High Level Design Approach & Implementation

For

Kafka POC

|  |  |
| --- | --- |
| **Document:** | Kafka Producer and Consumer |
| **Author:** | Srinadh |
| **Revision Number:** | 0.1 |
| **Status:** | Final |
| **Revised Date:** | 12/10/2018 |

HLDS Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision Number | Date | Author | Description of Change |
| 0.1 | 12/10/2018 | Srinadh | Final version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of contents

1.0 Introduction and Overview 4

2.1 Purpose 4

2.2 Document Scope 4

3.0 Assumptions 4

4.0 System Architecture Diagram 4

5.0 Steps to Achieving Requirements 4

6.0 Technologies 9

6.1 JAVA SE 8 9

6.2 Zookeeper-3.4.12 9

6.3 Kafka\_2.12-2.0.0 9

6.4 Maven Build Tool 9

7.0 Appendix 10

7.1 Reporting 10

# Introduction and Overview

1. Create one POC project which holds the functionality of Kafka. Need to trigger the email or sms from the project. Kafka is one data robust communication channel which can passing the data one end another end with losing data. Kafka is often used in real-time streaming data architectures to provide real-time analytics. Since Kafka is a fast, scalable, durable, and fault-tolerant publish-subscribe messaging system, Kafka is used in use cases where JMS, RabbitMQ, and AMQP may not even be considered due to volume and responsiveness

## Purpose

The purpose of this document is to provide the implementation procedures of Kafka message process by using producer and consumer.

## Document Scope

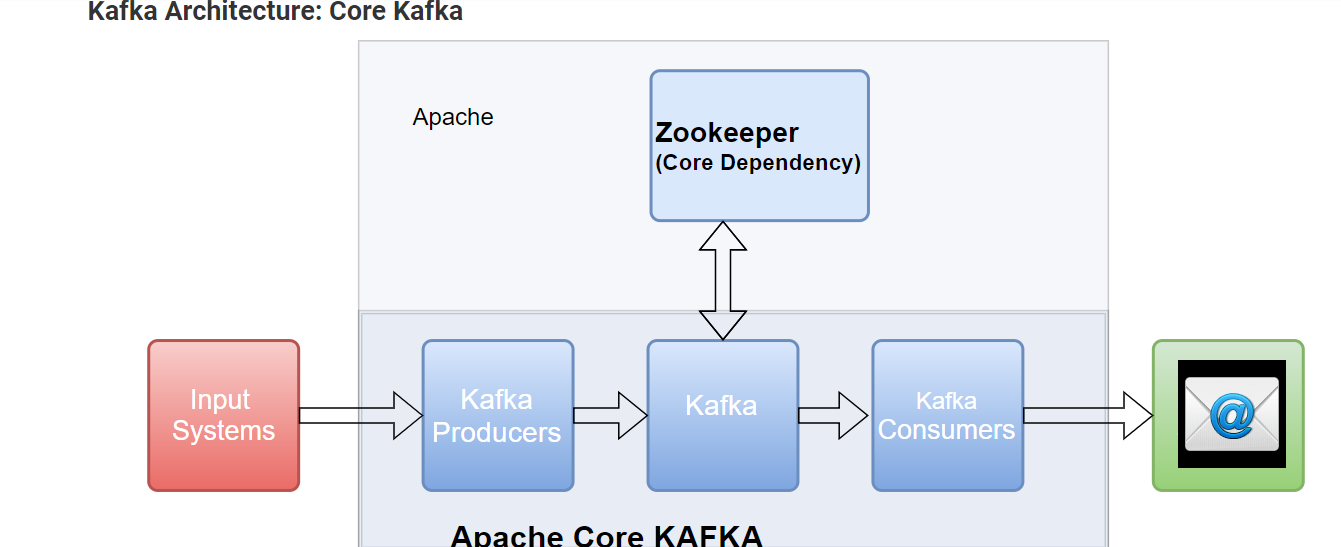
The scope of this document is to provide detailed view of Kafka POC data streaming approach involved in implementation of producer and consumer technique along with their interactions with external systems.

# Assumptions

* This POC used only for technical assessment verification.

# System Architecture Diagram

Below diagram depicts different components involved in the system.



# Steps to Achieving Requirements

* Download and install Zookeeper

|  |
| --- |
| * Zookeeper Setup On windows * Assuming that you have jdk 8 installed already let us start with installing and configuring zookeeper on Windows.Download zookeeper from https://zookeeper.apache.org/releases.html. I have downloaded zookeeper version 3.4.10 as in the kafka lib directory, the existing version of zookeeper is 3.4.10.Once downloaded, follow following steps: * 1. Extract it and in my case I have extracted kafka and zookeeper in following directory: * C:\D\softwares\kafka\_2.12-1.0.1 --kafka location * C:\D\softwares\kafka-new\zookeeper-3.4.10 --zookeeper location * 2. Once this is extracted, let us add zookeeper in the environment variables.For this go to Control Panel\All Control Panel Items\System and click on the Advanced System Settings and then Environment Variables and then edit the system variables as below:   https://i.imgur.com/ICVZ0Va.png  **3.**Also, edit the PATH variable and add new entry as %ZOOKEEPER\_HOME%\bin\ for zookeeper.  **4.**Rename file C:\D\softwares\kafka-new\zookeeper-3.4.10\zookeeper-3.4.10\conf\zoo\_sample.cfg to zoo.cfg  **5.**Now, in the command prompt, enter the command zkserver and the zookeeper is up and running on [http://localhost:2181](http://localhost:2181/) |

* Download and install Kafka

|  |
| --- |
| * **1.**Unzip the downloaded binary. In my case it is - C:\D\softwares\kafka\_2.12-1.0.1 * **2.**Go to folder C:\D\softwares\kafka\_2.12-1.0.1\config and edit server.properties * **3.**log.dirs=.\logs * **4.**Now open a new terminal at C:\D\softwares\kafka\_2.12-1.0.1. * **5.**Execute .\bin\windows\kafka-server-start.bat .\config\server.properties to start Kafka. Since, we have not made any changes in the default configuration, Kafka should be up and running on [http://localhost:9092](http://localhost:9092/)     Let us create a topic with a name devglan-test  cd C:\D\softwares\kafka\_2.12-1.0.1\bin\windows  kafka-topics.bat --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic devglan-test  Now let us create a producer and consumer for this topic.  cd C:\D\softwares\kafka\_2.12-1.0.1\bin\windows  C:\D\softwares\kafka\_2.12-1.0.1\bin\windows>kafka-console-producer.bat --broker-list localhost:9092 --topic devglan-test --producer  C:\D\softwares\kafka\_2.12-1.0.1\bin\windows>kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic devglan-test --consumer  Producer can produce messages and consumer can consume messages in the following way from the terminal |

* Import Attached Maven Project code into eclipse

|  |
| --- |
| Please start the consumer by running java command on Consumer.java and Start the producer by running java class Producer.java and give user input messages these messages received by consumer and it send to configured email in code. |

# Technologies

## JAVA SE 8

Java Development Kit used for developing this POC which uses standard library provided by Oracle JDK in Java Runtime environment.

## Zookeeper-3.4.12

ZooKeeper is used to coordinate the brokers/cluster topology. ZooKeeper is a consistent file system for configuration information. ZooKeeper gets used for leadership election for Broker Topic Partition Leaders

## Kafka\_2.12-2.0.0

Kafka is often used in real-time streaming data architectures to provide real-time analytics. Since Kafka is a fast, scalable, durable, and fault-tolerant publish-subscribe messaging system, Kafka is used in use cases where JMS, RabbitMQ, and AMQP may not even be considered due to volume and responsiveness

## Maven Build Tool

The maven build used for this POC to download all supported artifacts for implementing kafka producer and consumer.

## Reference Links

<http://blog.empeccableweb.com/wp/2016/11/30/kafka-producer-and-consumer-example/>

<https://www.devglan.com/apache-kafka/apache-kafka-java-example/>

# Appendix

## Reporting