

# Guide d'installation et d'utilisation de Kafka sur une instance EC2 Ubuntu

Konate Bachirou

## Introduction

Apache Kafka est une plateforme distribuée conçue pour gérer des flux de données en temps réel. Elle permet à des applications de produire, transmettre, stocker et consommer des données sous forme de messages.

Kafka est particulièrement utile dans les projets Big Data, d'analytique en temps réel, d'intelligence artificielle ou encore pour synchroniser plusieurs systèmes entre eux via un bus de messages fiable et scalable.

## Prérequis

Avant de commencer, vous devez disposer de :

- Une instance EC2 Ubuntu (22.04 de préférence)
- Le port TCP 9092 ouvert dans les règles de sécurité AWS
- Un terminal pour accéder à la machine via SSH
- Une connexion Internet

## 1. Connexion à l'instance EC2

Utilisez SSH pour accéder à la machine :

```
ssh -i votre_cle.pem ubuntu@<IP_PUBLIC_EC2>
```

*Remplacez 'votre\_cle.pem' par votre fichier de clé privée, et '<IP\_PUBLIC\_EC2 >' par l'adresse IP publique de l'instance.*

## 2. Installation de Java

*Kafka nécessite Java pour fonctionner. Installez OpenJDK 17 :*

```
sudo apt update
sudo apt install openjdk-17-jdk -y
java -version
```

*Vous devez voir une sortie comme :*

```
openjdk version "17..."
```

### 3. Téléchargement et extraction de Kafka

Téléchargez Kafka depuis le site officiel puis décompressez le fichier :

```
wget https://downloads.apache.org/kafka/4.0.0/kafka_2.13-4.0.0.tgz
tar -xvzf kafka_2.13-4.0.0.tgz
cd kafka_2.13-4.0.0
```

Si le téléchargement ne fonctionne pas, accédez au site via votre ordinateur physique et téléchargez kafka puis importez le dans votre VM

### 4. Configuration de Kafka (mode KRaft sans Zookeeper)

Kafka fonctionne maintenant sans Zookeeper grâce à KRaft. Modifiez la configuration :

```
nano config/broker.properties
```

Ajoutez ou modifiez les lignes suivantes :

```
process.roles=broker,controller
node.id=1
controller.quorum.voters=1@localhost:9093

listeners=PLAINTEXT://:9092,CONTROLLER://:9093
advertised.listeners=PLAINTEXT://<VOTRE_IP_PUBLIC_EC2>:9092

inter.broker.listener.name=PLAINTEXT
log.dirs=/tmp/kraft-combined-logs
```

Remplacez ' $\langle VOTRE_IP_{PUBLIC\_EC2} \rangle$ ' par l'adresse IP publique de votre machine.

### 5. Génération de l'identifiant de cluster

Kafka nécessite un identifiant unique :

```
bin/kafka-storage.sh random-uuid
```

Notez et copiez l'UUID généré (exemple : 'e89c1d9b-...').

### 6. Formatage du stockage Kafka

Initialisez le stockage en utilisant l'UUID obtenu :

```
bin/kafka-storage.sh format -t VOTRE_UUID -c config/broker.properties
```

### 7. Démarrage de Kafka en arrière-plan

Lancez Kafka en arrière-plan avec 'nohup' : (nohup permet de maintenir la connexion même si vous fermez votre terminal)

```
nohup bin/kafka-server-start.sh config/broker.properties > kafka.log 2>&1 &
```

Vérifiez que le serveur est en cours d'exécution :

```
ps aux | grep kafka
```

"Tu dois voir un processus long contenant *kafka.Kafka* → là tu sauras que *Kafka* est bien actif."

## 8. Création d'un topic Kafka

Créez un topic nommé *mon-topic* : (Un topic est comme un "canal" ou une "file de messages" nommée, dans lequel les producteurs envoient des messages et que les consommateurs lisent.)

```
bin/kafka-topics.sh --create \  
  --topic mon-topic \  
  --bootstrap-server localhost:9092 \  
  --partitions 1 \  
  --replication-factor 1
```

## 9. Test avec un producteur et un consommateur

### Producteur Kafka

Dans un terminal :

```
bin/kafka-console-producer.sh --topic mon-topic --bootstrap-server localhost:9092
```

Saisissez un message (ex. : 'Bonjour Kafka') et appuyez sur Entrée.

### Consommateur Kafka

Dans un autre terminal :

```
bin/kafka-console-consumer.sh --topic mon-topic --from-beginning --bootstrap-server  
  ↪ localhost:9092
```

Les messages saisis côté producteur apparaîtront côté consommateur.

## 10. Conseils supplémentaires

### Elastic IP

Associez une Elastic IP à votre instance EC2 pour éviter que son IP change après un redémarrage.

Pour connecter kafka à notre producteur il suffit d'utiliser IPv4 de la VM de Kafka en ouvrant le port 9093 (ex: 16.16.25.53:9092) et l'url de notre modèle ml (ex: http://13.50.13.122:5000/) et on connecte le consommateur à notre modèle de ml pour recevoir les resultats.

### Démarrage automatique (optionnel)

Pour que Kafka démarre automatiquement à chaque redémarrage, vous pouvez créer un service systemd.

## Conclusion

*Nous avons désormais un environnement Kafka complet, prêt à être utilisé pour n'importe quel projet de traitement de flux de données en temps réel. Kafka peut facilement être intégré à des systèmes plus complexes pour créer des architectures distribuées fiables, performantes et scalables.*