

# Atelier 1 : Premier pas avec Apache KAFKA.

---

**Étape 1** : Vérifiez votre version Java.

**Étape 2** : Téléchargez la dernière version Apache Kafka et décompressez à la racine du disque.

**Étape 3** : Lancez le serveur Zookeeper

Naviguez dans le répertoire Kafka.

**Sous Linux :**

**bin/zookeeper-server-start.sh config/zookeeper.properties**

**Sous Windows :**

A la racine du dossier Kafka

**bin\windows\zookeeper-server-start.bat config/zookeeper.properties**

**Étape 4** : *Démarrez le serveur Kafka.*

**Sous Linux :**

**bin/kafka-server-start.sh config/server.properties**

**Sous Windows :**

**bin\windows\kafka-server-start.bat config\server.properties**

## Mise en place d'un cluster Kafka

### Configurer et Démarrer les Brokers 2 et 3

Pour chaque broker **Kafka** dans votre cluster, vous devrez :

Créer un fichier de configuration distinct pour chaque broker :

Copiez `server.properties` et modifiez-le pour chaque broker.

Par exemple :

**Pour le broker 1** : copiez `server.properties` en `server-1.properties`

**Pour le broker 2** : copiez `server.properties` en `server-2.properties`

**Pour le broker 3** : copiez `server.properties` en `server-3.properties`

### Modifier les configurations pour chaque broker :

Dans chaque fichier `server-x.properties`, modifiez les propriétés suivantes :

**broker.id**: Donnez un ID unique à chaque broker (0 pour le broker 1, 1 pour le broker 2, 2 pour le broker 3).

**listeners**: Si nécessaire, configurez des ports d'écoute différents pour chaque broker (par exemple, `PLAINTEXT://:9092`, `PLAINTEXT://:9093`, `PLAINTEXT://:9094`).

**log.dirs**: Configurez un répertoire de log unique pour chaque broker (par exemple, `/tmp/kafka-logs-1`, `/tmp/kafka-logs-2`, `/tmp/kafka-logs-3`).

### Démarrer chaque broker Kafka :

Ouvrez un terminal séparé pour chaque broker et exécutez :

#### # Pour le broker 1

```
bin/kafka-server-start.sh config/server-1.properties
```

#### # Pour le broker 2

```
bin/kafka-server-start.sh config/server-2.properties
```

#### # Pour le broker 3

```
bin/kafka-server-start.sh config/server-3.properties
```

## Création du premier topic

Nous allons procéder à la création d'un premier topic :

Sous Linux :

```
kafka-topics.sh --create --topic topic-test-name --bootstrap-server localhost:9092  
--replication-factor 1 --partitions 1
```

Sous Windows :

```
bin\windows\kafka-topics.bat --create --topic topic-test-name --bootstrap-server  
localhost:9092 --replication-factor 1 --partitions 1
```

## Topics détails

Pour avoir la description d'un topic on exécute la commande suivante (dans bin\windows):

```
kafka-topics.sh --bootstrap-server localhost:9092 --describe (Pour décrire tous les  
topics)
```

```
kafka-topics.sh --bootstrap-server localhost:9092 --describe --topic topic-test-name  
(Pour décrire un topic spécifique)
```

```
C:\kafka>bin\windows\kafka-topics.bat --bootstrap-server localhost:9092 --describe  
Topic: challenge      TopicId: 89ps_8pDTXe3c4Rnill3Lw PartitionCount: 1      ReplicationFactor: 1      Configs:  
Topic: challenge      Partition: 0      Leader: 0      Replicas: 0      Isr: 0  
C:\kafka>
```

## Modifier le nombre de partitions d'un topic après création

Sous Linux :

```
kafka-topics.sh --bootstrap-server localhost:9092 --alter --topic topic-test-name  
--partitions 3
```

Sous Windows:

```
kafka-topics.bat --bootstrap-server localhost:9092 --alter --topic topic-test-name  
--partitions 3
```

## Suppression d'un Topic

La suppression d'un topic entraînera naturellement la suppression des messages de ce topic

```
kafka-topics.sh --bootstrap-server localhost:9092 --delete --topic topic-test-name
```

## Lister les consumers groups

```
kafka-consumer-groups.sh --bootstrap-server localhost:9092 --list
```

## Producer & Consumer console.

Nous allons à présent tester le producer et le consumer console :

On démarre un producer en exécutant :

```
kafka-console-producer.sh --broker-list localhost:9092 --topic topic-test-name
```

Pour démarrer un consumer, depuis une nouvelle invite de commande vous pouvez exécuter la commande suivante :

```
kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic topic-test-name
```

Pour consommer les messages depuis le début :

```
kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test-topic  
--from-beginning
```

Consumer avec key et timestamp

```
kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic test-topic  
--from-beginning --property print.key=true --property print.timestamp=true
```

Une fois que les deux instances sont lancées vous pouvez émettre des messages depuis le producteur (producer) et constater leur propagation sur le consommateur (consumer).

**NB** : Vous pouvez arrêter le serveur en exécutant la commande : **kafka-server-stop**