**Kafka 설치 방법**

**윈도우는 다른 운영체제와 다른 방식으로 동작하므로 반드시 구분하여 설치/실행해야 한다.**

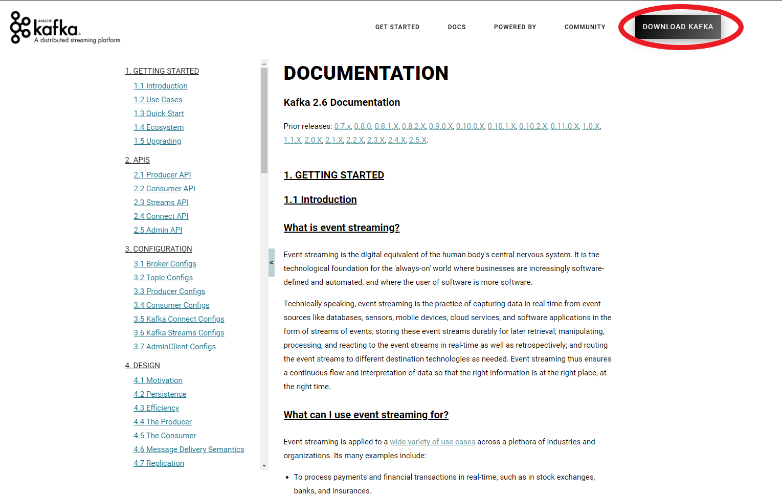
본 프로젝트에서는 JAVA 11.0.9 버전을 반드시 설치해야 합니다.

윈도우 : <https://www.oracle.com/java/technologies/javase-jdk11-downloads.html>

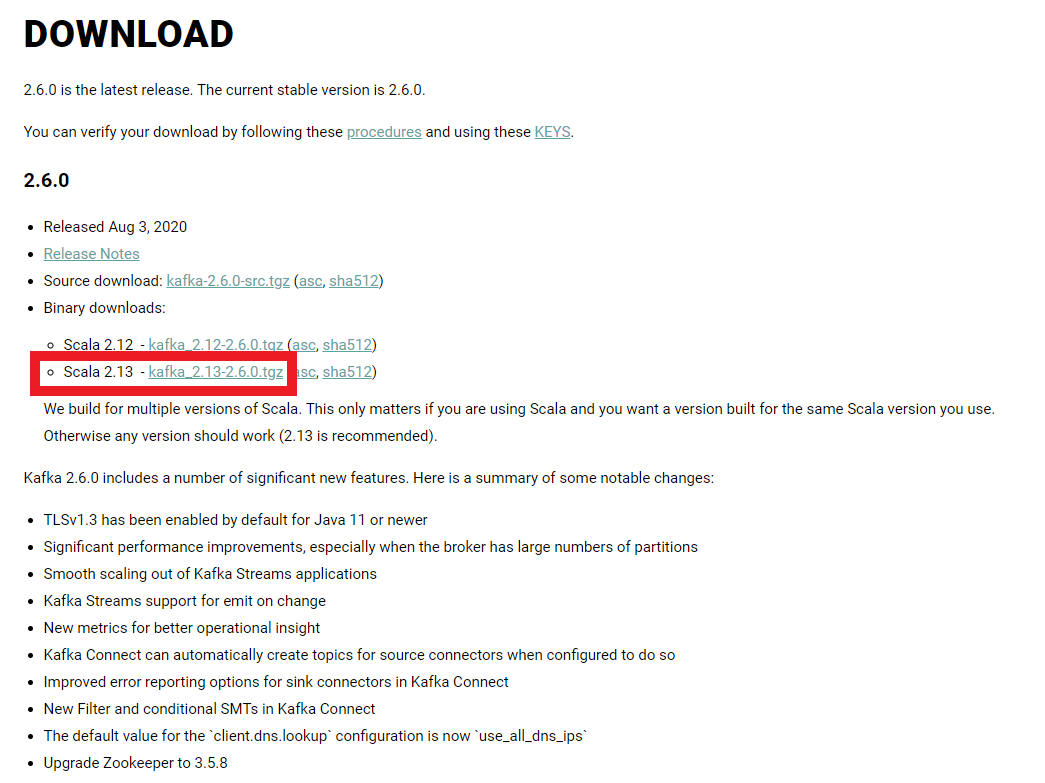
Linux: sudo apt-get install openjdk-11-jdk

**윈도우**

1. <https://kafka.apache.org/> 로 접속
2. DOWNLOAD KAFKA 클릭



1. kafka\_2.13-2.6.0.tgz 다운로드



1. kafka\_2.13-2.6.0.tgz 압축 해제
2. kafka\_2.13-2.6.0 폴더로 이동
3. kafka\_2.13-2.6.0\config\zookeeper.properties 파일에서 dataDir값을 원하는 값으로 설정

해당 directory에 zookeeper가 실행하면서 생성되는 데이터가 저장된다.

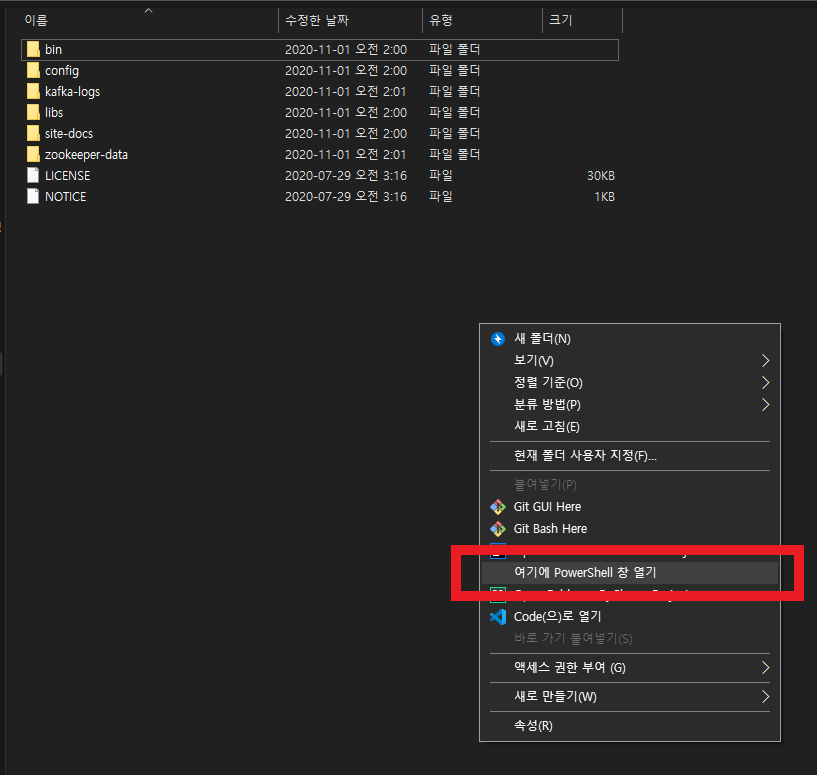
ex) dataDir=D:\\kafka\_2.13-2.6.0\\zookeeper-data

1. kafka\_2.13-2.6.0\config\server.properties 파일에서 log.dirs값을 원하는 값으로 설정

해당 directory에 kafka가 실행하면서 생성되는 데이터가 저장된다.

ex) log.dirs=D:\\kafka\_2.13-2.6.0\\kafka-logs

1. “shift + 오른쪽마우스클릭” 으로 “여기에 PowerShell 창 열기” 실행



1. bin\windows\zookeeper-server-start.bat config\zookeeper.properties
2. bin\windows\kafka-server-start.bat config\server.properties
3. 아래와 같은 명령어로 topic에 message 작성

$ bin\windows\kafka-console-producer.bat --topic quickstart-events --bootstrap-server localhost:9092

This is my first event

This is my second event

Ctrl + c 로 종료

1. 아래와 같은 명령어로 topic으로부터 message를 읽어옴

$ bin\windows\kafka-console-consumer.bat --topic quickstart-events --from-beginning --bootstrap-server localhost:9092

**Linux/Mac**

1. curl –O https://downloads.apache.org/kafka/2.6.0/kafka\_2.13-2.6.0.tgz
2. tar -xzf kafka\_2.13-2.6.0.tgz
3. cd kafka\_2.13-2.6.0
4. bin/zookeeper-server-start.sh config/zookeeper.properties
5. bin/kafka-server-start.sh config/server.properties
6. bin/kafka-topics.sh --create --topic quickstart-events --bootstrap-server localhost:9092
7. 아래와 같은 명령어로 topic에 message 작성

$ bin/kafka-console-producer.sh --topic quickstart-events --bootstrap-server localhost:9092

This is my first event

This is my second event

Ctrl + c 로 종료

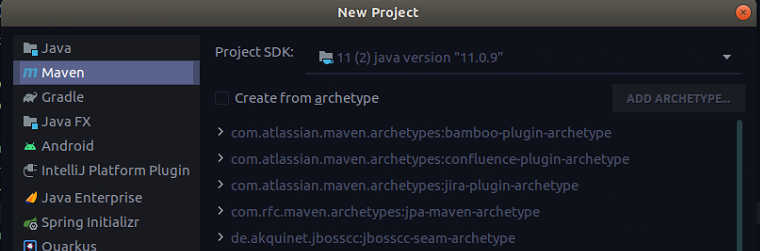
1. 아래와 같은 명령어로 topic으로부터 message를 읽어옴

$ bin/kafka-console-consumer.sh --topic quickstart-events --from-beginning --bootstrap-server localhost:9092

참고 자료 : <https://kafka.apache.org/documentation/#quickstacrt>

**Kafka 예시 프로그램**

1. IntelliJ를 설치합니다.
2. 새로운 maven 프로젝트를 생성합니다.



1. pom.xml 파일에 아래의 내용을 <project> 내부에 추가 & Ctrl + Shift + o

|  |
| --- |
| <build>  <plugins>  <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-compiler-plugin</artifactId>  <configuration>  <source>11</source>  <target>11</target>  </configuration>  </plugin>  </plugins>  </build>  <dependencies>  <!-- https://mvnrepository.com/artifact/org.apache.kafka/kafka-clients -->  <dependency>  <groupId>org.apache.kafka</groupId>  <artifactId>kafka-clients</artifactId>  <version>2.6.0</version>  </dependency>  <!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind -->  <dependency>  <groupId>com.fasterxml.jackson.core</groupId>  <artifactId>jackson-databind</artifactId>  <version>2.11.3</version>  </dependency>    </dependencies> |

1. Producer.java 파일을 생성하고 Shift + F10

|  |
| --- |
| import java.util.Properties;  import org.apache.kafka.clients.producer.KafkaProducer;  import org.apache.kafka.clients.producer.ProducerConfig;  import org.apache.kafka.clients.producer.ProducerRecord;  public class Producer {  public static void main(String[] args) {  Properties config = new Properties();  config.put(ProducerConfig.BOOTSTRAP\_SERVERS\_CONFIG, "localhost:9092");  config.put(ProducerConfig.CLIENT\_ID\_CONFIG, "id1");  config.put(ProducerConfig.KEY\_SERIALIZER\_CLASS\_CONFIG, "org.apache.kafka.common.serialization.StringSerializer");  config.put(ProducerConfig.VALUE\_SERIALIZER\_CLASS\_CONFIG, "org.apache.kafka.common.serialization.StringSerializer");  config.put(ProducerConfig.LINGER\_MS\_CONFIG, 1);  // config.put(ProducerConfig.COMPRESSION\_TYPE\_CONFIG, "lz4");  KafkaProducer<String, String> producer = new KafkaProducer<>(config);  ProducerRecord<String, String> record = new ProducerRecord<>("test2", "12", "hahdfdfa");  producer.send(record);  producer.close();  }  } |

1. Consumer.java 파일을 생성하고 Shift + F10

|  |
| --- |
| import java.util.Collections;  import java.util.Properties;  import org.apache.kafka.clients.consumer.ConsumerConfig;  import org.apache.kafka.clients.consumer.ConsumerRecord;  import org.apache.kafka.clients.consumer.ConsumerRecords;  import org.apache.kafka.clients.consumer.KafkaConsumer;  public class Consumer {  public static void main(String[] args) {  Properties config = new Properties();  config.put(ConsumerConfig.GROUP\_ID\_CONFIG, "fo2s234dfsdf");  config.put(ConsumerConfig.BOOTSTRAP\_SERVERS\_CONFIG, "localhost:9092");  config.put(ConsumerConfig.KEY\_DESERIALIZER\_CLASS\_CONFIG, "org.apache.kafka.common.serialization.StringDeserializer");  config.put(ConsumerConfig.VALUE\_DESERIALIZER\_CLASS\_CONFIG, "org.apache.kafka.common.serialization.StringDeserializer");  config.put(ConsumerConfig.AUTO\_OFFSET\_RESET\_CONFIG,"earliest");  config.put(ConsumerConfig.ENABLE\_AUTO\_COMMIT\_CONFIG, "false");  KafkaConsumer<String, String> consumer = new KafkaConsumer<>(config);  consumer.subscribe(Collections.singletonList("test2"));  try {  while (true) {  ConsumerRecords<String, String> records = consumer.poll(1000);  for (ConsumerRecord<String, String> record : records)  {  System.out.printf("topic = %s, partition = %s, offset = %d, customer = %s, country = %s\n",  record.topic(), record.partition(), record.offset(), record.key(), record.value());  }  }  } finally {  consumer.close();  }  }  } |

참고 자료

https://docs.confluent.io/current/clients/java.html#

Kafka: The Definitive Guide(무료로 e-book을 받을 수 있음)