN [Consumer clientId=console-consumer, groupId=consol
e-consumer-466] Error while fetching metadata with correlation id 6 : {R1=LEADER_N
OT_AVAILABLE} (org.apache.kafka.clients.NetworkClient)
[2023-09-30 22:03:28,873] WARN [Consumer clientId=console-consumer, groupId=consol
e-consumer-466] Error while fetching metadata with correlation id 8 : {R1=LEADER_N
OT_AVAILABLE} (org.apache.kafka.clients.NetworkClient)
[2023-09-30 22:03:29,288] WARN [Consumer clientId=console-consumer, groupId=consol
e-consumer-466] Error while fetching metadata with correlation id 10 : {R1=LEADER_
NOT_AVAILABLE} (org.apache.kafka.clients.NetworkClient)
aaa
dddd
f
e
r
oooo
oumaima
```

The screenshot shows a web browser with the address bar displaying `http://localhost:8080/publish/R1/blog`. The browser's address bar also shows several tabs: "Réseaux, Systè...", "Développement Inf...", "Search | Scribd", "Résumé cours - Go...", "Name | Faker", "Machine Learning i...", and "Téléversements by...".

The main content area of the browser displays a JSON object:

```
{
  "name": "blog",
  "user": "U2",
  "date": "2023-09-30T22:40:16.975+00:00",
  "duration": 7587
}
```

On the right side of the browser, there are two buttons: "Raw" and "Parse".

In the foreground, a terminal window is open, showing the command prompt `C:\Windows\system32\cmd.exe - bin\windows\kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic R1`. The terminal output shows a list of messages:

```
{ "name": null, "user": "U1", "date": "2023-09-30T22:35:02.077+00:00", "duration": 276 }
{ "name": null, "user": "U2", "date": "2023-09-30T22:37:00.003+00:00", "duration": 6927 }
{ "name": null, "user": "U2", "date": "2023-09-30T22:39:10.585+00:00", "duration": 4316 }
{ "name": "blog", "user": "U2", "date": "2023-09-30T22:40:16.975+00:00", "duration": 7587 }
```

- Créer le fichier `docker-compose.yml`



```
D: > S5_OMC > System distributed > Systemes-Distribues-S5-MrYousfi > Activité pratique N°1 = Event Driven Architecture > demo-springcloud-stream

1 |version: '3'
2 |services:
3 |  zookeeper:
4 |    image: confluentinc/cp-zookeeper:7.3.0
5 |    container_name: zookeeper
6 |    environment:
7 |      ZOOKEEPER_CLIENT_PORT: 2181
8 |      ZOOKEEPER_TICK_TIME: 2000
9 |
10 |  broker:
11 |    image: confluentinc/cp-kafka:7.3.0
12 |    container_name: broker
13 |    ports:
14 |      # To learn about configuring Kafka for access across networks see
15 |      # https://www.confluent.io/blog/kafka-client-cannot-connect-to-broker-on-aws-on-docker-etc/
16 |      - "9092:9092"
17 |    depends_on:
18 |      - zookeeper
19 |    environment:
20 |      KAFKA_BROKER_ID: 1
21 |      KAFKA_ZOOKEEPER_CONNECT: 'zookeeper:2181'
22 |      KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: PLAINTEXT:PLAINTEXT,PLAINTEXT_INTERNAL:PLAINTEXT
23 |      KAFKA_ADVERTISED_LISTENERS: PLAINTEXT://localhost:9092,PLAINTEXT_INTERNAL://broker:29092
24 |      KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR: 1
25 |      KAFKA_TRANSACTION_STATE_LOG_MIN_ISR: 1
26 |      KAFKA_TRANSACTION_STATE_LOG_REPLICATION_FACTOR: 1
27
```

- Démarrer les conteneurs docker : zookeeper et kafka-broker

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
7282bf99aaa7	confluentinc/cp-kafka:7.3.0	"/etc/confluent/dock..."	5 minutes ago	Up 57 seconds	0.0.0.0:9092->9092/tcp, :::9092->9092/tcp	broker
013afe50721f	confluentinc/cp-zookeeper:7.3.0	"/etc/confluent/dock..."	6 minutes ago	Up 58 seconds	2181/tcp, 2888/tcp, 3888/tcp	zookeeper

```
Creating network "spring-cloud-streams-kafka_default" with the default driver
Pulling zookeeper (confluentinc/cp-zookeeper:7.3.0)...
7.3.0: Pulling from confluentinc/cp-zookeeper
d5d2e87c6892: Downloading [=====] 18.52MB/37.55MB
008dba906bf6: Download complete
d5d2e87c6892: Pull complete
008dba906bf6: Pull complete
bfeaabe01655: Downloading [=====] 115.2MB/241.2MB
bfeaabe01655: Pull complete
2cb7eb0f5666: Pull complete
f70f416c6ce7: Pull complete
bc67d000e59b: Pull complete
d6e744651f37: Pull complete
d32323e291f3: Downloading [=====] 61.96MB/106.3MB
4108e73e61e1: Pull complete
d32323e291f3: Downloading [=====] 47.39MB/106.3MB
fa08a06f385f: Pull complete
bddb49e2fc4d: Pull complete
```

```
Status: Downloaded newer image for confluentinc/cp-kafka:7.3.0
Creating zookeeper ... done
Creating broker ... done
Attaching to zookeeper, broker for confluentinc/cp-zookeeper:7.3.0
broker | ==> Usertinc/cp-kafka:7.3.0...
broker | uid=1000(appuser) gid=1000(appuser) groups=1000(appuser)
broker | ==> Configuring ...
zookeeper | ==> Userg [=====] 46.85MB/106.3MB
zookeeper | uid=1000(appuser) gid=1000(appuser) groups=1000(appuser) ] 45.78MB/106.3MB
zookeeper | ==> Configuring ...
zookeeper | ==> Running preflight checks ... ] 45.23MB/106.3MB
zookeeper | ==> Check if /var/lib/zookeeper/data is writable ...
zookeeper | ==> Check if /var/lib/zookeeper/log is writable ... ] 44.7MB/106.3MB
zookeeper | ==> Launching ...
zookeeper | ==> Launching zookeeper ... ==> ] 44.16MB/106.3MB
broker | ==> Running preflight checks ...
broker | ==> Check if /var/lib/kafka/data is writable ... ] 43.63MB/106.3MB
zookeeper | [2023-10-15 12:47:11,504] INFO Reading configuration from: /etc/kafka/zookeeper.prop
ookeeper.server.quorum.QuorumPeerConfig)
```

- Tester avec Kafka-console-producer et kafka-console-consumer

```
docker exec -it dbafc21c611472193df90d24975f5d5d04af0a582feb92845532106879c9c...
sh-4.4$ kafka-console-producer --broker-list localhost:9093 --topic T1
```

## Partie 2: Création d'une Application d'analyse de donnée

En Utilisant KAFKA et Spring Cloud Streams, Créer :

- Un Service Producer KAFKA via un Rest Controller

```
@RestController
public class PageEventRestController {

    @Autowired
    private StreamBridge streamBridge;

    // publish method
    @GetMapping("/publish/{topic}/{name}")
    public PageEvent publishEvent(@PathVariable String topic, @PathVariable String
name) {
        PageEvent pageEvent = new PageEvent(name, "user_" + new Random().nextInt(100),
new Date(),
            10 + new Random().nextInt(100));
        streamBridge.send(
            topic,
            pageEvent);
        return pageEvent;
    }
}
```

```
C:\Windows\system32\cmd.exe - kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic R1
[2023-10-06 22:56:39,994] WARN [Consumer clientId=console-consumer, groupId=console-consumer-46143] Error
while fetching metadata with correlation id 7 : {R1=LEADER_NOT_AVAILABLE} (org.apache.kafka.clients.Netw
orkClient)
[2023-10-06 22:56:40,102] WARN [Consumer clientId=console-consumer, groupId=console-consumer-46143] Error
while fetching metadata with correlation id 8 : {R1=LEADER_NOT_AVAILABLE} (org.apache.kafka.clients.Netw
orkClient)
{"name":"Ahmed","user":"user_58","date":"2023-10-06T21:58:17.291+00:00","duration":10}
```

- Un Service Consumer KAFKA

```
// Supplier
@Bean
public Supplier<PageEvent> pageEventSupplier() {
    return () -> new PageEvent( name: "OUM_" + new Random().nextInt( bound: 100),
        user: "user_" + new Random().nextInt( bound: 100),
        new Date(), duration: 10 + new Random().nextInt( bound: 100));
}
```

- Un Service Supplier KAFKA

```
// Consumer
@Bean
public Consumer<PageEvent> pageEventConsumer() {
    return (input) -> {
        System.out.println("*****");
        System.out.println(input.toString());
        System.out.println("*****");
    };
}
```

- Un Service de Data Analytics Real Time Stream Processing avec Kafka Streams

```
// Function
@Bean
public Function<PageEvent, PageEvent> pageEventFunction() {
    return (input) -> {
        input.setDuration(input.getDuration() * 2);
        input.setName(input.getName().toUpperCase());
        input.setUser(input.getUser().toUpperCase());
        return input;
    };
}
```

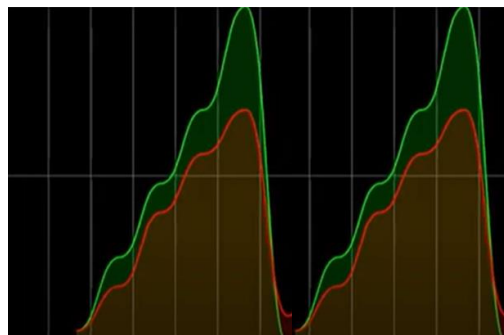
```
C:\Windows\system32\cmd.exe - kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic R4
[2023-10-07 00:02:01,337] WARN [Consumer clientId=console-consumer, groupId=console-consumer-17650] Error while fetching
metadata with correlation id 2 : {R4=LEADER_NOT_AVAILABLE} (org.apache.kafka.clients.NetworkClient)
[2023-10-07 00:02:01,448] WARN [Consumer clientId=console-consumer, groupId=console-consumer-17650] Error while fetching
metadata with correlation id 7 : {R4=LEADER_NOT_AVAILABLE} (org.apache.kafka.clients.NetworkClient)
[2023-10-07 00:02:01,579] WARN [Consumer clientId=console-consumer, groupId=console-consumer-17650] Error while fetching
metadata with correlation id 9 : {R4=LEADER NOT AVAILABLE} (org.apache.kafka.clients.NetworkClient)
```

```
C:\Windows\system32\cmd.exe - kafka-console-consumer.bat --bootstrap-server localhost:9092 --t...
{"name":"said","user":"user_36","date":"2023-10-07T11:28:53.470+00:00","duration":105}
```

- Une application Web qui permet d'afficher les résultats du Stream Data Analytics en temps réel

```
@RestController
public class PageEventRestController {
    @Autowired
    private StreamBridge streamBridge;
    @Autowired
    private InteractiveQueryService interactiveQueryService;
    @GetMapping("/{topic}/{name}")
    public PageEvent publish(@PathVariable String topic, @PathVariable String name){
        PageEvent pageEvent=new PageEvent(name,Math.random()>0.5?"U1":"U2",new Date(),new Random().nextInt(100));
        streamBridge.send(topic,pageEvent);
        return pageEvent;
    }
    @GetMapping(value = "/analytics", produces = MediaType.TEXT_EVENT_STREAM_VALUE)
    public Flux<Map<String,Long>> analytics(){
        return Flux.interval(Duration.ofSeconds(1))
            .map(seq->{
                Map<String,Long> stringLongMap=new HashMap<>();
                ReadOnlyWindowStore<String, Long> windowStore = interactiveQueryService.getQueryableStore(topic);
                Instant now=Instant.now();
                Instant from=now.minusMillis(5000);
                KeyValueIterator<Windowed<String>, Long> fetchAll = windowStore.fetchAll(from, now);
                while (fetchAll.hasNext()){
                    KeyValue<Windowed<String>, Long> next = fetchAll.next();
                    stringLongMap.put(next.key.key(),next.value);
                }
            })
    }
}
```

- Index.html





```

randomColor = function () {
    ++index;
    if (index >= colors.length) index = 0;
    return colors[index];
}

var pages = ["P1", "P2"];
var colors = [
    {stroke: 'rgba(0, 255, 0, 1)', fill: 'rgba(0, 255, 0, 0.2)'},
    {stroke: 'rgba(255, 0, 0, 1)', fill: 'rgba(255, 0, 0, 0.2)'}
];

var courbe = [];
var smoothieChart = new SmoothieChart({tooltip: true});
smoothieChart.streamTo(document.getElementById("chart2"), 500);
pages.forEach(function (v) {
    courbe[v] = new TimeSeries();
    col = randomColor();
    smoothieChart.addTimeSeries(courbe[v], {strokeStyle: col.stroke, fillStyle: col.fill, lineWidth: 2});
});

var stockEventSource = new EventSource("/analyticsAggregate");
stockEventSource.addEventListener("message", function (event) {
    pages.forEach(function (v) {
        val = JSON.parse(event.data)[v];
        courbe[v].append(new Date().getTime(), val);
    });
});

```

## - Application.properties

```

# Le nom de la destination est R1, le consommateur va consommer les messages de cette topic
spring.cloud.stream.bindings.pageEventConsumer-in-0.destination=R1

# Le nom de la destination est R2, le producteur va produire les messages dans cette topic R2
spring.cloud.stream.bindings.pageEventSupplier-out-0.destination=R2

# Definition des fonctions Spring Cloud
spring.cloud.function.definition=pageEventConsumer;pageEventSupplier;pageEventFunction;kStreamFunction

# Intervalle de pollage fixe
spring.cloud.stream.poller.fixed-delay=100

# Le nom de la destination est R1, le consommateur va consommer les messages de cette topic R2
spring.cloud.stream.bindings.pageEventFunction-in-0.destination=R1

spring.cloud.stream.bindings.pageEventFunction-out-0.destination=R3

spring.cloud.stream.bindings.kStreamFunction-in-0.destination=R2
|
spring.cloud.stream.bindings.kStreamFunction-out-0.destination=R4

```

## Conclusion

En conclusion, cette activité pratique a permis de se familiariser avec l'installation, la configuration et l'utilisation de Kafka, ainsi que son déploiement à l'aide de Docker. De plus, l'intégration de Kafka avec Spring Cloud Streams a été abordée, ce qui a permis de comprendre la mise en place de différents services Kafka pour la production, la consommation et le traitement de flux de données en temps réel. Ces compétences sont essentielles pour travailler dans des environnements de streaming de données et de traitement analytique en temps réel, ce qui en fait une activité pratique précieuse pour les professionnels de la technologie.