Hướng dẫn demo Kafka

Video demo: https://youtu.be/jAl1OpwFOd4

# Tải kafka trên ubuntu 18.04.5 LTS

## Vào thư mục main(user hiện tại có tên là ad):

**admin@LAPTOP-B4FFRFVL:~$**

## Kiểm tra java đã được cài hay chưa:

**java -version**

* Kết quả:

Ảnh có chứa văn bản

Mô tả được tạo tự động

* chưa có java

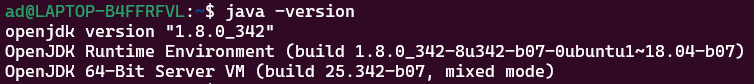
## Cài java (nếu chưa có)

**sudo apt-get update**

**sudo apt-get upgrade**

**sudo apt install openjdk-8-jdk**

Kết quả:

****

## Tải kafka

**wget** [**https://archive.apache.org/dist/kafka/2.0.0/kafka\_2.12-2.0.0.tgz**](https://archive.apache.org/dist/kafka/2.0.0/kafka_2.12-2.0.0.tgz)

Giải nén: **tar -xvf kafka\_2.12-2.0.0.tgz**

## Tạo folder chứa zookeeper và kafka server

**cd kafka\_2.12-2.0.0**

**mkdir zookeeper kafka**

## Chỉnh sửa nơi lưu snapshot của zookeeper và chạy

**nano config/zookeeper.properties**

* Điều chỉnh

dataDir=/home/ad/kafka\_2.12-2.0.0/zookeeper -> Lưu file

* Chạy zookeeper

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

## Chỉnh sửa nơi lưu log của kafka

**cd kafka\_2.12-2.0.0**

**nano config/server.properties**

* Chỉnh:

**log.dirs = /home/ad/kafka\_2.12-2.0.0/kafka** -> lưu file

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

# Demo câu lệnh kafka

**cd kafka\_2.12-2.0.0**

## Chạy zookeeper

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

## Chạy kafka

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

## Tạo một topic

* Phải khai báo số partition
* Khai báo số replication-factor
* Số replication-factor không được lớn hơn số kafka-server( broker )

**bin/kafka-topics.sh --zookeeper 127.0.0.1:2181 --topic first\_topic --create --partitions 3 --replication-factor 1**

**bin/kafka-topics.sh --zookeeper 127.0.0.1:2181 --topic second\_topic --create --partitions 6 --replication-factor 1**

## Xem danh sách topic đã được tạo

**bin/kafka-topics.sh --zookeeper 127.0.0.1:2181 --list**

## Xem mô tả một topic

**bin/kafka-topics.sh --zookeeper 127.0.0.1:2181 --topic first\_topic --describe**

## Xóa một topic

**bin/kafka-topics.sh --zookeeper 127.0.0.1:2181 --topic second\_topic --delete**

**bin/kafka-topics.sh --zookeeper 127.0.0.1:2181 --list**

## Tạo producer

**bin/kafka-console-producer.sh --broker-list 127.0.0.1:9092 --topic first\_topic**

* hello ad
* my name is an
* learning kafka

### Lựa chọn phổ cập kiến thức (acknowledgement) về dữ liệu ghi

* acks = 0: Producers gửi dữ liệu không cần phổ cập → có thể xảy ra mất mát dữ liệu (gửi nhầm vào broker đang bị sập)
* acks = 1: Producers sẽ đợi leader phổ cập - **thông báo đã nhận được dữ liệu hay không** (giới hạn được mất mát dữ liệu)
* acks =all: Leader +replicas phổ cập (không bị mất dữ liệu)

**bin/kafka-console-producer.sh --broker-list 127.0.0.1:9092 --topic first\_topic --producer-property acks=all**

* some message that is acked
* just for fun
* fun learning

## Tạo consumer

* Consumer chỉ đọc tin nhắn tại thời điểm mà nó vừa được khởi động (mặc định)

### Consume tin nhắn

**bin/kafka-console-consumer.sh --bootstrap-server 127.0.0.1:9092 --topic first\_topic**

### Produce tin nhắn

**bin/kafka-console-producer.sh --broker-list 127.0.0.1:9092 --topic first\_topic**

* Hi how are you
* This is a new message
* Play
* Hello new consumer

### Đọc toàn bộ tin nhắn

**bin/kafka-console-consumer.sh --bootstrap-server 127.0.0.1:9092 --topic first\_topic --from-beginning**

### Consume tin nhắn trong group

**bin/kafka-console-consumer.sh --bootstrap-server 127.0.0.1:9092 --topic first\_topic --group my-first-app**

### Xem consumer group

**bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 –list**

**bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --describe --group my-first-app**

* LAG: Là offset chưa đọc -> consumer đọc xong -> lag trở về 0

**bin/kafka-console-consumer.sh --bootstrap-server 127.0.0.1:9092 --topic first\_topic --group my-first-app**

**bin/kafka-console-groups.sh --bootstrap-server localhost:9092 --describe --group my-fisrt-app**

## Reset offset

### Offset trở về ban đầu

**bin/kafka-console-groups.sh --bootstrap-server localhost:9092 –group --group my-fisrt-app –reset-offsets --to-earliest –execute --topic first\_topic**

### Đọc lại tin nhắn

**bin/kafka-console-consumer.sh --bootstrap-server 127.0.0.1:9092 --topic first\_topic --group my-first-app**

### Kiểm tra lại group consumer

**bin/kafka-console-groups.sh --bootstrap-server localhost:9092 --describe --group my-first-app**

### Điểu chỉnh offsets

**bin/kafka-consumer-groups.sh –bootstrap-server localhost:9092 --group my-fisrt-app --reset-offsets –shift-by 2 –execute –topic -first-topic**

**bin/kafka-consumer-groups.sh –bootstrap-server localhost:9092 --group my-fisrt-app --reset-offsets –shift-by 2 –execute –topic -first-topic**

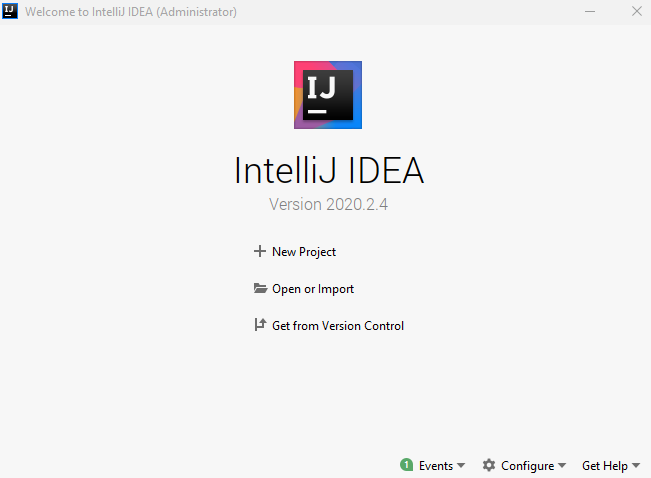
### Đọc lại tin nhắn

**bin/kafka-console-consumer.sh –bootstrap-server 127.0.0.1:9092 –topic first\_topic –group my-first-app**

# Lập trình Kafka với Java

Tải intelliji Idea: <https://download.jetbrains.com/idea/ideaIU-2020.2.4.exe>

**Tạo project mới**



Chọn maven project

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

**Vào trang:** <https://mvnrepository.com/artifact/org.apache.kafka>

Chọn kafka clients

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

Chọn phiên bản 2.0.0

Ảnh có chứa văn bản

Mô tả được tạo tự động

Ảnh có chứa văn bản

Mô tả được tạo tự động

**Copy nội dung xml rồi dán vào file pom.xml trong project**

Ảnh có chứa văn bản

Mô tả được tạo tự động

Vào trang : <https://mvnrepository.com/artifact/org.slf4j/slf4j-simple/1.7.25>

Copy content xml

Ảnh có chứa văn bản

Mô tả được tạo tự động

Dàn vào file pom.xml

Ảnh có chứa văn bản

Mô tả được tạo tự động

**Xoá phần:** <scope>test</scope>

**Tạo new package trong folder java**

Ảnh có chứa văn bản

Mô tả được tạo tự động



Tạo class ProducerDemo trong package vừa bảo bên trên

Ảnh có chứa văn bản

Mô tả được tạo tự động

## Tạo producer

Ảnh có chứa văn bản

Mô tả được tạo tự động

**Vì đang sử dụng windows subsystem linux nên ta phải lấy ip của linux -> sau khi kết nối đến thì nó sẽ tự map với ip của kafka server**

Ảnh có chứa văn bản

Mô tả được tạo tự động

## Điều chỉnh là kafka server.properties

Ảnh có chứa văn bản

Mô tả được tạo tự động

## Chạy sẵn zookeeper và kafa

**cd kafka\_2.12-2.0.0**

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

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

## Tạo mỗi consumer để lắng nghe

**bin/kafka-console-consumer.sh --bootstrap-server 127.0.0.1:9092 --topic first\_topic**

# Java producer callback

Tiếp tục thay đổi code trong ProducerDemo

Ảnh có chứa văn bản

Mô tả được tạo tự động

Và chạy demo