Help needed in resolving connectivity issue between Onprem and Kafka VM on AWS through proxy

**Document Details**

|  |  |
| --- | --- |
| Name of Document | TopGear Challenge - Help needed in resolving connectivity issue between Onprem and Kafka VM on AWS through proxy |
| Customer | Internal |
| Current Version | 1.0 |
| List of Contributors | Manish Chakraborty |
| Contact Information |  |

**Revision History**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | Date of Revision | Description of Change | Reason for Change | Affected Sections | Approved By |
| 1.0 |  | Final | - | All |  |

# Problem Statement

Challenge Title: Help needed in establishing connectivity between on-premise and Kafka VM on AWS Key details:-

• Kafka broker is running on AWS cloud VM on port 9092

• Kafka VM sits behind a proxy on AWS; Kafka VM is not directly exposed to outside world

• Kafka needs to communicate by outside components from on-premise (producer on premise) and from components within AWS cloud (consumer on same AWS)

• There are 2 Kafka brokers and they are configured through Cloudera (CDH: 5.12.1 Kafka: 0.10.2.0)

Issue faced:-

• Unable to communicate from on-premise component to Kafka on cloud through proxy

Help needed:-

• Need help in understanding how to enable the following communications:-

• Producer running from on-premise and trying to connect to Kafka on AWS through proxy

• Consumer running locally within AWS with the replication happening to the second Kafka node

Aldo, When the Producer is sending the message it is sent only to the Leader node of the Kafka broker and not getting replicated to the second broker. Also the consumer component running locally within AWS couldn’t connect via locally.

Key properties tried:- advertised.listeners=PLAINTEXT://proxyhostname:8080 - for external communication listeners=PLAINTEXT://:9092 = for internal communication When we set the above mentioned properties, internal communication is not happening.

This section onwards, we will discuss the steps to be taken to resolve the above mentioned issues.

# Configurations

This section discusses the expected configurations to be carried out.

## Broker Configuration

Assuming that there are 3 brokers, the below configurations needs to be carried out for 3 brokers:

|  |  |  |
| --- | --- | --- |
| # | Broker Details | Property File Name |
| 1 | Broker Id 0 | server-0.properties |
| 2 | Broker Id 1 | server-1.properties |
| 3 | Broker Id 2 | server-2.properties |

## Brokers Contents

Create 3 Brokers with the minimum information that are required.

|  |  |  |
| --- | --- | --- |
| # | Property File Name | Minimum Contents |
| 1 | server-0.properties | broker.id=0  port=9092  log.dir=/tmp/kafka-logs-0  min.insync.replicas=3  num.partitions=1  zookeeper.connect=localhost:2181  listeners=PLAINTEXT://0.0.0.0:9092  advertised.listeners=PLAINTEXT://10.91.192.124:8080 |
| 2 | server-1.properties | broker.id=1  port=9093  log.dir=/tmp/kafka-logs-1  min.insync.replicas=3  num.partitions=1  zookeeper.connect=localhost:2181  listeners=PLAINTEXT://0.0.0.0:9093  advertised.listeners=PLAINTEXT://10.91.192.125:8080 |
| 3 | server-2.properties | broker.id=2  port=9094  log.dir=/tmp/kafka-logs-2  min.insync.replicas=3  num.partitions=1  zookeeper.connect=localhost:2181  listeners=PLAINTEXT://0.0.0.0:9094  advertised.listeners=PLAINTEXT://10.91.192.126:8080 |

## Other Configurations

Other configurations are as below for **Producer**:

|  |  |  |
| --- | --- | --- |
| # | Area | Configurations Information |
| 1 | Bootstrap.servers | List of KAFKA Broker URL and PORT Number  Example:  props.put(“bootstrap.servers”, “host1. 9092, host2:9093, host3:9094”); |
| 2 | Acks | 0 [The producer will not wait for the response]  1 [The producer will wait for the response. In this case response is sent by **leader**]  all [The leader will ack post it receives from all replicas] |